

Kimon P. Valavanis
George J. Vachtsevanos
Editors

Handbook of Unmanned Aerial Vehicles



SpringerReference

Handbook of Unmanned Aerial Vehicles

Kimon P. Valavanis • George J. Vachtsevanos
Editors

Handbook of Unmanned Aerial Vehicles

With 1228 Figures and 233 Tables



Springer Reference

Editors

Kimon P. Valavanis
John Evans Professor and Chair
Department of Electrical and Computer
Engineering
Daniel Felix Ritchie School of
Engineering and Computer Science
University of Denver
Denver, CO, USA

George J. Vachtsevanos
Professor Emeritus
School of Electrical and Computer
Engineering
The Georgia Institute of Technology
Atlanta, GA, USA

ISBN 978-90-481-9706-4 ISBN 978-90-481-9707-1 (eBook)
ISBN 978-90-481-9708-8 (print and electronic bundle)
DOI 10.1007/978-90-481-9707-1
Springer Dordrecht Heidelberg New York London

Library of Congress Control Number: 2014944662

© Springer Science+Business Media Dordrecht 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Stella and Panos, my children, and to Dina

Kimón

*To my wife Athena and my granddaughters
Ellie, Athena, Sophia and Martha*

George

Foreword: Civil RPAS In European Union Airspace – The Road To Integration



By Peter van Blyenburgh
President, UVS International¹

Preamble

The European remotely piloted aircraft systems (RPAS)² industrial community consists of two stakeholder groups (Industry³ & SMEs/SMIs⁴, which both encompass manufacturers and operators⁵. The civil RPAS market comprises three user groups: commercial, non-commercial (incl. corporate operations & research), and governmental non-military (state & non-state flights) operators (See Fig. 1). Currently, the regulatory responsibilities in the European Union (EU) for civil remotely piloted aircraft (RPA) with a maximum take-off mass (MTOM) of more than 150 kg lay with the European Aviation Safety Agency (EASA), and for RPA with a MTOM of less than 150 kg with the relevant national aviation authorities (NAAs)⁶.

Background

Various initial national regulations relative to the operation of civil RPAS are now in place (Austria, Czech Rep., Denmark, France, Germany, Ireland, Italy,

| | | | | RPAS Ops | |
|------------------|---|--------------|---|--|--------------------------------------|
| Governmental | M U T U A L I Z A T I O N | Military | | ◆ | |
| Non-Governmental | | Non-Military | State Flights <i>Security related</i> | ◆ ◆ ◆ ◆ | N O N |
| | | | Not State Flights <i>Incl. Safety related</i> | ◆ ◆ ◆ | |
| | | Public | European Union | Flights on behalf of public EU agency <i>(without national oversight)</i> | M I L I T A R Y |
| | | | Commercial Air Transport <i>(Transport of Persons & Freight)</i> | Scheduled Air Service Non-scheduled Revenue Operations Non-revenue Operations | |
| Blyenburgh © | | | General Aviation | Corporate Operations Flight Training / Instruction Pleasure | R P A S |
| | | | Aerial Work / Specialized Operations | Commercial Non-Commercial <i>(incl. Corporate Operations)</i> Training / Instruction Other Miscellaneous | |

Fig. 1 Aerial Operations - Current & Near-Future RPAS Usage

Sweden, UK), are about to enter into force (Belgium, Finland, Lithuania, Norway, Switzerland), or are in preparation (Malta, The Netherlands, Spain) – See Fig. 2. These regulations principally concern Light RPAS⁷ and they are not harmonized on a pan-European level. The coming into force of these regulations, which is fairly recent in most of these countries, has brought about a dynamic growth in approved⁸ and authorized⁹ RPAS operators supplying aerial operation services with an ever-growing diversity of applications¹⁰ (See Fig. 3). For reasons of simplicity, approved and authorized operators are hereinafter jointly referred to as certified operators¹¹. In fact, there are currently more than 1500 certified civil RPAS operators in the European Union (See Fig. 4). In practically all cases, current flight operations are taking place within visual line-of-sight (VLOS), at a flight altitude of less than 500 ft above ground level (AGL) with RPA with a MTOM of less than 25 kg, which are principally produced and operated by SMIs. It should also be mentioned that a significant amount of European NAAs facilitate RPAS operations by granting Permits-to-Fly on a case-by-case basis. The European RPAS industrial community now has as objective to initiate activities that will make it possible to start to enlarge the currently permitted flight envelopes, and to operate larger RPAS.

A growing number of governmental authorities in various EU countries involved with internal security matters (e.g. municipal & national police, anti-terrorist squads, municipal fire brigades, forest fire fighters, coast guard, civil defence, environmental protection agencies), as well as FRONTEX (European agency responsible for EU border security), have manifested great interest in the use of RPAS. In addition, large corporate entities (e.g. electric grid operators, pipeline network operators, railway operators, oil companies) have woken up to the fact that RPAS could fulfil functions that would be extremely beneficial to their corporate operations. However, in all cases the use of RPAS will depend on the acceptability of the proposed RPAS flight hour cost.

| | | MTOM | In place | | MTOM | In Preparation | | Comments |
|----|--------------------------|----------------------------|-----------|----------|-----------------------------|----------------|----------|---|
| 01 | Austria ^F | < 150 kg | VLOS | | | | | |
| 02 | Belgium ^F | | | | < 150 kg | VLOS | | Finalised (2013) but not in force |
| 03 | Bulgaria | | | | | | | |
| 04 | Croatia | | | | | | | |
| 05 | Cyprus | | | | | | | |
| 06 | Czech Rep. ^F | < 150 kg | VLOS | BLOS | | | | In place (May 2013) |
| 07 | Denmark ^F | < 150 kg | VLOS | | | | | In place (Jan. 2004) |
| 08 | Estonia | | | | | | | |
| 09 | Finland ^F | | | | < 150 kg | VLOS | | Expected mid 2014 |
| 10 | France ^F | < 25 kg | VLOS | BLOS | < 150 kg | VLOS | BLOS | In place (Apr. '12) To be update in 2014 |
| 11 | Germany | < 25 kg | VLOS | | | | | In place (2013) |
| 12 | Greece ^F | | | | | | | |
| 13 | Hungary ^F | | | | < 150 kg | VLOS | | |
| 14 | Ireland ^F | < 20 kg | VLOS | | | | | In place May 2012) |
| 15 | Italy ^F | < 25 kg | VLOS | | | | | In place (Dec. 2013) |
| 16 | Latvia | | | | | | | |
| 17 | Lithuania ^F | < 25 kg | VLOS | | < 150 kg | VLOS | | Expected mid 2014 |
| 18 | Luxembourg | | | | | | | |
| 19 | Malta ^F | | | | < 150 kg | VLOS | | In preparation |
| 20 | Netherlands ^F | < 25 kg | VLOS | | < 150 kg | VLOS | | In place (2012) Update in preparation |
| 21 | Poland ^F | < 150 kg | VLOS | BLOS | | | | In place 2013 |
| 22 | Portugal | | | | | | | |
| 23 | Romania ^F | | | | | | | |
| 24 | Slovakia | | | | | | | |
| 25 | Slovenia ^F | | | | | | | |
| 26 | Spain ^F | | | | < 25 kg | VLOS | | In preparation |
| 27 | Sweden ^F | < 150 kg | VLOS | | | | | In place (Mar. 2013) |
| 28 | UK ^F | < 20 kg | VLOS | | | | | In place (2002) Several updates since |
| | Sub-Total | | 12 | 3 | | 8 | 1 | |
| 29 | Iceland | | | | < 150 kg | VLOS | BLOS | |
| 30 | Norway ^F | | | | | | | Expected mid 2014 |
| 31 | Switzerland ^F | Model aircraft rules apply | | | VLOS over people/ crowds | | | Directive expected 2014 |
| | Sub-Total | | | | | 1 | 2 | |
| | Total | | 12 | 3 | | 9 | 2 | |

^F RPAS aerial operations facilitated = Permit to Fly is granted by NAA, based on specific national rules, possibly on a case-to-case basis & for a limited duration (*incl. in countries where no national regulation exists*).

© Blyenburgh

Fig. 2 RPAS Regulation in Europe

| | | Explanation of Terms |
|---|---|--|
| Commercial & Non-Commercial <i>(including Corporate Operations)</i> <ul style="list-style-type: none"> - Aerial Advertising - Aerial Inspection - Aerial Monitoring - Aerial Observation & Surveillance - Aerial Patrol & Spotting - Aerial Photography, Video, Cinema - Aerial Survey & Mapping (Photogrammetry) - Aerial Spraying & Dispensing - Research & Scientific - Aerial Search & Rescue Assistance Blyenburgh © | Flight Training / Instruction <ul style="list-style-type: none"> - Duo <i>(student instruction by licensed pilot)</i> - Solo <i>(unaided student flight)</i> - Check <i>(verification of qualification of pilot license holder)</i> | Inspection: Examination with the intent to find faults, errors, problems, malfunctions or specific phenomena. Monitoring: Observing on a regular basis over a period of time. Observation: Examination of an activity, person, group, area or phenomena. Patrol: Searching for a specific activity, person, group or phenomena. Spotting: Looking for & noting geographical coordinates of an object or activity or phenomena. Surveillance: Close observation of an activity, person, group, area or phenomena. Survey: Detailed inspection of a geo-referenced section of the earth's surface (including structures) with the purpose to study or measure altitudes, angles, distances, phenomena, on the land and the structures flown over |
| | Other Miscellaneous <ul style="list-style-type: none"> - Test / Experimental - Demonstration - Ferry / Positioning - Air Show / Race | |

Fig. 3 Civil RPAS Aerial Work Categories

A considerable amount of civil RPAS related European Commission (EC) initiatives have taken place over the years, namely, the Hearing on Light RPAS (Oct. '09), the High Level UAS Conference (July '10)¹², the UAS Panel initiative (Jun. '11-Feb. '12)¹³, the European RPAS Steering Group initiative (Jun. '12 - today)¹⁴, and a significant amount of RPAS-related study contracts have been financed by various EC Directorate Generals (DG)¹⁵. It should be noted that two study contracts have recently been allocated by DG Enterprise & Industry concerning the societal topics of respectively Liability & Insurance, and Privacy & Data Protection. FRONTEX¹⁶ has financed several RPAS capability demonstrations in Spain and Greece to evaluate RPAS usage for illegal immigrant control. In addition, in November 2013 SESAR Joint Undertaking (SJU) announced the 9 consortia that had been selected for funded RPAS demonstration activities. The objective of these demonstrations, which will take place over a period of 24 months (ending in Oct. '15), is to prepare the integration of operational improvements in SESAR's ATM Master Plan that should contribute to the safe integration of RPAS operations into European non-segregated airspace starting in 2016.

The conclusions of the EC UAS Panel¹⁷, co-chaired by the EC's DG Enterprise & Industry and DG Mobility & Transport highlighted the potential of civil RPAS for the development of a wide range of commercial and non-commercial, as well as non-military governmental, applications. In addition, the EC concluded that civil RPAS offer significant potential for job creation in the industry and services sectors, and can generate positive economic growth. The underlying principal of the EC UAS Panel was to focus on civil RPAS applications with societal benefits for which a viable business could be made. The need to accelerate the safe integration of RPAS into European airspace - taking into account societal issues such as responsibility, liability, insurance, privacy, data protection, and public acceptance – was also recognized. Furthermore, it was brought to light that, viable business cases for civil RPAS operations can currently, and for the foreseeable future, only be made for Light RPAS.

The Joint Authorities for Rulemaking on Unmanned Systems (JARUS)¹⁸ federates the NAAs of 22 countries¹⁹, as well as EASA and EUROCONTROL. JARUS contributes to the harmonisation and coordination of the rulemaking, certification, and operational approval of civil RPAS, remote pilots and operators. Its intent

| | Quantity Certificated | Operator | Pilot | Manufacturer | RPAS | Regulation |
|----|--|-------------------|-----------|--------------|-----------|---|
| 01 | Austria ^F | | | | | In place (2014) |
| 02 | Belgium ^F | 10 | | | | Finalised (2013) but not in force |
| 03 | Bulgaria | | | | | |
| 04 | Croatia | | | | | |
| 05 | Cyprus | | | | | |
| 06 | Czech Rep. ^F | 24 | 29 | 31 | | In place (May 2013) |
| 07 | Denmark ^F | 13 | | | | In place (Jan. 2004) |
| 08 | Estonia | | | | | |
| 09 | Finland ^F | 33 | | | | Expected mid 2014 |
| 10 | France ^F | 497 | 30 | | | In place (Apr. 2012) To be updated in 2014 |
| 11 | Germany | 400 (estimate) | | | | In place (2013) [Länder & Federal level (no reciprocity)] |
| 12 | Greece ^F | | | | | |
| 13 | Hungary ^F | | | | | |
| 14 | Ireland ^F QE | 12 | | | | In place (May 2012) |
| 15 | Italy ^F | 1 | | 1 | | In place (Dec. 2013) |
| 16 | Latvia | | | | | |
| 17 | Lithuania ^F | | | | | In place (Jan. 2002) Update mid 2014 |
| 18 | Luxembourg | | | | | |
| 19 | Malta ^F QE | | | | | In preparation |
| 20 | Netherlands ^F QE | 13 | 30 | | | In place (2012) Update in preparation |
| 21 | Poland ^F | | | | | In place (2013) |
| 22 | Portugal | | | | | |
| 23 | Romania ^F | | | | | |
| 24 | Slovakia | | | | | |
| 25 | Slovenia ^F | | | | | |
| 26 | Spain ^F | 1 | | | | In preparation |
| 27 | Sweden ^F | 216 | | | | In place (March 2003) |
| 28 | UK ^F QE | 212 | | | | In place (2002) Several updates since |
| | Sub-Total | 1432 | 89 | 32 | 33 | |
| 29 | Iceland | | | | | |
| 30 | Norway ^F | 76 | | | | Expected mid 2014 |
| 31 | Switzerland ^F | | | | | Directive expected 2014 |
| | Sub-Total | 76 | | | | |
| | Total | 1508 | 89 | 32 | 33 | |
| | Certification: A form of official recognition of compliance with the applicable requirements by means of the issuance of a certificate attesting such compliance. ^F RPAS aerial operations facilitated = Permits to fly are granted by NAA, based on specific national rules, possibly on a case-to-case basis for a limited duration (<i>incl. in countries where no national regulation exists</i>). QE National Qualified Entity (QE) in place, QE from another country recognised, or QE in the process of being appointed. | | | | | |

© Blyenburgh

Fig. 4 RPAS (MTOM) Related Certification in Europe

is to eliminate the need for each country to write their individual requirements and to facilitate reciprocal acceptance of RPAS-related certificates, approvals and licenses. In November 2013, JARUS published its first certification specification (CS-LURS)²⁰ and the online comment period of its second specification (AMC RPAS.1309)²¹ ended on 28 March 2014.

The creation of the European RPAS Steering Group (ERSG), co-chaired by the EC's DG Enterprise & Industry and DG Mobility & Transport and representing 11 stakeholders²², was announced by the EC at the Paris Air Show in 2012. Its objective is to foster the development of civil RPAS operations by planning and coordinating all activities necessary to achieve the safe and incremental integration of civil RPAS into European air traffic starting in 2016. The ERSG has developed a comprehensive European RPAS Roadmap²³, consisting of three complementary sub-sets²⁴ that was remitted by Peter van Blyenburgh, as a representatives of UVS International, to Matthias Ruete, Director General Mobility & Transport & Philippe Brunet, Director DG Enterprise & Industry, as representatives of the European Commission at the Paris Air Show in 2013. The European RPAS Roadmap identified JARUS as a key player for its implementation - This fact will most probably bring with it the necessity to give JARUS, in the near future, a recognised status with a formal link-up to EASA.

The Current Situation

On Monday 3 February 2014, SJU published a tender²⁵, which has as purpose to select a consortium that will be entrusted with the preparation of the "Definition Phase" related to the insertion of civil RPAS into the European Aviation System; this tender should be seen in light of the guidelines defined by the European RPAS Roadmap, and within the context of the Single European Sky initiative. In order to be truly representative of the entire European RPAS industrial community and be able to stay in line with the conclusions of the EC Staff Working Document "Towards a European strategy for the development of civil applications of RPAS²⁶", as well as the European RPAS Roadmap, the selected consortium should federate large industrial enterprises and relevant SMEs/SMIs (manufacturers & operators) with measureable experience, so that it can effectively assure that the consortium's work will indeed address all the topics required for the incremental integration of all classes of civil RPAS, starting in 2016. The tender closing date was 2 April 2014 and the winning consortium should be announced before the summer of 2014.

At a news conference that took place in Brussels, Belgium on 8 April 2014, Siim Kallas, the vice president of the European Commission and Commissioner for Transport, officially announced the acceptance of the European RPAS Roadmap. On the same day, a communication to the European Parliament and the European Council entitled "A New Era for Aviation – Opening the aviation market to the civil use of remotely piloted aircraft systems in a safe & sustainable manner" was published by the EC. The EC also referred to the European Summit of 19 December 2013, which called for action to enable the progressive integration of RPAS into the

total aviation system and European airspace from 2016 onwards. In addition, the EC stated that the progressive integration of RPAS should be accompanied by adequate public debate on the development of measures that address societal concerns.

The implementation of the European RPAS Roadmap will be instrumental to lay the foundation for the creation of jobs and economic growth in the research, development, production and services sectors in Europe and will favourably position the European RPAS community on the world stage. This situation clearly demonstrates the coordinated and pragmatic European policy within the framework of the Single European Sky (SES) initiative, which has been made possible by the cooperation of multiple stakeholder groups.

In order to contribute in a positive and pro-active manner to this initiative UVS International has instigated the creation of a multi-disciplinary study group²⁷ to deal with the topics of responsibility, liability and insurance. It is the intent of this study group, which was announced in December 2013, to produce position papers relative to the topics identified in its Terms of Reference²⁷ and submit them for consideration to the European Commission, as well as all other interested parties. The first position papers produced by this group will be presented at the upcoming RPAS 2014 conference in Brussels, Belgium on 23-26 June 2014²⁸.

Considerations & Recommendations

RPAS encompass many different types of aircraft with a huge variety of airframe types and lift technologies, with maximum take-off weights ranging from several grams to well over 10 000 kg, with speeds varying from hovering to over 1 000 km/h, and with a very wide envelope of flight endurances (from minutes to days). Therefore, RPAS requires a very specific and proportionate approach to rulemaking, which does not impose an excessive burden on the SMEs/SMIs producing and operating them, but that also does not entail unacceptable administrative procedures for them, as well as for the NAAs. However, this proportionate approach should in no way comprise current aviation safety levels.

The current regulatory responsibility in the EU (MTOM <150 kg: NAAs; MTOM >150 kg: EASA), which is based on traditional airworthiness considerations, is not considered to be conducive for the creation of a coherent harmonised EU safety policy relative to RPAS. The challenge will be to create rules that are proportionate to risk, while taking into account some, or all, of the following points: MTOM, speed, system complexity, airspace class, population density in the area over-flown, and the specificity of operation.

Besides the technical and research aspects relative to the air traffic management and related issues, that will be addressed by SJU's aforementioned "Definition Phase" tender, there are a number of other just as critical topics that have to be addressed within the context of European and international harmonisation, and with the objective to be able to comply with the EU principal of free movement of persons, products and services throughout the 28 EU Member States.

These topics include, amongst others:

- a) Operator (commercial & non-commercial) certification & assessment;
- b) Remote pilot training (theoretical & practical), licensing & proficiency assessment;
- c) Airworthiness & continued airworthiness assessment;
- d) Design organization approval; e) Manufacturer organization approval.

In addition, particular attention should be paid to harmonizing the criteria relative to:

- 1) Remote pilot training syllabus;
- 2) RPAS operation manuals;
- 3) RPAS instruction manuals (supplied by the manufacturer);
- 4) RPAS maintenance manuals;
- 5) RPAS & operation categories;
- 6) Terminology.

As mentioned earlier, the following societal issues should also be addressed:

- a) Manufacturer, operator & remote pilot responsibility;
- b) Manufacturer, operator & remote pilot liability & insurance;
- c) Privacy & data protection;
- d) Positive awareness creation with the general public relative to RPAS.

Many of these subject matters will have to be dealt with by industry through standards organizations (EUROCAE) and dedicated industry working groups, in close coordination with the EC, JARUS (and NAAs), EASA, EUROCONTROL, SJU, the European Space Agency (ESA), and the European Defence Agency (EDA). It should be noted that the two study contracts awarded by EC DG Enterprise & Industry relative to RPAS Liability & Insurance and Privacy & Data Protection should be seen within this framework. It should also be noted that a substantial part of the initially required rules can be developed by JARUS and can be harmonised by the participating NAAs.

In view of the many “remote pilot schools” that are currently mushrooming up in a steadily increasing number of countries, the enormous differences between their training courses, the explosive growth of the number of certified RPAS operators, and the necessity of European harmonisation, there appears to be a urgent requirement to address all facets of remote pilot training, as well as the assessment of remote pilot schools. The recognition of remote pilot certificates (and operator certificates) beyond national borders should be a prime objective for the commercial RPAS community.

Within the context of the aforementioned, and taking into account the limited resources of the majority of the NAAs, as well as the fact that many of the SMEs/SMIs involved with RPAS as manufacturers or operators are relative “new comers” to aviation, the potential role that Qualified Entities (QEs)²⁹ can play should be closely evaluated. In addition, the concept of QE should be better understood by both NAAs and industry. Only then will it be possible for the relevant stakeholders to recognize the very positive role that QEs can play to help open up the commercial RPAS market, without putting an unnecessary burden on NAAs, in terms of finance and personnel. At the time of going to press, only one EU country (UK) had appointed two QEs. However, these QEs are also recognised in Ireland

and The Netherlands. In the near future, QEs are expected to be appointed by the NAAs in three additional EU countries.

Mutual recognition, without requirements for additional certificates, licenses, approvals or other documents relative to RPAS operators, remote pilots and RPAS, would appear to be the necessary way forward, in order to create a level commercial playing field for the European RPAS community, and it would influence investment in the civil RPAS sector in a positive way.

Lastly, it should be recognised that the principal of “regulator proximity”, will be of critical importance for SMIs/SMIs in the EU. In order to reduce the costs of travel and fees, but also for cultural and linguistic reasons, as well as reasons of comprehension, it seems imperative that all certificates, licenses and approvals inside the EU are based on common rules, but that they are issued by the NAAs in their country’s native language.

In order to realise the aforementioned, various stakeholder communities, including RPAS operators (commercial & non-commercial), remote pilot schools, and QEs, have to be federated on a European level. This task is being addressed by the European RPAS Coordination Council, which consists of representatives of the 10 European national RPAS associations (Austria, Belgium, France, Germany, Italy, The Netherlands, Norway, Romania, Spain, UK) and UVS International, in coordination with the national RPAS operator associations in Australia, Botswana, Namibia and South Africa.

Conclusion

The European civil RPAS community has grown substantially and has come a long way in a relatively short period of time, and has now passed through the first of several gates on its way to incremental integration into European airspace. All RPAS stakeholders and particularly the European industrial community, now have to demonstrate understanding and maturity, and jointly work together in order to successfully get through the remaining gates on the road to full RPAS integration. By doing so, they will not only serve themselves, but they will also be doing an immense favour to the global RPAS community, by showing that full RPAS integration is indeed reachable objective.

Explanations

- ¹ UVS International is a non-profit association registered in The Netherlands and operating out of offices in Paris, France which is dedicated to the promotion of RPAS (manufacturers & operators). In addition to having corporate, non-corporate & natural person members, it also federates European national RPAS associations & working groups* in Austria (AAI), Belgium (BeUAS), Denmark (UAS Denmark*), France (FPDC), Germany (UAV-DACH), Italy (ASSORPAS), The Netherlands (DARPAS), Norway (UAS Norway), Romania (UVS Romania), Spain (AERPAS), UK (ARPAS), as well as Australia (ACUO), and South

Africa + Botswana + Namibia (CUAASA). UVS International has a partnership agreement with the European Spatial Data Research Network, a not-for-profit organisation linking national mapping and cadastral agencies with research institutes & universities for the purpose of applied research in spatial data provision, management & delivery. See: www.eurosdnr.net

UVS International represents over 800 companies & organisations. See www.uvs-international.org

- ² Remotely Piloted Aircraft (RPA) & remotely piloted aircraft system (RPAS) - These terms are recommended by ICAO for current use [instead of unmanned aircraft system (UAS), unmanned aerial vehicle (UAV)] - The acronyms are invariant.
- ³ Industry = Enterprise with > 250 employees & annual turnover of > Euro 50 million.
- ⁴ Small & medium-sized enterprise (SME) & Small & medium-sized Industry (SMI) = < 250 employees & annual turnover < Euro 50 million.
- ⁵ Operators = persons, enterprises or organizations engaged in or offering to engage in an RPAS operation.
- ⁶ The NAA of each of the 28 EU member states is responsible in their country for rulemaking, certification & operational approval of civil RPAS with a MTOM < 150 kg, their RPAS flight crews (pilots) & RPAS operators employing the flight crews.
- ⁷ Light RPA = MTOM < 150 kg.
- ⁸ Approved operators = Operators approved by the NAA in countries with regulation in place:
- ⁹ Authorised operators = Operators authorized by the NAA in countries without regulation in place where flight permits are given on a case-to-case basis.
- ¹⁰ Civil RPAS applications (commercial & non-commercial): advertising, agriculture, cinema, forestry, inspection, monitoring, observation, patrol, photogrammetry, photography, survey & mapping, research, search & rescue, training/ instruction.
- ¹¹ Certified operators = A form of official recognition of compliance with the applicable requirements by means of the issuance of a certificate attesting such compliance: Belgium (10), Czech Rep. (24), Denmark (12), Finland (33), France (497), Germany (400 – estimate), Ireland (12), Italy (1), Netherlands [13 (& 30 in the pipeline)], Norway (76), Spain (1), Sweden (216), UK (212).
- ¹² UVS International was a member of the organising committee.
- ¹³ UVS International was a member of the EC UAS Panel.
- ¹⁴ UVS International is a member of the European RPAS Steering Group.
- ¹⁵ Financed RPAS related study contracts - See <http://goo.gl/TP1op8>
- ¹⁶ FRONTEX: European Commission agency responsible for EU border security.
- ¹⁷ EC UAS Panel - See: <http://goo.gl/fg4mHX>
- ¹⁸ JARUS - See: www.jarus-rpas.org
- ¹⁹ JARUS members: Australia, Austria, Belgium, Brazil, Canada, Colombia, Czech Rep., Denmark, Finland, France, Germany, Ireland, Israel, Italy, Malta, Netherlands, Norway, Russian Fed., South Africa, Spain, Switzerland, UK, USA

-
- ²⁰ CS-LURS = Certification Specification for Light Unmanned Rotorcraft Systems - See: <http://goo.gl/Y0RgYW>
- ²¹ AMC RPAS.1309 = Acceptable Means of Compliance for RPAS - System Safety Assessment.
- ²² Participants: DG Connect, DG Research, EASA, ECAC, EUROCONTROL, EDA, SESAR JU, JARUS, EUROCAE, ASD, ECA, EREA, UVS International
- ²³ European RPAS Roadmap - See: <http://goo.gl/oB2kup>
- ²⁴ Regulatory Matters, Strategic Research, Societal Impact.
- ²⁵ SESAR Joint Undertaking Tender (ref. SJU/LC/0101-CFT) - See: www.sesarju.eu/about/procurement
- ²⁶ ‘Towards a European strategy for the development of civil applications of RPAS’ - See: <http://goo.gl/RjGAoj>
- ²⁷ SG01 RPAS Responsibility, Liability & Insurance ToR Definition Team - See: <http://goo.gl/UXI59R>
- ²⁸ SG01 RPAS Responsibility, Liability & Insurance Terms of Reference - See: <http://goo.gl/lsvgTR>
- ²⁹ Qualified Entity: Entity to which a specific certification task has been allocated by the European Aviation Safety Agency (EASA) or a NAA (*and in the latter case is not necessarily compliant with EC Regulation No 216/2008*).

Preface

On December 30, 2013, the U.S. Federal Aviation Administration (FAA) announced the selection of six unmanned aircraft systems (UAS) research and test site operators across the USA that are tasked to study a broad range of issues and challenges related to unmanned aircraft. This announcement was the outcome of a rigorous 10-month selection process involving 25 proposals from 24 states. The selected sites are headed by the University of Alaska, the State of Nevada, New York's Griffiss International Airport, the North Dakota Department of Commerce, Texas A&M University – Corpus Christi, and Virginia Polytechnic Institute and State University. The six test sites can operate until February 13, 2017, with Congress expected to consider extending their operations beyond that date. Furthermore, “The FAA is committed to the safe and efficient integration of UAS into the National Airspace System (NAS), thus enabling this emerging technology to safely achieve its full potential.”

Along the same lines, the International Civil Aviation Organization (ICAO), which is a special agency of the United Nations (UN), has long been committed to “the safe and orderly development of international civil aviation throughout the world,” being the organization that also sets standards and regulations necessary for aviation safety, security, efficiency, and regularity, as well as aviation environmental protection.

Regardless, the December 30, belated and much overdue, announcement from the FAA will perhaps mark the dawn of a new era for unmanned aircraft, worldwide, as it opens the door of civil UAS to be utilized for a wide range of civilian and public domain applications, also outlining actions and considerations that are prerequisite to enabling integration of UAS into NAS. Whether unmanned aircraft are called Unmanned Aerial Vehicles (UAVs) or Remotely Piloted Aircraft (RPA), Unmanned Aircraft Systems (UAS) or Remotely Piloted Aircraft Systems (RPAS), a “domino effect” is soon expected to: accelerate research and development; increase funding; create new jobs; generate provable novel and cutting-edge technologies for increased safety and security of unmanned aircraft; contribute to developing functional and operational “standards” for unmanned aviation; enforce closer collaboration and coordination among national and international interested parties; bring closer academia, government, industry, and the private sector; engage

interested parties in discussions and eventual resolution of ethical, legal, and privacy issues, to say the least and name but a few opportunities.

From that point of view, the *Handbook of Unmanned Aerial Vehicles* (the preferred and most known name at the time this project started) offers a unique opportunity to the interested reader as its main objective is to become a comprehensive reference text for the academic and research communities, industry, manufacturers, users, practitioners, the Federal Government, Federal and State Agencies, and the private sector, as well as all organizations that are and will be using UAS in a wide spectrum of applications. The handbook covers all aspects of UAVs, from design to logistics and ethical issues. The contents of the handbook include material that addresses the needs and *know-how* of all of the above sectors, targeting a very diverse audience. It is also targeting the young investigator, the future inventor and entrepreneur by providing an overview and detailed information of the state of the art as well as useful new concepts that may lead to innovative research.

The handbook offers a unique and comprehensive treatise of everything one needs to know about UAVs/UAS, from conception to operation, from technologies to business activities, users, OEMs, reference sources, conferences, publications, professional societies, etc. It may serve as a thesaurus, an indispensable part of the library of everyone involved in the area. For the first time, contributions by the world's top experts from academia, industry, and the government are brought together to provide unique perspectives on the current state of the art in UAS, as well as future directions. The handbook is intended for the expert/practitioner who seeks specific technical/business information and for the technically oriented scientists and engineers, but also for the novice who wants to learn more about the status of UAS and UAS-related technologies.

The handbook is arranged in a user-friendly format divided into 22 parts: Introduction; UAV Design Principles; UAV Fundamentals; Sensors and Sensing Strategies; UAV Propulsion; UAV Control; UAV Communication Issues; UAV Architectures; UAV Health Management Issues; UAV Modeling, Simulation, Estimation, and Identification; MAVs and Bio-inspired UAVs; UAV Mission and Path Planning; UAV Autonomy; UAV Sense and Avoid Systems; Networked UAVs and UAV Swarms; UAV Integration into the National Airspace; UAV-Human Interfaces and Decision on Support Systems; Human Factors and Training; UAV Logistics Support; UAV Applications; Social and Ethical Implications; and Epilogue, highlighting the economics and future of this rapidly expanding area. Each part contains either original material or enhanced versions of already published work, written by internationally renowned authors who are authorities in their respective fields.

At this stage, it is essential and crucial to state that the reader should not be confused with the different names that have already been used to describe unmanned aircraft, as the context in which such names are used in the handbook is self-explanatory in the respective chapters. For example, the terms UAVs and UAS have distinct definitions and it should be clear why, at first, UAVs became UAS: A UAV refers to a pilotless aircraft, a flying machine without an onboard human pilot or passengers. As such, "unmanned" implies total absence of a human who directs

and actively pilots the aircraft. Control functions for unmanned aircraft may be either on-board or off-board (remote control). The term UAS was introduced by the U.S. Department of Defense, followed by the FAA and the European Aviation Safety Agency (EASA). This term was meant to signify that UAS are aircraft and as such their airworthiness will need to be demonstrated and they are also systems consisting of ground control stations, communication links and launch and retrieval systems in addition to the aircraft itself. Other names include Remotely Piloted Vehicles (RPVs), a term that was used in the Vietnam War, while the U.S. military also calls them Remotely Piloted Aircraft (RPA), a term used to include both the aircraft and the pilot, while the UK and the ICAO have designated them as Remotely Piloted Air (or Aircraft) System (RPAS), to demonstrate the presence of the man-in-the-loop to control them. The ICAO-recommended definition for RPAS is “A set of configurable elements consisting of a remotely-piloted aircraft, its associated remote pilot station(s), the required command and control links and any other system elements as may be required, at any point during flight operation.”

However, despite any terminology issues, one thing is certain: The unmanned aircraft arena is moving forward at a rapid pace. New models, users, relevant technologies and events are finding the light of publicity on a daily basis. Even a massive handbook of this type is not sufficient to capture all aspects and developments in this emerging field. There bound to be omissions and this unique volume will require updates as the technology marches on. It is the editors’ hope that, in collaboration with Springer, subsequent editions will follow with new and updated information as the field continues to rapidly expand, and it is very likely that “standards” will soon be introduced.

We realize that the Handbook is not complete. It is being produced amidst “rapidly evolving developments” in unmanned aviation with new information surfacing on a daily basis. Regardless, we did not want this to be an endless project, thus, we brought closure to it.

We would like to thank wholeheartedly the section editors and the contributing authors who have made this project a reality. We are grateful to the Springer team, headed by Ms. Nathalie Jacobs, who have been patient, very understanding, very cooperative, and easy to work with.

June, 2014

Kimon P. Valavanis, Denver, USA
George J. Vachtsevanos, Atlanta, USA
Editors-in-Chief

About the Editors



Kimon P. Valavanis received the Diploma in Electrical and Electronic Engineering – Diplôme d'Ingénieur (5 years of study) – from the National Technical University of Athens (<http://www.ntua.gr>), Greece, in 1981, and he completed the Professional Engineer (PE) exams in Electrical and Mechanical Engineering in February 1982. He has been a member of the Technical Chamber of Greece since 1982. He received the MS degree in Electrical Engineering and the PhD degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute (RPI) (<http://www.renselaer.edu>) in 1984 and 1986, respectively.

From 1987 to 1990, he held the Analog Devices Career Development Chair for Assistant Professors in the Department of Electrical and Computer Engineering (ECE), Northeastern University (<http://www.northeastern.edu>), Boston, where he was also Director of the Robotics Laboratory. From 1991 to 1999, he was with the Center for Advanced Computer Studies (CACS), University of Louisiana, at Lafayette (<http://www.cacs.louisiana.edu>), where he served as Associate Professor from 1991 to 1995 and as Professor of Computer Engineering since 1995. He was also Associate Director for Research at the A-CIM Center (1993–1999) and Director of the Robotics and Automation Laboratory. He also held the A-CIM/[TC]²/Regents Professorship in Manufacturing.

From 1999 to 2003, he was Professor in the Department of Production Engineering and Management (DPEM), Technical University of Crete (TUC) (<http://www.tuc.gr>), Greece, where he also established and served as Founding Director of the Intelligent Systems and Robotics Laboratory (ISRL). He was also Director of the DPEM Graduate Program and Chair of the TUC Industrial Advisory Board (IAB) and Technical Council.

From August 2003 to August 2008, he was Professor in the Department of Computer Science and Engineering (CSE), University of South Florida (<http://www.cse.usf.edu>), where he also served as Deputy Director of the Center for Robot-Assisted Search and Rescue (CRASAR) until the summer of 2005. In late 2005, he established the Unmanned Systems Laboratory (USL) in the College of Engineering, in which he served as Founding Director. He was also the Managing Director of the National Institute for Applied Computational Intelligence (NIACI) and a Faculty Associate at the Center for Urban Transportation Research (CUTR). From 2004 to March 2006, he was also the Managing Director of the American Foundation for Greek Language and Culture (AFGLC) (<http://www.afglc.org>).

He joined the University of Denver on September 1, 2008, as Chair of the ECE Department, where he also is the John Evans Professor. He also served as Interim/Acting Chair of the Computer Science (CS) Department for 2 years (July 2009 to June 2011). He is also the Founding Director of the DU Unmanned Systems Laboratory (DU²SL), which he established in early 2009. In November 2012, he officially established the DU Unmanned Systems Research Institute (DU²SRI), in which he serves as Founding Director.

Further, since the mid-1990s, he has been Guest Professor in the Faculty of Electrical Engineering and Computing, Department of Telecommunications, University of Zagreb (<http://www.fer.hr>), Croatia. He was also invited through the European Union PhD Research Program in Italy to teach Robotics and Unmanned Systems at the Dipartimento di Ingegneria Informatica, Gestionale e dell' Automazione, Università Politecnica delle Marche, Ancona, Italy.

During his career thus far, Dr. Valavanis has graduated 34 PhD students and more than 80 MS students. He has attracted more than \$35 million in research funds from federal and state agencies, industry, and the private sector, while in Europe he was funded by the Greek Secretariat of Research and Technology, the European Union, industry, and from the Croatian Ministry of Science and Technology (joint projects).

Dr. Valavanis' research interests are focused in the areas of unmanned systems, distributed intelligence systems, robotics, and automation. He has published over 350 book chapters, technical journals/transactions, refereed conference papers, invited papers, and technical reports. He has authored, coauthored, and/or edited the following books: *Intelligent Robotic Systems: Theory, Design and Applications* (with Dr. G. N. Saridis), Kluwer Academic Publishers, 1992; *Control Problems in Robotics and Automation* (coedited with B. Siciliano), Lecture Notes in Control and Information Sciences, Vol. 230, Springer-Verlag, 1998; *Intelligent Manufacturing Systems: Programming and Control* (with J. Balic, N. Tsourveloudis, and S. Ioannidis), University of Maribor Publications, 2003; *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy* (editor and coauthor

of nine chapters), Springer, 2007; *Unmanned Aircraft Systems – International Symposium on Unmanned Aerial Vehicles, UAV’08*, (coedited with P. Oh and L. A. Piegl), Springer, 2009; *On Integrating Unmanned Aircraft Systems into the National Airspace System: Issues, Challenges, Operational Restrictions, Certification, and Recommendations* (with K. Dalamagkidis and L. A. Piegl), Springer, 2009, with a second edition published in 2012; *Intelligent Control Applications to Engineering Systems* (editor), Springer, 2009; *Unmanned Aircraft Systems – 2nd International Symposium on Unmanned Aerial Vehicles, UAV’09* (coedited with R. Beard, P. Oh, A. Ollero, L. Piegl, and H. D. Shim), Springer, 2010; *Unmanned Aerial Vehicles* (editor), Springer, 2011; *Linear and Nonlinear Control of Small-Scale Unmanned Helicopters* (coauthored with I. A. Raptis), Springer, 2012; *Recent Developments in Unmanned Systems* (editor), Springer, 2012; and *Unmanned Aircraft Systems: Challenges and state of the art* (editor), Springer, 2013. In addition, he is the coauthor (with G. Atsalakis and K. Zopounidis) of the book *Stock Market Forecasting Techniques* (in Greek), Klidarithmos, 2008. He also holds several patents.

Dr. Valavanis has organized two International Advanced Robotics Programme (IARP) meetings in Lisbon, Portugal, and Lafayette, LA, which were funded by the National Science Foundation. He has taught tutorial workshops at the IEEE CDC, ACC, and ICRA, at the Conference on Telecommunications (ConTel), and at the Mediterranean Conference on Control and Automation. He served as Associate Editor of the IEEE Transactions in Robotics and Automation from February 1996 to February 1999, as the Robotics and Automation Society “Discrete Event Dynamic Systems Technical Committee” Co-chair for 2 years, and as an Associate Editor of the *IEEE Robotics & Automation Magazine* from 1994 to 1995. He was Editor in Chief of the same magazine for 10 years (1996 to 2005). He was also the Book Review Editor of the *Journal of Intelligent and Robotic Systems (JINT)* until 2006, and since then he serves as the Editor in Chief of JINT. He serves on the Editorial Advisory Board of the International Series on Microprocessor-Based and Intelligent Systems Engineering published by Springer. He also served as a member of the IEEE Robotics and Automation Society Awards Committee for 3 years and as Cochair/Chair of the Aerial Robotics and Unmanned Aerial Vehicles Technical Committee (2008–2012).

Dr. Valavanis has also received invitations by the Hellenic Quality Assurance Agency (HQAA) for Higher Education of the Hellenic Republic to evaluate Greek universities (ABET Evaluation) and by the European Union to be an evaluator of IP and STREP projects as part of the 7th European Union (EU) Framework Programme for Research and Technology Development (FP7) – Information and Communication Technologies (ICT-2009.2.1 Cognitive Systems and Robotics), and he also served as a member of the Hearing Committee panel in Luxemburg. He is an appointed Scientific Project Reviewer in projects financed by the Ministry of Science, Education and Sports of the Republic of Croatia and in research projects evaluated by the Italian Evaluation of Research Quality exercise (VQR 2004-2010).

Dr. Valavanis has been on the organizing committee of many conferences, serving as General, Program, Registration, as well as Local Arrangements Chair, including Registration Chair of the 36th IEEE CDC; Local Arrangements Chair of

the 34th IEEE CDC; General Chair (with F. Lewis) of the 11th Mediterranean Conference on Control and Automation, June 2003; Program Chair of the 2004 IEEE ICRA; General Chair (with P. Antsaklis) of the 15th Mediterranean Conference on Control and Automation, June 2007; General Chair (with W. Gruver) of the IEEE SMC International Conference on Distributed Human-Machine Systems, March 2008; General Chair of the 2011 IEEE Multi-Conference on Systems and Control, September 2011; and General Chair (with P. Antsaklis) of the 21st Mediterranean Conference on Control and Automation, June 2013. He has also served as General Chair of the International Conference on Unmanned Aircraft Systems (ICUAS), which he created in 2008. In addition, he is the Founding President of the ICUAS Association, Inc., a nonprofit organization that organizes and sponsors the annual ICUAS technical meeting. In 1998, he was elected Vice President – Administration of the IEEE Mediterranean Control Association (MCA). He was a Distinguished Speaker in the IEEE Robotics and Automation Society (–2003), a Senior Member of IEEE, and a Fellow of the American Association for the Advancement of Science. He is also a Fulbright Scholar.



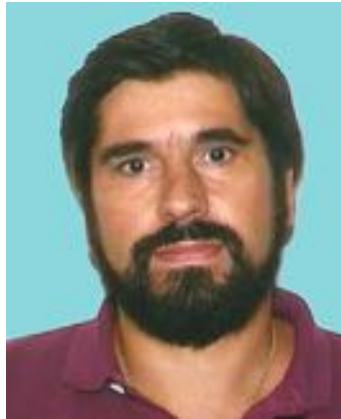
George J. Vachtsevanos is currently serving as Professor Emeritus at the Georgia Institute of Technology. He served as Professor of Electrical and Computer Engineering at Georgia Tech from 1984 until September 2007. He also served on a part-time basis as the Chief Scientist of Impact Technologies, LLC. Dr. Vachtsevanos directs at Georgia Tech the Intelligent Control Systems Laboratory where faculty and students are conducting interdisciplinary research in intelligent control, fault diagnosis and failure prognosis of complex dynamical systems with emphasis on rotorcraft, and hierarchical/intelligent control of unmanned aerial vehicles. His work in unmanned aerial vehicles dates back to 1994 with major projects funded by the U.S. Army and DARPA. He has served as the Co-PI for DARPA's Software Enabled Control program and directed the development and flight testing of novel fault-tolerant control algorithms for unmanned aerial vehicles. He has represented Georgia Tech at DARPA's HURT program where multiple UAVs performed intelligence, surveillance, reconnaissance, and tracking missions in an urban environment. Under AFOSR sponsorship, his research team has been developing a biologically inspired micro aerial vehicle. Under ARO sponsorship, his team has been developing game-theoretic and reasoning strategies for target recognition and tracking with multiple UAVs. Under support by NASA, his research team has been developing and testing an autonomous hovercraft. This research effort is aiming at the design and operation of unmanned systems that possess improved attributes of autonomy and fault tolerance. Dr. Vachtsevanos and his research team were involved in a series of programs since 1985 in diagnostics and more recently in prognostics funded by the government and industry. His research work has been supported over the years by ONR, NSWC, the MURI Integrated Diagnostic program at Georgia Tech, the U.S. Army's Advanced Diagnostic program, General Dynamics, General Motors Corporation, the Academic Consortium for Aging Aircraft program, the U.S. Air Force Space Command, Bell Helicopter, and Fairchild Controls, among others. The innovative technologies have relied on

both data-driven and model-based algorithms from the domains of soft computing, Dempster-Shafer theory, and Bayesian estimation techniques with emphasis on particle filtering and physics-based modeling architectures. The application domains range from UAVs to automotive electrical storage and distribution systems, high-power amplifiers, environmental control systems, and critical engine and drive system aircraft components. Dr. Vachtsevanos has developed and has been administering an intensive four-day short course on “Fault Diagnostics/Prognostics for Equipment Reliability and Health Maintenance.” He is a member of the board of directors of the Prognostics and Health Management Society. He has published over 300 technical papers and is the recipient of the 2002–2003 Georgia Tech School of ECE Distinguished Professor Award and the 2003–2004 Georgia Institute of Technology Outstanding Interdisciplinary Activities Award. His research team has been awarded numerous awards for its technical presentations at professional conferences. He is the lead author of the book *Intelligent Fault Diagnosis and Prognosis for Engineering Systems* published by Wiley in 2006.

Section Editors



Randal W. Beard received his BS degree in electrical engineering from the University of Utah, Salt Lake City, in 1991; MS degree in electrical engineering in 1993; MS degree in mathematics in 1994; and PhD degree in electrical engineering in 1995, all from Rensselaer Polytechnic Institute, Troy, NY. Since 1996, he has been with the Electrical and Computer Engineering Department at Brigham Young University, Provo, UT, where he is currently professor. In 1997 and 1998, he was a Summer Faculty fellow at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. In 2006–2007, he was a National Council Research fellow at the Air Force Research Laboratory, Munitions Directorate, Eglin AFB, FL. His research interests include guidance and control, autonomous systems, and multiple-vehicle coordination and control with particular emphasis on small and miniature air vehicles.



Christos G. Cassandras is head of the Division of Systems Engineering and professor of electrical and computer engineering at Boston University. He is also the cofounder of Boston University's Center for Information and Systems Engineering (CISE). He received his degrees from Yale University, Stanford University, and Harvard University. In 1982–1984, he was with ITP Boston, Inc. In 1984–1996, he was a faculty member at the University of Massachusetts, Amherst. He specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, computer simulation, and applications to computer and sensor networks, manufacturing systems, and transportation systems. He has published over 300 refereed papers in these areas and five books. He has collaborated with the MathWorks, Inc., in the development of the discrete event and hybrid system simulator SimEvents. Dr. Cassandras was editor-in-chief of the *IEEE Transactions on Automatic Control* from 1998 to 2009 and has served on several journal editorial boards. He was the 2012 president-elect of the IEEE Control Systems Society (CSS) and has served as vice president for publications. He has been a plenary speaker at various international conferences and an IEEE distinguished lecturer (2001–2004). He is the recipient of several awards, including the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook) for *Discrete Event Systems: Modeling and Performance Analysis*, a 2011 prize for the IBM/IEEE Smarter Planet Challenge competition, a 1991 Lilly Fellowship, and a 2012 Kern Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi. He is also a fellow of the IEEE and a fellow of the IFAC.



David A. Castañón is professor and chair of the Department of Electrical and Computer Engineering at Boston University. He received his BS degree in electrical engineering from Tulane University and his PhD in applied mathematics from MIT in 1976. He was a research scientist in LIDS at MIT from 1976 to 1981 and then chief scientist with AlphaTech, Inc., until joining Boston University in 1990. At Boston University, he is codirector of the Center for Information and Systems Engineering, deputy director of the Gordon-CenSSIS NSF Engineering Research Center on Subsurface Sensing and Imaging, and associate director for the Department of Homeland Security Center of Excellence on Explosives. He has served in numerous positions for the IEEE Control Systems Society, including president in 2008, and served on the Air Force Scientific Advisory Board. His research includes stochastic control, estimation, game theory, and optimization, with applications to cooperative control.



Konstantinos Dalamagkidis has been research associate at the Technische Universität München since June 2010. He holds a PhD in computer science and engineering from the University of South Florida (2009), and before his current appointment, he was a postdoctoral research associate at the University of Denver. He is currently the coordinator of the EU FP7-ICT Myrobotics project and has been involved in different capacities in several research projects in the EU and the USA.

In the past, he was also employed as an instructor at the Technological Educational Institute of Crete in Greece. He is the author of the first book on the integration of unmanned aircraft systems into the national airspace, as well as a number of peer-reviewed conference and journal publications. His research interests include mobile robotics and in particular modeling, simulation, control, and planning topics, especially as they relate to unmanned aircraft safety and reliability.



Graham Drozeski is a program manager and chief engineer for advanced aircraft development programs at Aurora Flight Sciences. His current programs target improved autonomy for vertical takeoff and landing platforms and extended reach for ship-launched unmanned systems. Dr. Drozeski specializes in flight technologies including flight control, simulation, handling qualities, vehicle management, mission management, and human–systems interaction. Formerly, Dr. Drozeski was Karem Aircraft’s Director of Flight Technologies and served as program manager for the Karem Aircraft–Lockheed Martin team on the Joint Heavy Lift (JHL) program. He holds a doctorate from Georgia Tech for his development of novel control technologies for unmanned aerial vehicles. Graham is a former US Army aviator. While a company commander, he oversaw operational testing of the first UH-60 A2C2S (Army Airborne Command and Control System).



Eric W. Frew is an associate professor in the Aerospace Engineering Sciences Department and the director of the Research and Engineering Center for Unmanned Vehicles (RECUV) at the University of Colorado Boulder (CU). He received his BS in mechanical engineering from Cornell University in 1995 and his MS and PhD in aeronautics and astronautics from Stanford University in 1996 and 2003, respectively. Dr. Frew's research efforts focus on autonomous flight of heterogeneous unmanned aircraft systems, optimal distributed sensing by mobile robots, controlled mobility in ad hoc sensor networks, miniature self-deploying systems, and guidance and control of unmanned aircraft in complex atmospheric phenomena. He is currently the CU site director for the National Science Foundation Industry/University Cooperative Research Center (I/UCRC) for Unmanned Aircraft Systems. He received his NSF Faculty Early Career Development (CAREER) Award in 2009 and was selected for the 2010 DARPA Computer Science Study Group.



Kai Goebel is a deputy area lead of the Discovery and Systems Health Technology Area at NASA Ames Research Center. He also coordinates the Prognostics Center of Excellence and is the technical lead for Prognostics and Decision Making in NASA's System-wide Safety and Assurance Technologies Project. Prior to joining NASA in 2006, he was a senior research scientist at General Electric Corporate Research and Development Center since 1997. Dr. Goebel received his PhD at the University of California at Berkeley in 1996. He has carried out applied research in the areas of real-time monitoring, diagnostics, and prognostics, and he has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems. He holds 17 patents and has coauthored more than 250 technical papers in the field of systems health management. Dr. Goebel was an adjunct professor of the CS Department at Rensselaer Polytechnic Institute (RPI), Troy, NY, between 1998 and 2005 where he taught classes in Soft Computing and Applied Intelligent Reasoning Systems. He has been the co-advisor of 6 PhD students. Dr. Goebel is a member of several professional societies, including ASME, AAAI, AIAA, IEEE, VDI, SAE, and ISO. He was the general chair of the Annual Conference of the PHM Society, 2009, has given numerous invited and keynote talks, and held many chair positions at the PHM conference and the AAAI Annual meetings series. He is currently member of the board of directors of the PHM Society and associate editor of the International Journal of PHM.



Michael A. Goodrich received his PhD at Brigham Young University (BYU) in 1996 from the Electrical and Computer Engineering Department. Following graduation, he completed a two-year postdoctoral research associate position at Nissan CBR in Cambridge, Massachusetts. He is currently a professor within the Computer Science Department at BYU. He is the director of the Human Centered Machine Intelligence (HCMI) laboratory with research focused on human-robot interaction, multi-agent learning, artificial intelligence, and decision theory. He has published extensively in the field of human-robot interaction and has been heavily involved in serving in that community as conference chair, as steering committee chair, and currently as managing editor of the *Journal of Human-Robot Interaction*.



Jonathan P. How is the Richard C. Maclaurin professor of aeronautics and astronautics at the Massachusetts Institute of Technology. He received his BASc from the University of Toronto in 1987 and his SM and PhD in aeronautics and astronautics from MIT in 1990 and 1993, respectively. He then studied for two years at MIT as a postdoctoral associate for the Middeck Active Control Experiment (MACE) that flew onboard the Space Shuttle Endeavour in March 1995. Prior to joining MIT in 2000, he was an assistant professor in the Department of Aeronautics and Astronautics at Stanford University. He has graduated a total of 30 PhD students while at MIT and Stanford University on topics related to GPS navigation, multi-vehicle planning, and robust/hybrid control. He has published more than 290 articles in peer-reviewed proceedings and 84 papers in technical journals. His current research interests include (1) design and implementation of distributed robust planning algorithms to coordinate multiple autonomous vehicles in dynamic uncertain environments; (2) spacecraft navigation, control, and autonomy, including GPS sensing for formation-flying vehicles; and (3) adaptive flight control to enable autonomous agile flight and aerobatics. Professor How was the planning and control lead for the MIT DARPA Urban Challenge team that placed fourth in the recent race at Victorville, CA. He was the recipient of the 2002 Institute of Navigation Burka Award, recipient of the AIAA Best Paper Award in the Guidance Navigation and Control Conference in 2011 and 2012, and recipient of the 2011 IFAC Automatica Award for best applications paper and is an associate fellow of AIAA and a senior member of IEEE.



Isaac Kaminer received his PhD in electrical engineering systems from the University of Michigan in 1992. Before that, he spent four years working at Boeing Commercial first as a control engineer in 757/767/747-400 Flight Management Computer Group and then as an engineer in Flight Control Research Group. Since 1992, he has been with the Naval Postgraduate School first at the Aeronautics and Astronautics Department and currently at the Department of Mechanical and Aerospace Engineering where he is a professor. He has a total of over 20 years of experience in the development and flight testing of guidance, navigation, and control algorithms for both manned and unmanned aircraft. His more recent efforts were focused on the development of coordinated control strategies for multiple UAVs and vision-based guidance laws for a single UAV. Professor Kaminer has coauthored more than a hundred refereed publications. Over the years, his research has been supported by ONR, NASA, US Army, NAVAIR, and USSOCOM.



George Lucas is Class of 1984 Distinguished Chair in Ethics in the Vice Admiral James B. Stockdale Center for Ethical Leadership at the United States Naval Academy (Annapolis) and professor of Ethics and Public Policy at the Graduate School of Public Policy at the Naval Postgraduate School (Monterey, CA). He has taught at Georgetown University, Emory University, Randolph-Macon College, and the Catholic University of Louvain, Belgium, and served as Philosophy Department Chairman at Santa Clara University in California. He has received research fellowships from the Fulbright Commission and the American Council of Learned Societies and has served three times (in 1986, 1990, and 2004) as the director of National Endowment for the Humanities Summer Institutes for College and University Faculty. Dr. Lucas is the author of five books, more than fifty journal articles, translations, and book reviews and has also edited eight book-length collections of articles in philosophy and ethics. Among these titles are *New Warriors/New Weapons: Ethics and Emerging Military Technologies* [a special issue of the *Journal of Military Ethics* 9/4 (2010)]; *Anthropologists in Arms: The Ethics of Military Anthropology* (AltaMira Press, 2009); *Perspectives on Humanitarian Military Intervention* (University of California Press, 2001); *Lifeboat Ethics: The Moral Dilemmas of World Hunger* (Harper & Row, 1976); and *Poverty, Justice, and the Law: Essays on Needs, Rights, and Obligations* (UPA, 1986). His essays and peer-reviewed scholarly articles have appeared in *The Journal of Philosophy*, the *Journal of the History of Philosophy*, the *Journal of Military Ethics*, *Ethics and International Affairs*, the *Journal of National Security Law & Policy*, the *Case-Western Review of International Law*, the *Columbia Review of Law and Technology*, and the *International Journal of Applied Ethics*. Dr. Lucas is also coeditor (with Capt. Rick Rubel, US Navy, retired) of the textbook, *Ethics and the Military Profession: the Moral Foundations of Leadership*, and a companion volume, *Case Studies in Military Ethics*, both published by Longman/Pearson Education (New York, 2004; 3rd edition 2010). These texts are used in core courses devoted to ethical leadership at the United States Naval Academy, the United States Air Force Academy, and at Naval ROTC units at over 57 colleges and universities throughout the nation.



Robert C. Michelson is an American engineer and academic widely known for inventing the Entomopter, a biologically inspired flapping-winged aerial robot, and for having established the International Aerial Robotics Competition. He has received degrees in electrical engineering from Virginia Polytechnic Institute and Georgia Institute of Technology. He is the president of Millennial Vision, LLC and SEPDAC Inc., he is Principal Research Engineer Emeritus at the Georgia Tech Research Institute, and adjunct associate professor (Ret.) in the School of Aerospace Engineering at the Georgia Institute of Technology. He is the author of three US patents and over 87 journal papers, book chapters, and reports. Michelson also developed classes in avionics and taught in the School of Aerospace Engineering at the Georgia Institute of Technology and as a Visiting Technology professor in six other nations. Michelson is an associate fellow of the American Institute of Aeronautics and Astronautics (AIAA), senior member of the Institute of Electrical and Electronics Engineers (IEEE), and a full member of the Scientific Research Society of North America, Sigma Xi. During the 1990s, he served as president and member of the board of directors of the Association for Unmanned Vehicle Systems (AUVSI) International organization. He is the recipient of the 1998 AUVSI Pioneer Award and the 2001 Pirelli Award for the diffusion of scientific culture as well as the first Top Pirelli Prize.



Nafiz Alemdaroğlu received his Bsc. degree from the Mechanical Engineering Department of Middle East Technical University (METU) in 1972. He got his masters and Ph degrees from The Engineering School in Poitiers France (Ecole Nationale Supérieure de Mécanique et D'Aérotechnique, ENSMA), in 1975 and 1979 respectively.

After completing his PhD studies he returned to METU and joined the faculty of ME department in 1981. He was promoted to associate professor in 1984. Same year he joined the California State University in Long Beach (CSULB) as a visiting professor where he worked first in the Mechanical Engineering Department and then he helped the founding of the new Aerospace Engineering Department. In 1991 he returned back to METU but this time to the Department of Aeronautical Engineering. In 1992, he was promoted to full professor in the same department. Since then, he is a faculty member of the Aerospace Engineering Department. Between 1997 and 2006, he served as the chairman of the Aerospace Engineering Department.

Prof. Alemdaroğlu served as the Turkish panel member for the Flight Vehicle Integration (FVP) panel of NATO-AGARD between 1994 and 1998. After the establishment of the Research and Technology Organization (RTO) of NATO in 1998, he served as a panel member in the Systems Concepts and Integration Panel (SCI) of this organization. Between 2000 and 2004 he was the chairman of the Systems Concepts and Integration Panel. In 2010 he was assigned as the Turkish national board member for NATO Research and Technology Organization (RTO). He is still a board member in the same organization. However, the name of the organization is changed to NATO, Science and Technology Organization (STO). Prof. Alemdaroğlu also served and is still serving in the Flight Test Technology Team (FTTT) of the SCI panel since 1995.

Between 1991 and 2010 Prof Alemdaroğlu was consultant to Roketsan Inc., Turkish Missile Industries Company, and contributed significantly to the development of various missile systems.

Prof. Alemdaroğlu teaches courses related to aerodynamics, aeronautical design, heat transfer, fluid dynamics, boundary layer theory, experimental aerodynamics, and turbulent flows.

Prof. Alemdaroglu is heavily involved with research in Unmanned Aerial Systems. Since 2004 he has developed three Unmanned Aerial Systems, two man portable/man launchable mini UAS and one tactical-level UAS. He has been working with ASELSAN Inc., the leading Turkish Military Electronics Company since 2010, for the development of mini and tactical-level unmanned aerial systems.



Paul Oh is the Lincy Professor of Unmanned Aerial Systems in the Mechanical Engineering Department at the University of Nevada, Las Vegas (UNLV). Formerly, from 2000 to 2014, he served as a mechanical engineering professor at Drexel University in Philadelphia and both founded and directed the Drexel Autonomous Systems Laboratory (DASL). He received his Mechanical Engineering degrees from McGill (B.Eng. 1989), Seoul National (MSc 1992), and Columbia (PhD 1999). Honors he received include faculty fellowships at NASA Jet Propulsion Lab (2002) and Naval Research Lab (2003), the NSF CAREER Award (2004), and the SAE Ralph Teetor Award for Engineering Education Excellence (2005). He was named a Boeing Welliver fellow (2006), elected as ASME fellow (2011), and appointed as Senior Faculty Fellow (2013) at the Naval Seas Systems Command (NAVSEA) for the Office of Naval Research (ONR). He was also the founding chair of the IEEE Technical Committee on Aerial Robotics and UAVs. He served for two years (2008–2010) at the National Science Foundation (NSF) as the program director managing the robotics research portfolio.



Aníbal Ollero (<http://grvc.us.es/aollero>) is full professor and head of the GRVC Group (55 researchers) at the University of Seville and scientific advisor of the Center for Advanced Aerospace Technologies in Seville. He has been full professor at the Universities of Santiago de Compostela and Malaga in Spain and researcher at Carnegie Mellon University (Pittsburgh, USA) and LAAS-CNRS (Toulouse, France). He is author/coauthor of 460 publications, including 9 books and 110 papers in SCI journals. He led about 120 projects and is currently the coordinator of the ARCAS and EC-SAFEMOBIL FP7 Integrated Projects of the European Commission and associated coordinator of FP7 PLANET and CONET Network of Excellence. He is the recipient of 13 national and international awards for his R&D activities including the second 2010 EUROP-EURON Technology Transfer Award and the IV Javier Benjumea Award by Scientific Excellence and Social Impact. He has been advisor of 25 PhD thesis.



Daniel J. Pack is Mary Clark professor and chair of the Electrical and Computer Engineering Department at the University of Texas, San Antonio. He was a professor of electrical and computer engineering and director of the Academy Center for Unmanned Aircraft Systems Research at the United States Air Force Academy, CO. He currently holds the Professor Emeritus position at the Academy. He received his Bachelor of Science degree in electrical engineering, the Master of Science degree in engineering sciences, and the PhD degree in electrical engineering from the Arizona State University, the Harvard University, and the Purdue University, respectively. He was a visiting scholar at the Massachusetts Institute of Technology-Lincoln Laboratory. Dr. Pack has coauthored six textbooks on embedded systems (including 68HC12 Microcontroller: Theory and Applications and Embedded Systems: Design and Applications with the 68HC12 and HCS12) and published over 120 book chapters, technical journal/transactions articles, and conference papers on unmanned systems, cooperative control, robotics, pattern recognition, and engineering education. He is the recipient of a number of teaching and research awards including the Carnegie US Professor of the Year Award, the Frank J. Seiler Research Excellence Award, the Tau Beta Pi Outstanding Professor Award, the Academy Educator Award, and the Magoon Award. He is a member of Eta Kappa Nu (Electrical Engineering Honorary), Tau Beta Pi (Engineering Honorary), IEEE (senior member), and the American Society of Engineering Education. He is a registered professional engineer in Colorado. His research interests include unmanned aerial vehicles, intelligent control, automatic target recognition, and robotics.



Fulvia B. Quagliotti took the degree of doctor in Aeronautical Engineering at the Politecnico di Torino (Torino Technical State University), Italy. Then she was consultant for development of European space programs, at the Italian Delegation at OCSE, in Paris. Then she was at FIAT Aviation Division (later AERITALIA and now ALENIAAERMACCHI), Space Division, taking part in several programs: the development of the Italian satellite SIRIO, the project of the thermal shields for the European launcher Europa III, the GEOS Satellite program, the METEOSAT program, as program manager for AERITALIA and, finally, the first development of the SPACE LAB program, responsible for project control.

Since November 1974 she is staff member at Politecnico di Torino, where she is presently an associate professor of Flight Mechanics, in the Department of Mechanical and Aerospace Engineering (formerly Aerospace Engineering Department). She is responsible of several research activities in the field of theoretical and experimental Flight Mechanics and experimental Aerodynamics, Flight Simulation, Flight Control and design of Micro and Mini aerial vehicles (she is holding a patent for the configuration). She joined the AGARD Flight Mechanics Panel in 1992 till 1997, when NATORTO (Research and Technology Organization), now NATO-STO (Scientific and Technology Organization), was stated. In those organizations she is the Italian Academia representative in the Applied Vehicle Technology Panel. Previously she represented the Italian Academia in the System Concept Integration Panel, where she served as vice-chairman of the Panel and chairman of the Program Committee for seven years. Within the AVT Panel, she is a member of the Strategic Committee and panel mentor for a Task Group on UASs.

She is an international editor of Journal of Aircraft and an associate fellow of AIAA, senior editor for Springer, member of the editorial board of Automatic Control in Aerospace – online journal.



Michael J. Roemer is the cofounder of Impact Technologies, now part of Sikorsky Aircraft, with over 20 years experience in the development of automated health monitoring, diagnostic and prognostic systems for a wide range of military and commercial applications. He is currently the Sikorsky Technical fellow for Prognostics and Health Management (PHM). His experience includes a wide range of integrated vehicle health management system implementations to detect and predict system faults in real time and perform automated troubleshooting and maintenance planning. He is the cofounder and past vice president of the PHM Society, chairman of the SAE HM-1 Integrated Vehicle Health Management Committee, board member and past chairman of the Machinery Failure Prevention Technology (MFPT) Society, Member of the IGTI Marine Committee and ASME Controls and Diagnostics Committee and Adjunct Professor of Mechanical Engineering at the Rochester Institute of Technology. He is the coauthor of a recent book titled “Intelligent Fault Diagnosis and Prognosis for Engineering Systems” and has written or coauthored more than 100 technical papers related to integrated systems health management.



David Hyunchul Shim received his BS and MS degrees in Mechanical Design and Production Engineering from Seoul National University, Seoul, Korea, and the PhD degree in Mechanical Engineering from the University of California, Berkeley, in 1991, 1993, and 2000, respectively. From 1993 to 1994, he was with Hyundai Motor Company, Korea, as transmission design engineer. From 2001 to 2005, he was with Maxtor Corporation, California, USA, as staff servo engineer. From 2005 to 2007, he was with UC Berkeley as principal engineer, in charge of the Berkeley Aerobot Team, working on advanced UAV research. In 2007, he joined the Department of Aerospace Engineering at KAIST, South Korea, as assistant professor. He is now associate professor and also director of KAIST Field Robotics Research Center, working on various research projects on unmanned aerial and ground vehicles in the area of novel platform development and control, advanced navigation, and multi-agent coordination.



L.J.P. (Lennaert) Speijker is employed as senior scientist at the Air Transport Safety Institute of NLR, where he started in 1995. He focuses on aviation safety and certification research and consultancy for EC, EASA, EUROCONTROL, ANSPs, manufacturers, and regulators. He is currently managing the EC Project ASCOS (Aviation Safety and Certification of new Operations and Systems). Within a multi-year cooperation agreement between FAA and CAA Netherlands, he worked on safe integration of RPAS in non-segregated airspace. His RPAS research includes safety risk management, safety risk criteria, safety methods, research on detect and avoid systems, and risk analysis of command and control. For EASA, he investigated the use of detect and avoid systems to improve “See and Avoid” for manned general aviation. He is a member of Joint Authorities for Rulemaking on Unmanned Systems (JARUS, on behalf of CAA Netherlands) and the European General Aviation Safety Team (EGAST).



Richard S. Stansbury is an associate professor of computer engineering and computer science at Embry-Riddle Aeronautical University (ERAU) in Daytona Beach, FL. He is the coordinator for the Master of Science in Unmanned and Autonomous Systems Engineering program, a multidisciplinary engineering program focused upon air, ground, and sea-based robotic systems. He currently teaches Introduction to Artificial Intelligence and Data Structures and Algorithms courses. From 2007 to 2013, he co-taught the capstone senior design course for electrical engineering, computer engineering, and software engineering students.

As a graduate student at the University of Kansas, where he received his BS and MS in computer engineering and his PhD in computer science, Dr. Stansbury developed autonomous ground vehicles for operation in Greenland and Antarctica. His research as part of the NSF Center for Remote Sensing of Ice Sheets supported the autonomous surveys of radar data to measure properties of the polar ice sheets. He and his two prototype vehicles MARVIN I and II traveled to Greenland and Antarctica for field exercises.

At ERAU, his research has transitioned from ground robots to unmanned aircraft systems (UAS). His early UAS research funded by the FAA focused upon the survey of current and near-term future UAS technologies and their regulatory impact (through a regulatory gap analysis). This research has been continued under a grant from NASA. Starting in 2010, Dr. Stansbury began development as part of a multidisciplinary team on an unmanned aircraft for NOAA to provide in situ measurement of atmospheric conditions within tropical cyclones. This unmanned aircraft, Gale, is expected to be test flown in 2012.

Dr. Stansbury is involved in a variety of professional service activities in support of UAS research. He advises ERAU's Team Blackbird at the AUVSI Student UAS competition. He is a member of the AUVSI UAS Advocacy Committee and the AUVSI 2013 Unmanned Systems Conference Program Committee. He is also a member of the AIAA Unmanned Systems Program Committee.



Salah Sukkarieh received his BE in mechatronics in 1997 and PhD in robotics in 2000 at the University of Sydney. He is the director of Research and Innovation at the Australian Centre for Field Robotics and professor of Robotics and Intelligent Systems at The University of Sydney. His interests in UAS are on data fusion techniques for mapping, tracking and localization, and in the area of cooperative perception and decision making. He was the project and research lead on a number of grants in the application of UAS in security and the environment, including funding from BAE Systems, US Air Force, Land and Water Australia, Meat and Livestock Australia, US Office of Naval Research, Ministry of Defense UK, the Department of Agriculture, Forestry and Fishery, and the Australian Research Council.



Ben Upcroft received his undergraduate degree (with honors) in science and his PhD degree in Ultracold Atomic Physics from the University of Queensland, Australia, in 2003. He subsequently joined the Australian Research Council (ARC) Centre of Excellence for Autonomous Systems, The University of Sydney, Sydney, N.S.W., Australia, as a postdoctoral fellow where he was engaged in the field of robotics. He then moved to the University of Queensland as a senior lecturer in mechatronics and focused on computer vision-aided localization and navigation for autonomous ground and aerial vehicles. He is now a senior lecturer of mechatronics at the Queensland University of Technology, Brisbane, QLD, Australia, and has participated in major industrial projects involving autonomous aircraft, off-road and urban vehicles, and network communications. His current research interests include computer vision for navigation and environment classification applied to mobile robots.



Lora Weiss is a lab chief scientist and technical director of Autonomous Systems at the Georgia Tech Research Institute. Her research focus is on the design, development, and implementation of technologies for unmanned and autonomous systems. She has conducted research on unmanned systems spanning the domains of air, ground, sea-surface, and undersea. She has chaired technical sessions at professional conferences and has provided over 300 technical presentations. Dr. Weiss was on the board of directors for the Association of Unmanned Vehicle Systems International (AUVSI), which is the world's largest unmanned systems organization. She was on the Technical Advisory Board of the Robotics Technology Consortium, and she was an executive board member of the National Defense Industrial Association. She currently chairs the ASTM Committee on Unmanned Maritime Vehicle Autonomy and Control and is a member of the Information Science and Technology Study Group for DARPA. In 2012, Dr. Weiss received the AUVSI Foundation Award for Academic Champion. In 2012, she also received Women in Technology's Women of the Year Award for medium-sized businesses.

Contents

Volume 1

| | |
|---|------------|
| Section I Introduction | 1 |
| <i>Konstantinos Dalamagkidis, George J. Vachtsevanos and Kimon P. Valavanis</i> | |
| 1 Introduction..... | 3 |
| Kimon P. Valavanis and George J. Vachtsevanos | |
| 2 Introduction to the Handbook on UAVs | 5 |
| Kimon P. Valavanis and George J. Vachtsevanos | |
| 3 Definitions and Terminology | 43 |
| Konstantinos Dalamagkidis | |
| 4 Aviation History and Unmanned Flight | 57 |
| Konstantinos Dalamagkidis | |
| 5 Classification of UAVs..... | 83 |
| Konstantinos Dalamagkidis | |
| 6 Military and Civilian Unmanned Aircraft | 93 |
| George J. Vachtsevanos and Kimon P. Valavanis | |
| Section II UAV Design Principles | 105 |
| <i>Daniel J. Pack and Graham Drozeski</i> | |
| 7 UAV Design Principles: Introduction | 107 |
| Kimon P. Valavanis and George J. Vachtsevanos | |
| 8 Computational and Experimental Design of a Fixed-Wing UAV | 109 |
| Helen Ryaciotaki-Boussalis and Darrell Guillaume | |

| | | |
|--|--|------------|
| 9 | UAV Handbook: Payload Design of Small UAVs | 143 |
| | Scott Gruber, Hyukseong Kwon, Chad Hager, Rajnikant Sharma, Josiah Yoder, and Daniel Pack | |
| 10 | Small UAV Design Development and Sizing | 165 |
| | Steven A. Brandt | |
| 11 | Systematic Design Methodology and Construction of Micro Aerial Quadrotor Vehicles | 181 |
| | Swee King Phang, Kun Li, Ben M. Chen, and Tong H. Lee | |
| 12 | Dexterous UAVs for Precision Low-Altitude Flight | 207 |
| | Guangying Jiang and Richard M. Voyles | |
| Section III UAV Fundamentals | | 239 |
| <i>Isaac I. Kaminer</i> | | |
| 13 | UAV Fundamentals: Introduction | 241 |
| | Kimion P. Valavanis and George J. Vachtsevanos | |
| 14 | Kinematics and Dynamics of Fixed-Wing UAVs | 243 |
| | Vladimir Dobrokhodov | |
| 15 | Dynamic Model for a Miniature Aerobatic Helicopter | 279 |
| | Vladislav Gavrillets | |
| 16 | Quadrotor Kinematics and Dynamics | 307 |
| | Caitlin Powers, Daniel Mellinger, and Vijay Kumar | |
| 17 | Dynamics and Control of Flapping Wing MAVs | 329 |
| | David B. Doman, Michael W. Oppenheimer, and David O. Sigthorsson | |
| 18 | Principles of Guidance, Navigation, and Control of UAVs | 347 |
| | Gabriel Hugh Elkaim, Fidelis Adhika Pradipta Lie, and Demoz Gebre-Egziabher | |
| Section IV Sensors and Sensing Strategies | | 381 |
| <i>Ben Upcroft and Salah Sukkarieh</i> | | |
| 19 | Sensors and Sensing Strategies: Introduction | 383 |
| | Kimion P. Valavanis and George J. Vachtsevanos | |
| 20 | Sensors for Missions | 385 |
| | Luis Mejias, John Lai, and Troy Bruggemann | |
| 21 | Inertial Sensor-Based Simultaneous Localization and Mapping for UAVs | 401 |
| | Mitch Bryson and Salah Sukkarieh | |

| | | |
|-----------|--|------------|
| 22 | UAV Localization Using Inertial Sensors and Satellite Positioning Systems | 433 |
| | Mitch Bryson and Salah Sukkarieh | |
| 23 | Data Fusion and Tracking with Multiple UAVs | 461 |
| | Matthew F. Ridley, Ben Uperoft, and Salah Sukkarieh | |
| | Section V UAV Propulsion | 491 |
| | <i>George J. Vachtsevanos and Kimon P. Valavanis</i> | |
| 24 | UAV Propulsion: Introduction | 493 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 25 | Power Managements of a Hybrid Electric Propulsion System Powered by Solar Cells, Fuel Cells, and Batteries for UAVs | 495 |
| | Bohwa Lee, Poomin Park, and Chuntaek Kim | |

Volume 2

| | |
|---|---|
| Section VI UAV Control | 525 |
| <i>Jonathan P. How</i> | |
| 26 UAV Control: Introduction | 527 |
| | Kimon P. Valavanis and George J. Vachtsevanos |
| 27 Linear Flight Control Techniques for Unmanned Aerial Vehicles ... | 529 |
| | Jonathan P. How, Emilio Frazzoli, and Girish Vinayak Chowdhary |
| 28 Nonlinear Flight Control Techniques for Unmanned Aerial Vehicles | 577 |
| | Girish Vinayak Chowdhary, Emilio Frazzoli, Jonathan P. How, and Hugh Liu |
| 29 Adaptive Control of Unmanned Aerial Vehicles: Theory and Flight Tests | 613 |
| | Suresh K. Kannan, Girish Vinayak Chowdhary, and Eric N. Johnson |
| 30 Robust and Adaptive Control Methods for Aerial Vehicles | 675 |
| | Eugene Lavretsky |
| Section VII UAV Communication Issues | 711 |
| <i>Eric W. Frew</i> | |
| 31 UAV Communication Issues: Introduction | 713 |
| | Kimon P. Valavanis and George J. Vachtsevanos |

| | | |
|--|---|-------------|
| 32 | Problem of UAV Communications..... | 715 |
| | Stephen B. Heppe | |
| 33 | Cognitive Networking for UAV Swarms | 749 |
| | Christian Wietfeld and Kai Daniel | |
| 34 | Layered Approach to Networked Command and Control of Complex UAS | 781 |
| | Jack Elston, Maciej Stachura, Cory Dixon, Brian Argrow, and Eric W. Frew | |
| 35 | Cognitive Radio Architectures for Unmanned Aircraft Systems | 813 |
| | Timothy X. Brown, Mark McHenry, and Supapol Jaroonvanichkul | |
| Section VIII UAV Architectures | | 845 |
| <i>George J. Vachtsevanos and Kimon P. Valavanis</i> | | |
| 36 | UAV Architectures: Introduction | 847 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 37 | HDRC3: A Distributed Hybrid Deliberative/Reactive Architecture for Unmanned Aircraft Systems | 849 |
| | Patrick Doherty, Jonas Kvarnstrom, Mariusz Wzorek, Piotr Rudol, Fredrik Heintz, and Gianpaolo Conte | |
| 38 | Classification of Multi-UAV Architectures | 953 |
| | Ivan Maza, Aníbal Ollero, Enrique Casado, and David Scarlatti | |
| 39 | Operator Interaction with Centralized Versus Decentralized UAV Architectures | 977 |
| | M. L. Cummings | |
| Section IX UAV Health Management Issues | | 993 |
| <i>Kai Goebel and Michael J. Roemer</i> | | |
| 40 | UAV Health Management Issues: Introduction | 995 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 41 | Integrated Vehicle Health and Fault Contingency Management for UAVs | 999 |
| | Michael J. Roemer and Liang Tang | |
| 42 | Automated Failure Effect Analysis for PHM of UAV | 1027 |
| | Neal A. Snooke | |

| | | |
|-----------|---|------|
| 43 | Prognostics Applied to Electric Propulsion UAV | 1053 |
| | Kai Goebel and Bhaskar Saha | |
| 44 | Actuator Fault Detection in UAVs | 1071 |
| | Guillaume Ducard | |
| 45 | Experimental Validation of Fault Detection and Diagnosis for Unmanned Aerial Vehicles | 1123 |
| | Abbas Chamseddine, Mohammad Hadi Amoozgar, and Youmin Zhang | |
| 46 | Fault Detection and Diagnosis for NASA GTM UAV with Dual Unscented Kalman Filter | 1157 |
| | Youmin M. Zhang | |
| 47 | Fault Diagnosis of Skew-Configured Inertial Sensor System for Unmanned Aerial Vehicles | 1183 |
| | Seungho Yoon, Seungkeun Kim, Jonghee Bae, Youdan Kim, and Eung Tai Kim | |

Volume 3

Section X UAV Modeling, Simulation, Estimation and Identification **1213**

Lora Weiss

| | | |
|-----------|--|------|
| 48 | UAV Modeling, Simulation, Estimation and Identification: Introduction | 1215 |
| | Kimion P. Valavanis and George J. Vachtsevanos | |
| 49 | Flight Dynamics Modeling of Coaxial Rotorcraft UAVs | 1217 |
| | Fei Wang, Jinqiang Cui, Ben M. Chen, and Tong H. Lee | |
| 50 | Modeling of a Micro UAV with Slung Payload | 1257 |
| | Ying Feng, Camille Alain Rabbath, and Chun-Yi Su | |
| 51 | Command and Control of Autonomous Unmanned Vehicles | 1273 |
| | David H. Scheidt | |

Section XI MAVs and Bio-inspired UAVs **1299**

Robert C. Michelson

| | | |
|-----------|---|------|
| 52 | MAVs and Bio-inspired UAVs: Introduction | 1301 |
| | Kimion P. Valavanis and George J. Vachtsevanos | |
| 53 | Micro Air Vehicles | 1305 |
| | Robert C. Michelson | |

| | | |
|--|--|-------------|
| 54 | Survey of the Human-Centered Approach to Micro Air Vehicles ... | 1311 |
| | Stuart Michelson | |
| 55 | Development of Insect-Sized MAVs | 1329 |
| | Shigeru Sunada, Hao Liu, Hiroshi Tokutake, and Daisuke Kubo | |
| 56 | Flapping-Wing Propelled Micro Air Vehicles | 1359 |
| | Kevin D. Jones and Max F. Platzer | |
| 57 | Inventing a Biologically Inspired, Energy-Efficient Micro Aerial Vehicle | 1385 |
| | Jayant Ratti and George Vachtsevanos | |
| 58 | Issues Surrounding Communications with Micro Air Vehicles | 1415 |
| | Christian Michelson | |
| Section XII UAV Mission and Path Planning | | 1441 |
| <i>Camille Alain Rabbath and David Hyunchul Shim</i> | | |
| 59 | UAV Mission and Path Planning: Introduction | 1443 |
| | Kimion P. Valavanis and George J. Vachtsevanos | |
| 60 | Cooperative Mission Planning for Multi-UAV Teams | 1447 |
| | Sameera S. Ponda, Luke B. Johnson, Alborz Geramifard, and Jonathan P. How | |
| 61 | Multi-team Consensus Bundle Algorithm | 1491 |
| | Matthew E. Argyle, Randal W. Beard, and David W. Casbeer | |
| 62 | Cooperative Mission and Path Planning for a Team of UAVs | 1509 |
| | Hyondong Oh, Hyo-Sang Shin, Seungkeun Kim, Antonios Tsourdos, and Brian A. White | |
| 63 | Cooperative Task Assignment and Path Planning for Multiple UAVs | 1547 |
| | Sangwoo Moon, David Hyunchul Shim, and Eunmi Oh | |
| 64 | On the Decentralized Cooperative Control of Multiple Autonomous Vehicles | 1577 |
| | Michael J. Hirsch and Daniel Schroeder | |
| 65 | Innovative Collaborative Task Allocation for UAVs | 1601 |
| | Sertac Karaman, Emre Koyuncu, and Gokhan Inalhan | |
| 66 | Control of Communication Networks for Teams of UAVs | 1619 |
| | Andrew Kopeikin, Sameera S. Ponda, and Jonathan P. How | |
| 67 | Information-Theoretic Exploration of Unmanned Aerial Vehicle in Unknown Cluttered Environment | 1655 |
| | Kwangjin Yang, Seng Keat Gan, and Salah Sukkarieh | |

| | | |
|---|--|-------------|
| 68 | Implementing Dubins Airplane Paths on Fixed-Wing UAVs | 1677 |
| | Mark Owen, Randal W. Beard, and Timothy W. McLain | |
| 69 | Health Monitoring of a Drone Formation Affected by a Corrupted Control System | 1703 |
| | Nicolas Léchevin, Camille Alain Rabbath, and Patrick Maupin | |
| Section XIII UAV Autonomy | | 1721 |
| <i>Christos Cassandras, David Castañón and Lora Weiss</i> | | |
| 70 | UAV Autonomy: Introduction | 1723 |
| | Kimón P. Valavanis and George J. Vachtsevanos | |
| 71 | Integrated Hardware/Software Architectures to Enable UAVs for Autonomous Flight | 1725 |
| | Charles Pippin | |
| 72 | Distributed Optimization of Autonomous UAVs with Event-Driven Communication | 1749 |
| | Minyi Zhong and Christos G. Cassandras | |
| 73 | UAV Guidance Algorithms via Partially Observable Markov Decision Processes | 1775 |
| | Shankarachary Ragi and Edwin K. P. Chong | |

Volume 4

| | | |
|--|---|-------------|
| Section XIV UAV Sense, Detect and Avoid Systems | | 1811 |
| <i>Lennaert Speijker</i> | | |
| 74 | UAV Sense, Detect and Avoid: Introduction | 1813 |
| | Kimón P. Valavanis and George J. Vachtsevanos | |
| 75 | Development of a Regulatory Safety Baseline for UAS Sense and Avoid | 1817 |
| | Ahmet Oztekin and Rombout Wever | |
| 76 | Achieving Sense and Avoid for Unmanned Aircraft Systems: Assessing the Gaps for Science and Research | 1841 |
| | K. Douglas Davis and Stephen P. Cook | |
| 77 | Automatic Traffic Alert and Collision Avoidance System (TCAS) Onboard UAS | 1857 |
| | Jörg Meyer, Matthias Götken, Christoph Vernaleken, and Simon Schärer | |

| | | |
|---|--|-------------|
| 78 | Test Flights to Demonstrate Effectiveness of a Ground-Based Detect and Avoid Integration Concept | 1873 |
| | Andreas Udovic, Hans de Jong, and Jürgen Vielhauer | |
| 79 | Scalable RADAR-Based Sense-and-Avoid System for Unmanned Aircraft | 1895 |
| | Allistair Moses, Matthew J. Rutherford, and Kimon P. Valavanis | |
| 80 | Assessment of Detect and Avoid Solutions for Use of Unmanned Aircraft Systems in Nonsegregated Airspace | 1955 |
| | Joram Verstraeten, Martijn Stuij, and Tom van Birgelen | |
| Section XV Networked UAVs and UAV Swarms | | 1981 |
| <i>Randal Beard and Camille Alain Rabbath</i> | | |
| 81 | Networked UAVs and UAV Swarms: Introduction | 1983 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 82 | UAV Swarms: Models and Effective Interfaces | 1987 |
| | Airlie Chapman and Mehran Mesbahi | |
| 83 | Decentralized Multi-UAV Coalition Formation with Limited Communication Ranges | 2021 |
| | P. B. Sujit, J. G. Manathara, Debasish Ghose, and J. B. de Sousa | |
| 84 | Coordinated Convoy Protection Among Teams of Unmanned Aerial Vehicles | 2049 |
| | Magnus Egerstedt, Amir Rahmani, and Shin-Yih (Ryan) Young | |
| 85 | UAV Routing and Coordination in Stochastic, Dynamic Environments | 2079 |
| | John J. Enright, Emilio Frazzoli, Marco Pavone, and Ketan Savla | |
| Section XVI UAV Integration into the National Airspace | | 2111 |
| <i>Konstantinos Dalamagkidis and Richard S. Stansbury</i> | | |
| 86 | UAV Integration into the National Airspace: Introduction | 2113 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 87 | Aviation Regulation | 2117 |
| | Konstantinos Dalamagkidis | |
| 88 | Human Factors of Unmanned Aircraft System Integration in the National Airspace System | 2135 |
| | Bill Kaliardos and Beth Lyall | |

| | | |
|---------------------|--|-------------|
| 89 | Methodologies for Regulatory Compliance and Harmonization..... | 2159 |
| | Douglas Marshall | |
| 90 | Certification Strategy for Small Unmanned Aircraft Performing Nomadic Missions in the U.S. National Airspace System | 2177 |
| | Maciej Stachura, Jack Elston, Brian Argrow, Eric Frew, and Cory Dixon | |
| 91 | Hazard and Safety Risk Modeling | 2199 |
| | Konstantinos Dalamagkidis | |
| 92 | Safety Risk Management of Unmanned Aircraft Systems..... | 2229 |
| | Reece A. Clothier and Rodney A. Walker | |
| 93 | Certification of Small UAS | 2277 |
| | Ron van de Leijgraaf | |
| 94 | Technology Surveys and Regulatory Gap Analyses of UAS Subsystems Toward Access to the NAS | 2293 |
| | Richard S. Stansbury and Timothy A. Wilson | |
| 95 | Concept of Operations of Small Unmanned Aerial Systems: Basis for Airworthiness Towards Personal Remote Sensing | 2339 |
| | Brandon Stark, Calvin Coopmans, and YangQuan Chen | |
| 96 | Standards and Certification of a UAS Sense and Avoid Capability | 2361 |
| | Andrew D. Zeitlin | |
| Section XVII | UAV - Human Interfaces and Decision Support Systems | 2383 |
| | <i>Paul Oh and Michael Goodrich</i> | |
| 97 | UAV - Human Interfaces and Decision Support Systems: Introduction..... | 2385 |
| | Kimion P. Valavanis and George J. Vachtsevanos | |
| 98 | Human Interfaces in Micro and Small Unmanned Aerial Systems | 2389 |
| | Joshua M. Peschel and Robin R. Murphy | |
| 99 | Human Factors Perspective on Next Generation Unmanned Aerial Systems | 2405 |
| | M. A. Goodrich and M. L. Cummings | |
| 100 | Cognitive Task Analysis for Unmanned Aerial System Design | 2425 |
| | Julie A. Adams | |

| | | |
|------------|---|------|
| 101 | Display and Control Concepts for Multi-UAV Applications | 2443 |
| | Gloria L. Calhoun and Mark H. Draper | |
| 102 | UAV Operators Workload Assessment by Optical Brain Imaging Technology (fNIR) | 2475 |
| | Kurtulus Izzetoglu, Hasan Ayaz, James T. Hing, Patricia A. Shewokis, Scott C. Bunce, Paul Oh, and Banu Onaral | |

Volume 5

Section XVIII Human Factors and Training **2501** *Fulvia Quagliotti*

| | | |
|------------|--|------|
| 103 | Human Factors and Training: Introduction | 2503 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 104 | Using Best Practices as a Way Forward for Remotely Piloted Aircraft Operators: Integrated Combat Operations Training-Research Testbed | 2505 |
| | Leah J. Rowe, Sharon L. Conwell, Sean A. Morris, and Noah P. Schill | |
| 105 | From Research to Operations: The PITVANT UAS Training Experience | 2525 |
| | Maria da Luz Madruga Matos, João Vieira Caetano, A. P. Morgado, and J. B. de Sousa | |

Section XIX UAV Logistics Support **2561** *Fulvia Quagliotti and Nafiz Alemdaroglu*

| | | |
|------------|---|------|
| 106 | UAV Logistics Support: Introduction | 2563 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 107 | UAV Logistic Support Definition | 2565 |
| | Sergio Chiesa and Marco Fioriti | |
| 108 | UAV Logistics for Life-Cycle Management | 2601 |
| | Cengiz Karaağaç, Ahmet G. Pakfiliz, Fulvia Quagliotti, and Nafiz Alemdaroglu | |

Section XX UAV Applications **2637** *Aníbal Ollero*

| | | |
|------------|---|------|
| 109 | UAV Applications: Introduction | 2639 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |

| | | |
|------------|--|-------------|
| 110 | Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring | 2643 |
| | Konstantinos Kanistras, Goncalo Martins, Matthew J. Rutherford, and Kimon P. Valavanis | |
| 111 | Measurement and Exploration in Volcanic Environments | 2667 |
| | Carmelo Donato Melita, Domenico Longo, Giovanni Muscato, and Gaetano Giudice | |
| 112 | Cooperative Unmanned Aerial Systems for Fire Detection, Monitoring, and Extinguishing | 2693 |
| | Luis Merino, José Ramiro Martínez-de Dios, and Aníbal Ollero | |
| 113 | Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States | 2723 |
| | Vincent G. Ambrosia and Thomas Zajkowski | |
| 114 | Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation | 2755 |
| | Konstantin Kondak, Aníbal Ollero, Ivan Maza, Kai Krieger, Alin Albu-Schaeffer, Marc Schwarzbach, and Maximilian Laiacker | |
| 115 | Unmanned Aircraft Systems for Maritime Operations | 2787 |
| | J. B. de Sousa, Philip McGuillivary, João Vicente, Maria Nunes Bento, José A. P. Morgado, Maria Madruga Matos, Ricardo Ayres Gomes Bencatel, and Paulo Mónica de Oliveira | |
| 116 | Autonomous Remote Sensing of Invasive Species from Robotic Aircraft | 2813 |
| | Ali Haydar Göktoğan and Salah Sukkarieh | |
| 117 | Cyber-Physical Systems Enabled by Small Unmanned Aerial Vehicles | 2835 |
| | Calvin Coopmans, Brandon Stark, Austin Jensen, YangQuan Chen, and Mac McKee | |
| | Section XXI Social and Ethical Implications | 2861 |
| | <i>George R. Lucas Jr.</i> | |
| 118 | Social and Ethical Implications: Introduction | 2863 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| 119 | Ethics and UAVs | 2865 |
| | George R. Lucas Jr. | |

| | | |
|------------|---|-------------|
| 120 | International Governance of Autonomous Military Robots | 2879 |
| | Gary E. Marchant, Braden Allenby, Ronald Arkin, Jason Borenstein, Lyn M. Gaudet, Orde Kittrie, Patrick Lin, George R. Lucas Jr., Richard O'Meara, and Jared Silberman | |
| 121 | The Moral Case Against Autonomous and SemiAutonomous UAVs | 2919 |
| | Noel Sharkey | |
| 122 | The Moral Case for Autonomy in Unmanned Systems | 2933 |
| | Ronald C. Arkin | |
| 123 | Moral Predators: The Duty to Employ Uninhabited Aerial Vehicles | 2943 |
| | Bradley Jay Strawser | |
| 124 | Killer Robots: Ethical Issues in the Design of Unmanned Systems for Military Applications | 2965 |
| | Robert Sparrow | |
| 125 | Ethical Issues of UAVs for National Academy of Sciences | 2985 |
| | George R. Lucas Jr. | |
| | Section XXII Epilogue | 2991 |
| | <i>George J. Vachtsevanos and Kimon P. Valavanis</i> | |
| 126 | Future of Unmanned Aviation | 2993 |
| | Kimon P. Valavanis and George J. Vachtsevanos | |
| | Index | 3011 |

Contributors

Julie A. Adams Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA

Alin Albu-Schaeffer Institute of Robotics and Mechatronics, DLR (German Aerospace Center), Oberpfaffenhofen-Wessling, Germany

Nafiz Alemdaroglu Aerospace Engineering Department, Middle East Technical University, Ankara, Turkey

Braden Allenby School of Sustainable Engineering and the Built Environment in Civil Engineering, Arizona State University, Tempe, AZ, USA

Vincent G. Ambrosia NASA-Ames Research Center, California State University – Monterey Bay, Moffett Field, CA, USA

Mohammad Hadi Amoozgar Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

Brian Argrow Director Research and Engineering Center for Unmanned Vehicles, University of Colorado, Boulder, CO, USA

Matthew E. Argyle Department of Electrical Engineering, Brigham Young University, Provo, UT, USA

Ronald C. Arkin Department of Computer Science, Georgia Tech University, Atlanta, GA, USA

Hasan Ayaz School of Biomedical Engineering, Science and Health Systems, Drexel University, Philadelphia, PA, USA

Jonghee Bae School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea

Randal W. Beard Electrical and Computer Engineering Department, Brigham Young University, Provo, UT, USA

Ricardo Ayres Gomes Bencatel Aerospace Engineering Department, University of Michigan, Ann Arbor, MI, USA

Maria Nunes Bento Centro de Investigação, Academia da Força Aérea, Sintra, Portugal

Tom van Birgelen National Aerospace Laboratory of the Netherlands – Air Transport Safety Institute, Amsterdam, The Netherlands

Jason Borenstein Center for Ethics & Technology, Georgia Tech University, Atlanta, GA, USA

J.B. de Sousa Department of Electrical and Computer Engineering, University of Porto, Porto, Portugal

Steven A. Brandt Department of Aeronautics, United States Air Force Academy, Colorado Springs, CO, USA

Timothy X. Brown University of Colorado, Boulder, CO, USA

Troy Bruggemann Australian Research Centre for Aerospace Automation, Queensland University of Technology, Brisbane, QLD, Australia

Mitch Bryson Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW, Australia

Scott C. Bunce Department of Psychiatry, M.S. Hershey Medical Center, Penn State University College of Medicine, Hershey, PA, USA

João Vieira Caetano Centro de Investigação, Academia da Força Aérea, Sintra, Portugal

Gloria L. Calhoun Wright-Patterson Air Force Base, 711 HPW/RHCI, Air Force Research Laboratory, Dayton, OH, USA

Enrique Casado Boeing Research & Technology Europe, Madrid, Spain

David W. Casbeer Control Science Center of Excellence, Air Force Research Laboratory, WPAFB, OH, USA

Christos G. Cassandras Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

Abbas Chamseddine Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

Airlie Chapman Aeronautics and Astronautics, University of Washington, Seattle, WA, USA

Ben M. Chen Control and Simulation Lab, Department of Electrical and Computer Engineering, National University of Singapore, Singapore

YangQuan Chen Mechatronics, Embedded Systems and Automation (MESA) Lab, School of Engineering, University of California, Merced, CA, USA

Sergio Chiesa Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy

Edwin K. P. Chong Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA

Girish Vinayak Chowdhary Department of Aeronautics and Astronautics, Aerospace Controls Laboratory, Massachusetts Institute of Technology Laboratory for Information and Decision Systems, Cambridge, MA, USA

Reece A. Clothier School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Bundoora, Melbourne, VIC, Australia

Stephen P. Cook MITRE Corporation, Bedford, MA, USA

Calvin Coopmans The Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State University, Logan, UT, USA

Gianpaolo Conte Department of Computer and Information Science, Linköping University, Linköping, Sweden

Sharon L. Conwell 711th Human Performance Wing Human Effectiveness Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA

Jinqiang Cui Control and Simulation Lab, Department of Electrical and Computer Engineering, National University of Singapore, Singapore

M.L. Cummings Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

Konstantinos Dalamagkidis Institut für Informatik I6, Technische Universität München, Garching bei München, Germany

Kai Daniel Communication Networks Institute, Faculty of Electrical Engineering and Information Technology, TU Dortmund University, Dortmund, Germany

K. Douglas Davis Physical Science Laboratory, New Mexico State University, Las Cruces, NM, USA

Hans de Jong Aeronautical Solutions, DFS Deutsche Flugsicherung GmbH (German Air Navigation Services), Langen, Germany

Paulo Mónica de Oliveira Instituto Português do Mar e da Atmosfera, Lisboa, Portugal

Cory Dixon College of Engineering and Applied Science, University of Colorado at Boulder, Boulder, CO, USA

Vladimir Dobrokhodov Mechanical and Aerospace Engineering Department, Naval Postgraduate School, Monterey, CA, USA

Patrick Doherty Department of Computer and Information Science, Linköping University, Linköping, Sweden

David B. Doman Control Design and Analysis Branch, Air Force Research Laboratory AFRL/RBCA, Wright-Patterson AFB, OH, USA

Mark H. Draper Wright-Patterson Air Force Base, 711 HPW/RHCI, Air Force Research Laboratory, Dayton, OH, USA

Guillaume Ducard I3S CNRS-UNS, Sophia Antipolis, France
ETH Zurich, IDSC, Zurich, Switzerland

Magnus Egerstedt School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Gabriel Hugh Elkaim Computer Engineering Department, University of California Santa Cruz, Santa Cruz, CA, USA

Jack Elston Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, USA

John J. Enright Kiva Systems, North Reading, MA, USA

Ying Feng Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

Marco Fioriti Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy

Emilio Frazzoli Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

Eric W. Frew Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, USA

Seng Keat Gan Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW, Australia

Lyn M. Gaudet Center for Law, Science & Innovation Sandra Day O'Connor School of Law, Arizona State University, Tempe, AZ, USA

Vladislav Gavrillets Flight Control and Navigation Group, Rockwell Collins Control Technologies, Warrenton, VA, USA

Demoz Gebre-Egziabher Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA

Alborz Geramifard Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts Institute of Technology, Cambridge, MA, USA

Debasish Ghose Guidance, Control, and Decision Systems Laboratory (GCDL), Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

Gaetano Giudice Istituto Nazionale di Geofisica e Vulcanologia, Palermo, Italy

Kai Goebel NASA Ames Research Center, Moffett Field, CA, USA

Ali Haydar Göktoğan Australian Centre for Field Robotics (ACFR), School of Aerospace, Mechanical & Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW, Australia

M. A. Goodrich Computer Science Department, Brigham Young University, Provo, UT, USA

Matthias Göttken Cassidian, Manching, Germany

Scott Gruber Academy Center for Unmanned Aircraft Systems Research, Department of Electrical and Computer Engineering, United States Air Force Academy, Colorado Springs, CO, USA

Darrell Guillaume Department of Mechanical Engineering, NASA University Research Center (SPACE), California State University, Los Angeles, CA, USA

Chad Hager Academy Center for Unmanned Aircraft Systems Research, Department of Electrical and Computer Engineering, United States Air Force Academy, Colorado Springs, CO, USA

Fredrik Heintz Department of Computer and Information Science, Linköping University, Linköping, Sweden

Stephen B. Heppe Telenergy, Inc., Hood River, OR, USA

James T. Hing Department of Mechanical Engineering and Mechanics, Drexel Autonomous Systems Lab, Drexel University, Philadelphia, PA, USA

Michael J. Hirsch Intelligence, Information, and Services, Raytheon Company, Maitland, FL, USA

Jonathan P. How Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts Institute of Technology, Cambridge, MA, USA

Gokhan Inalhan Faculty of Aeronautics and Astronautics, Istanbul Technical University, Istanbul, Turkey

Kurtulus Izzetoglu School of Biomedical Engineering, Science and Health Systems, Drexel University, Philadelphia, PA, USA

Supapol Jaroovanichkul University of Colorado, Boulder, CO, USA

Austin Jensen The Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State University, Logan, UT, USA

Guangying Jiang Department of Electrical and Computer Engineering, University of Denver, Denver, CO, USA

Eric N. Johnson School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Luke B. Johnson Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts Institute of Technology, Cambridge, MA, USA

Kevin D. Jones Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, USA

Bill Kaliardos Federal Aviation Administration, Washington, DC, USA

Suresh K. Kannan Controls Group, Systems Department, United Technologies Research Center, East Hartford, CT, USA

Konstantinos Kanistras Department of Electrical and Computer Engineering, University of Denver, Denver, CO, USA

Cengiz Karaağaç Systems Department, STM A.Ş., Ankara, Turkey

Sertac Karaman Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

Chuntaek Kim Aero Propulsion Division, Korea Aerospace Research Institute, Daejeon, South Korea

Eung Tai Kim Flight Control and Avionics Team, Korea Aerospace Research Institute, Daejeon, South Korea

Seungkeun Kim Department of Aerospace Engineering, Chungnam National University, Daejeon, South Korea

Youdan Kim School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea

Orde Kittrie Arizona State University School of Law, Tempe, AZ, USA

Konstantin Kondak Institute of Robotics and Mechatronics, DLR (German Aerospace Center), Oberpfaffenhofen-Wessling, Germany

Andrew Kopeikin Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts Institute of Technology, Cambridge, MA, USA

Emre Koyuncu Controls and Avionics Laboratory, Istanbul Technical University, Istanbul, Turkey

Kai Krieger Institute of Robotics and Mechatronics, DLR (German Aerospace Center), Oberpfaffenhofen-Wessling, Germany

Daisuke Kubo Aviatios Program Group, Japan Aerospace Exploration Agency, Tokyo, Japan

Vijay Kumar Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, USA

Jonas Kvarnström Department of Computer and Information Science, Linköping University, Linköping, Sweden

Hyukseong Kwon Academy Center for Unmanned Aircraft Systems Research, Department of Electrical and Computer Engineering, United States Air Force Academy, Colorado Springs, CO, USA

John Lai Australian Research Centre for Aerospace Automation, Queensland University of Technology, Brisbane, QLD, Australia

Maximilian Laiacker Institute of Robotics and Mechatronics, DLR (German Aerospace Center), Oberpfaffenhofen-Wessling, Germany

Eugene Lavretsky Boeing Senior Technical Fellow, The Boeing Company, Huntington Beach, CA, USA

Bohwa Lee Aero Propulsion Division, Korea Aerospace Research Institute, Daejeon, South Korea

Tong H. Lee Control and Simulation Lab, Department of Electrical and Computer Engineering, National University of Singapore, Singapore

Nicolas Léchevin Defence Research and Development Canada – Valcartier, Québec, QC, Canada

Kun Li Control and Simulation Lab, National University of Singapore, Singapore

Patrick Lin Center for Ethics & Emerging Technologies, California Polytechnic University, San Luis Obispo, CA, USA

Fidelis Adhika Pradipta Lie Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA

Hugh Liu University of Toronto UTIAS Flight Systems and Control Research Group, Toronto, ON, Canada

Hao Liu Graduate School of Engineering, Chiba University, Chiba, Japan
Shanghai Jiao Tong University and Chiba University International Cooperative Research Center (SJTU-CU ICRC), Shanghai, China

Domenico Longo Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Catania, Italy

George R. Lucas Jr. United States Naval Academy, Annapolis, MD, USA
Naval Postgraduate School, Monterey, CA, USA

Beth Lyall Research Integrations, Incorporated, Beaverton, OR, USA

Maria Madruga Matos Centro de Investigação, Academia da Força Aérea, Sintra, Portugal

J. G. Manathara Guidance, Control, and Decision Systems Laboratory (GCDSL), Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

Gary E. Marchant Arizona State University School of Law, Tempe, AZ, USA

Douglas Marshall Physical Science Laboratory, New Mexico State University, Las Cruces, NM, USA

Maria da Luz Madruga Matos Centro de Investigação, Academia da Força Aérea, Sintra, Portugal

Patrick Maupin Department of Engineering, Concordia University, Montreal, QC, Canada

José Ramiro Martínez-de Dios University of Seville, Seville, Spain

Goncalo Martins Department of Electrical and Computer Engineering, University of Denver, Denver, CO, USA

Ivan Maza Robotics, Vision and Control Group, University of Seville, Seville, Spain

Philip McGuillivray United States Coast Guard, PACAREA Science Liaison, Alameda, CA, USA

Mark McHenry Shared Spectrum Company, Vienna, VA, USA

Mac McKee Utah Water Research Laboratory, Utah State University, Logan, UT, USA

Timothy W. McLain Mechanical Engineering Department, Brigham Young University, Provo, UT, USA

Luis Mejias Australian Research Centre for Aerospace Automation, Queensland University of Technology, Brisbane, QLD, Australia

Carmelo Donato Melita Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Catania, Italy

Daniel Mellinger KMeI Robotics, Philadelphia, PA, USA

Luis Merino Pablo de Olavide University, Seville, Spain

Mehran Mesbahi Aeronautics and Astronautics, University of Washington, Seattle, WA, USA

Jörg Meyer Cassidian, Manching, Germany

Christian Michelson Test Engineering Division, Georgia Tech Research Institute, Atlanta, GA, USA

Robert C. Michelson Aerospace, Transportation, and Advanced Systems Laboratory, Georgia Tech Research Institute, Smyrna, GA, USA

Stuart Michelson Human Systems Integration Division, Georgia Tech Research Institute, Atlanta, GA, USA

Sangwoo Moon Department of Aerospace and Mechanical Engineering, Korea Air Force Academy, Cheongju, South Korea

José A. P. Morgado Centro de Investigação, Academia da Força Aérea, Sintra, Portugal

Sean A. Morris Air Force Research Laboratory, 711th Human Performance Wing Human Effectiveness Directorate, OH, USA

Allistair Moses University of Denver, Denver, CO, USA

Robin R. Murphy Center for Robot-Assisted Search and Rescue, Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA

Giovanni Muscato Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Catania, Italy

Eunmi Oh CNS/ATM & Satellite Navigation Research Center, Korea Aerospace Research Institute, Daejeon, South Korea

Hyondong Oh Department of Engineering Physics, School of Engineering, Cranfield University, Cranfield, Bedfordshire, UK

Paul Oh Formerly with Drexel University

Currently: University of Nevada, Las Vegas, NV, USA

Aníbal Ollero Robotics, Vision and Control Group, University of Seville, Seville, Spain

Center for Advanced Aerospace Technologies (CATEC), Parque Tecnológico y Aeronáutico de Andalucía, La Rinconada, Spain

Richard O'Meara Division of Global Affairs - Rutgers, The State University of New Jersey, Conklin Hall, Conklin Hall, NJ, USA

Banu Onaral School of Biomedical Engineering, Science and Health Systems, Drexel University, Philadelphia, PA, USA

Michael W. Oppenheimer Control Design and Analysis Branch, Air Force Research Laboratory, Wright-Patterson AFB, OH, USA

Mark Owen Sandia National Labs, Albuquerque, NM, USA

Ahmet Oztekin Hi-Tec Systems, Inc., FAA William J. Hughes Technical Center, Atlantic City, NJ, USA

Daniel Pack Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX, USA

Ahmet G. Pakfiliz 3rd Air Supply and Maintenance Center, Turkish Air Force, Ankara, Turkey

Poomin Park Aero Propulsion Division, Korea Aerospace Research Institute, Daejeon, South Korea

Marco Pavone Aeronautics and Astronautics Department, Stanford University, Stanford, CA, USA

Joshua M. Peschel Human-Infrastructure Interaction Lab, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Swee King Phang Control and Simulation Lab, National University of Singapore, Singapore

Charles Pippin Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, GA, USA

Max F. Platzer Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, USA

Sameera S. Ponda Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts Institute of Technology, Cambridge, MA, USA

Caitlin Powers Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, USA

Fulvia Quagliotti Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy

Camille Alain Rabbath Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

Shankarachary Ragi Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA

Amir Rahmani Department of Mechanical and Aerospace Engineering, University of Miami, Coral Gables, FL, USA

Jayant Ratti Robotics & Intelligent Machines, TechJect Inc., Atlanta, GA, USA

Matthew F. Ridley Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW, Australia

Michael J. Roemer Impact Technologies, LLC, Rochester, NY, USA

Leah J. Rowe 711th Human Performance Wing Human Effectiveness Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA

Piotr Rudol Department of Computer and Information Science, Linköping University, Linköping, Sweden

Matthew J. Rutherford Department of Computer Science, University of Denver, Denver, CO, USA

Helen Ryaciotaki-Boussalis Department of Electrical and Computer Engineering, NASA University Research Center (SPACE), California State University, Los Angeles, CA, USA

Bhaskar Saha Palo Alto Research Center, Palo Alto, CA, USA

Ketan Savla Sonny Astani Department of Civil and Environmental Engineering, University of Southern California, Los Angeles, CA, USA

David Scarlatti Boeing Research & Technology Europe, Madrid, Spain

Simon Schärer Cassidian, Manching, Germany

David H. Scheidt Johns Hopkins University Applied Physics Laboratory, Laurel, MD, USA

Noah P. Schill 711th Human Performance Wing Human Effectiveness Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA

Daniel Schroeder Intelligence, Information, and Services, Raytheon Company, State College, PA, USA

Marc Schwarzbach Institute of Robotics and Mechatronics, DLR (German Aerospace Center), Oberpfaffenhofen-Wessling, Germany

Noel Sharkey Department of Computer Science, University of Sheffield, Sheffield, UK

Rajnikant Sharma Academy Center for Unmanned Aircraft Systems Research, Department of Electrical and Computer Engineering, United States Air Force Academy, Colorado Springs, CO, USA

Patricia A. Shewokis College of Nursing and Health Professions and School of Biomedical Engineering, Science and Health Systems, Drexel University, Philadelphia, PA, USA

David Hyunchul Shim Department of Aerospace Engineering, Korean Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea

Hyo-Sang Shin Department of Engineering Physics, School of Engineering, Cranfield University, Cranfield, Bedfordshire, UK

David O. Sigthorsson General Dynamics Information Technology, Air Force Research Laboratory, Wright-Patterson AFB, OH, USA

Jared Silberman Associate Counsel for Arms Control and International Law, United States Navy Office of Strategic Systems Programs (SSP), Washington, DC, USA

Neal A. Snooke Department of Computer Science, Aberystwyth University, Llandinam Extension, Aberystwyth, Ceredigion, UK

Robert Sparrow Department of Philosophy, Monash University, Canberra, Australia

Maciej Stachura Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, USA

Richard S. Stansbury Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA

Brandon Stark Mechatronics, Embedded Systems and Automation (MESA) Lab, School of Engineering, University of California, Merced, CA, USA

Bradley Jay Strawser Naval Postgraduate School, Monterey, CA, USA

Martijn Stuij National Aerospace Laboratory of the Netherlands, Amsterdam, The Netherlands

Chun-Yi Su Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

P. B. Sujit Department of Electrical and Computer Engineering, University of Porto, Porto, Portugal

Salah Sukkarieh Australian Centre for Field Robotics (ACFR), School of Aerospace, Mechanical & Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW, Australia

Shigeru Sunada Department of Aerospace Engineering, Osaka Prefecture University, Sakai, Osaka, Japan

Liang Tang Impact Technologies, LLC, Rochester, NY, USA

Hiroshi Tokutake Faculty of Mechanical Engineering, Kanazawa University, Kanazawa, Ishikawa, Japan

Antonios Tzourdos Department of Engineering Physics, School of Engineering, Cranfield University, Cranfield, Bedfordshire, UK

Andreas Udovic Research and Development, DFS Deutsche Flugsicherung GmbH (German Air Navigation Services), Langen, Germany

Ben Upcroft School of Electrical Engineering and Computer Science, Brisbane, QLD, Australia

George J. Vachtsevanos Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

Kimón P. Valavanis John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

Ron van de Leijgraaf Civil Aviation Authorities – The Netherlands, Airworthiness Inspectorate, Hoofddorp, The Netherlands

Christoph Vernaleken Cassidian, Manching, Germany

Joram Verstraeten National Aerospace Laboratory of the Netherlands – Air Transport Safety Institute, Amsterdam, The Netherlands

João Vicente Portuguese Joint Command and Staff College/Security and Defense Research Center, Lisboa, Portugal

Jürgen Vielhauer Research and Development, DFS Deutsche Flugsicherung GmbH (German Air Navigation Services), Langen, Germany

Richard M. Voyles Department of Electrical and Computer Engineering, University of Denver, Denver, CO, USA

Rodney A. Walker Australian Research Centre for Aerospace Automation, Queensland University of Technology, Brisbane Airport, Brisbane, QLD, Australia

Fei Wang Control and Simulation Lab, Department of Electrical and Computer Engineering, National University of Singapore, Singapore

Rombout Weaver NLR Air Transport Safety Institute (NLR-ATSI), Amsterdam, The Netherlands

Brian A. White Department of Engineering Physics, School of Engineering, Cranfield University, Cranfield, Bedfordshire, UK

Christian Wietfeld Communication Networks Institute, Faculty of Electrical Engineering and Information Technology, TU Dortmund University, Dortmund, Germany

Timothy A. Wilson Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA

Mariusz Wzorek Department of Computer and Information Science, Linköping University, Linköping, Sweden

Kwangjin Yang Department of Aerospace and Mechanical Engineering, Korea Air Force Academy, Chungbuk, Republic of Korea

Josiah Yoder Academy Center for Unmanned Aircraft Systems Research, Department of Electrical and Computer Engineering, United States Air Force Academy, Colorado Springs, CO, USA

Seungho Yoon School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea

Shin-Yih (Ryan) Young Advanced Technology Center, Rockwell Collins, Cedar Rapids, IA, USA

Thomas Zajkowski Next Gen Air Transportation (NGAT) Center, Institute for Transportation Research and Education, North Carolina State University, Raleigh, NC, USA

Andrew D. Zeitlin The MITRE Corporation, Mc Lean, VA, USA

Youmin M. Zhang Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

Minyi Zhong Server and Tools Division, Microsoft Corporation, Redmond, WA, USA

Section I

Introduction

***Konstantinos Dalamagkidis,
George J. Vachtsevanos and
Kimon P. Valavanis***

Kimon P. Valavanis and George J. Vachtsevanos

This Introduction of the handbook presents background information related to unmanned aircraft in general. It is mostly nontechnical, focusing on introducing the concept and idea of the unmanned/pilotless aircraft and how unmanned aircraft evolved over the years reaching their current state.

To begin with, it is crucial to understand that the field of unmanned aircraft has been plagued with different terms, which basically describe the same “concept.” Even the term UAV is controversial, and it is being replaced in many places with RPA, UAS, or RPAS. Different terms often result from the different requirements and concepts between military and civilian systems or from regulatory and legal considerations. Regardless, ►[definitions](#) for some of the most widely used terminology are presented in this section. Further, throughout the handbook, the different acronyms/terms that define unmanned aircraft are used interchangeably; however, the context in which such terms are used is clear and noncontroversial.

►[A historical and pictorial perspective of unmanned aviation](#) is also presented, which depicts that unmanned flight has a rich history that goes back all the way to ancient times, although the first systems that qualify with the modern definition of UAVs are quite recent and mainly involve the reconnaissance drones developed and deployed during the Cold War.

►[Proposed UAV classification](#) schemes from a variety of sources, both ►[civilian and military](#), are included to help differentiate existing systems based on their

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

operational characteristics and their capabilities. Several of these schemes are also of regulatory importance since the metrics used directly correlate with risk from ground impact or midair collision accidents.

Concise information about the rest of the handbook is also included in this part, since it aims at familiarizing the reader with the unmanned/pilotless aircraft concept at large and the potential of unmanned aviation for a very wide and diverse spectrum of applications.

In summary, this section educates the reader who will be ready to continue with the basic UAV design principles at a technical level.

Kimon P. Valavanis and George J. Vachtsevanos

Contents

| | | |
|--------|--|----|
| 2.1 | Introduction | 6 |
| 2.2 | UAVs in the Context of Unmanned Systems | 11 |
| 2.3 | Outline of the Handbook Sections | 13 |
| 2.3.1 | UAV Design Principles | 13 |
| 2.3.2 | UAV Fundamentals | 14 |
| 2.3.3 | UAV Sensors and Sensing Strategies | 15 |
| 2.3.4 | UAV Propulsion | 16 |
| 2.3.5 | UAV Control | 17 |
| 2.3.6 | UAV Communication Issues | 18 |
| 2.3.7 | UAV Architectures | 19 |
| 2.3.8 | UAV Health Management Issues | 20 |
| 2.3.9 | UAV Modeling, Simulation, Estimation, and Identification | 23 |
| 2.3.10 | MAVs and Bio-inspired UAVs | 24 |
| 2.3.11 | UAV Mission and Path Planning | 25 |
| 2.3.12 | UAV Autonomy | 29 |
| 2.3.13 | UAV Sense, Detect, and Avoid Systems | 30 |
| 2.3.14 | Networked UAVs and UAV Swarms | 32 |
| 2.3.15 | UAV Integration into the National Airspace | 34 |
| 2.3.16 | UAV-Human Interfaces and Decision Support Systems | 36 |
| 2.3.17 | Human Factors and Training | 38 |
| 2.3.18 | UAV Logistics Support | 39 |
| 2.3.19 | UAV Applications | 39 |
| 2.3.20 | Social and Ethical Implications | 41 |
| 2.4 | Remarks | 42 |

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

Abstract

This chapter presents an overall introduction and overview of the handbook, including each section objectives along with summaries of the chapters included in each section. The goal of this chapter is twofold: (i) familiarize the reader with the contents of each section and (ii) provide the reader with a choice to prioritize reading based on her/his specific interest and knowledge level. After reading this chapter, the reader will have a rather complete picture of the material included in the handbook.

2.1 Introduction

The *Handbook of Unmanned Aerial Vehicles* presents a unique and comprehensive treatise of everything one needs to know about unmanned aircraft, from conception to operation, from technologies to business activities, users, OEMs, reference sources, conferences, publications, and professional societies. The main objective is for the handbook to serve as a Thesaurus, an indispensable part of the library of everyone involved in the area of unmanned aviation. For the first time, contributions by the world's top experts from academia, industry, government, and the private sector are brought together to provide unique perspectives on the current state of the art in unmanned aviation, as well as future directions.

The handbook is intended for the academic and research communities, industry, manufacturers, users, practitioners, federal government, federal and state agencies, the private sector, as well as all organizations that are and will be using unmanned aircraft in a wide spectrum of applications. It covers all aspects of UAVs, from design to logistics and ethical issues. It is also targeting the young investigator, the future inventor, and entrepreneur by providing an overview and detailed information of the state of the art as well as useful new concepts that may lead to innovative research. The contents of the handbook include material that addresses the needs and *know-how* of all of the above sectors targeting a very diverse audience.

The title of the handbook, *Handbook of Unmanned Aerial Vehicles*, reflects nothing more but the preferred term at the time the proposal was made to the publisher, and it should be treated as such. It is noted that different names are being used to describe unmanned aviation and aircraft (UAV, RPA, UAS, RPAS) in subsequent chapters, but the context in which such names are used in the handbook is self-explanatory. For example, the terms UAVs and UAS have distinct definitions, and it should be clear why, at first, UAVs became UAS: A UAV refers to a pilotless aircraft, a flying machine without an onboard human pilot or passengers. As such, "unmanned" implies total absence of a human who directs and actively pilots the aircraft. Control functions for unmanned aircraft may be either onboard or offboard (remote control). The term UAS was introduced by the U.S. Department of Defense, followed by the FAA and the European Aviation Safety Agency (EASA). This term was meant to signify that UAS are aircrafts, and as such airworthiness will need to be demonstrated, and they are also systems consisting of ground control stations,

communication links and launch, and retrieval systems in addition to the aircraft itself. Similar distinctions are true for RPA versus RPAS. Other names include remotely piloted vehicles (RPVs), a term that was used in the Vietnam War, while the U.S. military also calls them RPA, a term used to include both the aircraft and the pilot, while the UK and ICAO have designated them as RPAS, to demonstrate the presence of the man in the loop to control them. The ICAO recommended definition for RPAS is “A set of configurable elements consisting of a remotely-piloted aircraft, its associated remote pilot station(s), the required command and control links and any other system elements as may be required, at any point during flight operation.”

Material in the handbook is arranged in a user-friendly format divided into twenty-one (21) sections in addition to the Introduction section. Each section contains either original material or enhanced versions of already published work, written by internationally renowned authors who are authorities in their respective fields. The corresponding sections are:

- UAV Design Principles
- UAV Fundamentals
- UAV Sensors and Sensing Strategies
- UAV Propulsion
- UAV Control
- UAV Communication Issues
- UAV Architectures
- UAV Health Management Issues
- UAV Modeling, Simulation, Estimation, and Identification
- MAVs and Bio-inspired UAVs
- UAV Mission and Path Planning
- UAV Autonomy
- UAV Sense, Detect, and Avoid Systems
- Networked UAVs and UAV Swarms
- UAV Integration into the National Airspace
- UAV-Human Interfaces and Decision Support Systems
- Human Factors and Training
- UAV Logistics Support
- UAV Applications
- Social and Ethical Implications
- Epilogue

The Introduction section of the handbook presents background information related to unmanned aircraft in general. It is mostly nontechnical, focusing on introducing the concept and idea of the unmanned/pilotless aircraft and how unmanned aircraft evolved over the years reaching their current state. A historical and pictorial perspective of unmanned aviation is also presented, which depicts that unmanned flight has a rich history that goes back all the way to ancient times, although the first systems that qualify with the modern definition of UAVs are quite recent and mainly involve the reconnaissance drones developed and deployed during the cold war.

Section ► [UAV Design Principles](#) presents the basic design principles of UAVs and the components that comprise a complete unmanned aircraft system, avoiding in-depth technical details. It aims at creating the “overall picture” with respect to UAV functionality and, consequently, to indirectly motivate the reader about the challenges that need to be overcome in order to make UAVs fully operational, reliable, and safe.

Section ► [UAV Fundamentals](#) details what constitutes the foundational elements of UAV design technologies, covering kinematics and dynamics for fixed-wing, rotorcraft, quadrotors, and flapping-wing MAVs, as well as *principles* of UAV guidance, navigation, and control.

Section ► [UAV Sensors and Sensing Strategies](#) focuses on what enables an unmanned aircraft to “sense,” “see,” “hear,” and “understand” the world around it so that it may function intelligently in an unknown and cluttered environment and in the absence of an onboard pilot. In essence, sensors and sensing strategies are crucial since they provide the technologies that will result in “unmanned aircraft operating as if there were a human pilot onboard.”

Section ► [UAV Propulsion](#) refers to that critical and indispensable module of the UAV structure, which provides the necessary thrust to maintain a flight. The UAV performance, effectiveness, and utility depend strongly on the onboard propulsion capabilities. Jointly with other UAV design considerations, the propulsion system determines the endurance, size, weight, and application domain of all UAV classes.

Section ► [UAV Control](#) presents a very comprehensive treatment of UAV control and related technologies. The inherently unstable nature of typical UAV configurations necessitates a rigorous approach to the analysis and design of UAV control technologies, as well as a thorough understanding of stability issues.

Section ► [UAV Communication Issues](#) discusses the development and application of robust, effective, and secure communication technologies that will enable unmanned aircraft to unmanned aircraft, unmanned aircraft to ground or mother ship communications, and unmanned aircraft to air traffic controller communications. Unmanned aircraft control and communication issues are tightly coupled, and they are treated as such, since the *Achilles heel* of UAVs is controlling them when they are out of range.

Section ► [UAV Architectures](#) refers to UAVs as complex engineered systems requiring combination of a diverse grouping of hardware and software technologies for their performance and effectiveness. UAVs must incorporate in their design and operation sophisticated hybrid architectures that integrate in a rather robust way and meet stringiest performance requirements. This section presents and discusses UAV system concepts migrating from the individual UAV component technologies toward a *system of systems* view addressing and answering the question *how do we put it all together?*

Section ► [UAV Health Management](#) addresses fundamental technologies and their application to UAVs that relate to monitoring and sensing strategies for health management, data acquisition and processing/analysis onboard and offboard the vehicle, fault diagnosis, and failure prognosis algorithms, as well as fault-tolerant control routines that enhance the UAV’s useful life and mitigate potential

catastrophic events. Condition-based maintenance methods and tools are also covered that take advantage of the prevailing paradigm shift from scheduled or breakdown maintenance toward maintenance practices that are based on the current condition of these assets.

Section ►[UAV Modeling, Simulation, Estimation, and Identification](#) presents a treatment of these topics from first principles to mechanisms that lead to UAV qualification and certification. Modeling and simulation techniques are essential and crucial in the design, operation, and performance assessment of UAVs. Together with estimation and identification, they are integral components of the overall UAV design process. They afford the opportunity to develop, test, and evaluate sensing, control, and cooperative control algorithms, among others. Software-in-the-loop and hardware-in-the-loop simulations are typical prerequisites to flight testing.

Section ►[MAVs and Bio-inspired UAVs](#) addresses the emerging UAV area of micro aerial vehicles (MAVs) and bio-inspired MAVs and UAVs. Research and development activities in MAVs have accelerated significantly over the past years, driven by a need for small autonomous vehicles that can execute a variety of tasks, such as intelligence, surveillance, and reconnaissance (ISR) in complex urban environments, search and rescue operations, and security and border patrol, among other applications. A substantial component of the MAV research is inspired by the exceptional flying behaviors of biological species, i.e., birds and insects. MAVs can be operated by a single person offering low weight and cost, extreme maneuvering capabilities, and rapid response times to requests for visual observations. Innovative MAV concepts are motivating advanced research and development sponsored by government and industry.

Section ►[UAV Mission and Path Planning](#) moves beyond fundamental UAV technologies to higher-level functionalities such as mission and path planning for single and multiple UAVs, as UAVs are employed to execute a variety of “missions” in both military and civilian environments. Planning and executing a mission in terms of prescribing the path or trajectory generation and other related topics are challenging problems that require careful consideration. A plethora of methods and techniques are presented, along with specific examples from the utility of mission- and path-planning technologies, suitable to assist the planner to complete an assigned mission.

Section ►[UAV Autonomy](#) addresses perhaps the most challenging and important topic that is central to unmanned aircraft performance, autonomy. Autonomous flight is a major target goal of all technologists. The ability of a UAV to take off, execute a mission, and return to its base without significant human intervention (this is the human-on-the-loop concept rather than the human-in-the-loop current requirement) promises to enhance UAV deployment in many application domains. Contributions address levels of autonomy and those specific technical challenges that must be overcome if one aims at approaching eventually “full UAV autonomy.” Hardware and software requirements for increased autonomy are discussed with emphasis on algorithms that will bestow to the UAV features of autonomy.

Section ►[UAV Sense, Detect, and Avoid Systems](#) introduces the reader to those sensing, processing, and control enabling technologies that constitute the

“sense-detect-and-avoid” technical foundations. Requirements, technologies, and future trends in sense, detect, and avoid are covered thoroughly. Sense and avoid (SAA) remains one of the key enabling capabilities required for the safe airspace integration of UAS into the global airspace system. Very little has been done to satisfy the technical and operational requirements necessary for allowing the unfettered access to civil airspace, domestically and internationally, projected to be needed by the U.S. DoD and the civil/commercial market. Although standard organizations like RTCA and EUROCAE are focusing on the problem, the current lack of specific UAS standards is clearly slowing the progress of the defense and commercial development of this exploding technology. In addition, as it relates to SAA, the impact of the lack of a SAA focused “program of record” within one or all of the federal agencies, and the associated funding that will be required and/or the lack of existing technical capability to move toward an autonomous solution believed by many to be necessary for complete, unfettered access to the national airspace appears to be slowing progress.

Section ► [Networked UAVs and UAV Swarms](#) addresses sensing, supervisory control, coordination and cooperation, computing, and communication issues that constitute the basic ingredients of safe and effective flight for multiple UAVs, emphasizing the “intelligence” requirements for safe flight and cooperative and supervisory control. The unmanned aircraft military and civilian communities are already experimenting with UAV swarms flying in formation while cooperating in the execution of specific missions. Research in networked UAVs is also proceeding via simulation studies and limited experimentation.

Section ► [UAV Integration into the National Airspace](#) reviews and presents policies, procedures, and requirements for manned aviation and relates them to existing and under development equivalent policies and requirements for unmanned aviation. Integration of unmanned aviation into the (any) national airspace requires, again, that *UAVs function as if there were a human pilot onboard*. This requirement and restriction dictates that equivalent levels of safety (ELOS) must be derived for unmanned aviation (depending on the UAV classification), in addition to reliability and fault-tolerance requirements. The major deviation from manned aviation requirements is that a controlled crash may be allowed for UAVs provided that human lives are not endangered and that collateral damage is minimized.

Section ► [UAV-Human Interfaces and Decision Support Systems](#) focuses on research findings and efforts that are underway by the UAV user community to reduce the current UAV operator workload levels. Until a satisfactory level of autonomy is achieved, the human will continue to function as an integral part of the UAV ground control station (GCS) loop. Moreover, intelligence acquired by UAVs must be communicated in an appropriate format to humans on the ground or to a mother ship for further analysis, assessment, and action. The human must intervene when new intelligence suggests that the operational profile must be changed.

Section ► [Human Factors and Training](#) addresses the topic of training of OEMs, military, and other human operators to operate, monitor, and execute a complete mission for all classes of platforms currently available to the military and other sectors.

Section ► **UAV Logistics Support** refers to UAV lifetime operation support; scheduling issues; delivery of goods and services; maintenance, testing, and fielding of UAVs; design and operation for reliability, safety, availability, and maintainability; logistics for ground station support and mobile UAV platforms; and potentially human operator support.

Section ► **UAV Applications** offers a “taste” of possible civil and public domain applications in which single or multiple UAVs may be used. As the roadmap of the UAS integration into the global airspace is formulated amidst current obstacles and skepticism, it is only logical to consider that the spectrum of possible applications will be much wider in the years to come. At the present stage, UAVs have either been considered for or have seen (very) limited utilization in a wide range of applications that include but are not limited to power line inspection; pipeline inspection; ship inspection; mine inspection; dam inspection; anomaly detection/prevention; early fire detection and forest protection; hazard monitoring; traffic monitoring; environmental monitoring; search and rescue operations; emergency response; border patrol; harbor patrol; police surveillance; aerial photography; SWAT support; imaging and mapping; intelligence, surveillance, and reconnaissance (ISR); chemical spraying; crop dusting; night vision; and entertainment industry and filming.

Section ► **Social and Ethical Implications** offers comprehensive and detailed information aiming at answering the question *What are the social, legal and ethical concerns associated with the use of UAVs.*

The *Epilogue* concludes the handbook.

In what follows, UAVs are put in the context of “unmanned systems.” The term UAV is used loosely and freely representing (in the proper context) the also used terms like RPA, UAS, and RPAS.

2.2 UAVs in the Context of Unmanned Systems

Unmanned systems (US) are typically known as powered vehicles (air, maritime, and ground) that do not carry a human operator, can be operated autonomously or remotely, can be expendable or recoverable, and can carry a variety of payloads depending on their type, functionality, operational characteristics, and mission objectives.

Unmanned vehicles (UVs), the primary component of US, have captivated the imagination of humans for centuries. For example, the idea for a “flying machine” was first conceived close to 2,500 years ago, in ancient Greece and China! Pythagoras, Archimedes, and others studied the use of autonomous mechanisms for a variety of applications. The first known autonomous flying machine has been credited to Archytas of Tarentum in South Italy, known as Archytas the Tarantine. Archytas has been referred to as Leonardo da Vinci of the ancient world. In 425 B.C., he built a mechanical bird, which he called *the pigeon*. According to Cornelius Gellius in his *Noctes Atticae*, the bird was made of wood, nicely balanced with weights and flew using air (most likely steam) enclosed in its stomach. It is alleged that Archytas’ pigeon flew about 200 m before falling to the ground, once all energy

was used. During the same era in a different part of the Ancient World, China, at about 400 B.C., the Chinese were the first to document the idea of a vertical flight aircraft. The earliest version of the Chinese top consisted of feathers at the end of a stick. The stick was spun between the hands to generate enough lift before released into free flight. Several centuries later, Leonardo da Vinci, in 1483 A.D., designed an aircraft capable of hovering, called aerial screw or air gyroscope. Da Vinci also devised a mechanism bird in 1508 A.D. that would flap its wings by means of a double crank mechanism as it descended along a cable. A separate chapter is dedicated to the history of unmanned aviation later on in this section.

Nowadays, the world is witnessing an unprecedented flurry of US-related research and development activity. Pioneers in this area are experimenting with novel concepts and set the stage for even greater explosion in US development and applications.

Unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs), unmanned surface vehicles (USVs), and unmanned underwater vehicles (UUVs) share common features in terms of their fundamental architectures, propulsion, communications, and those technologies that aim to improve their autonomy attributes. These types of vehicles are capable of many different applications in the fields of law enforcement, environmental monitoring, disaster relief, and recovery operations. They can also undertake routine and persistent surveillance of a nation's internal and external borders as well as critical oil and gas infrastructure inspection, to name but a few applications in which such vehicles may be utilized.

There are no significant differences for human interaction issues with the exception possibly in response times required by a given mission and operator culture. Major differences are observed in the operational environment – physical environment communications and vehicle dynamics. Payload types are also significantly different depending upon sensing requirements, mission objectives, etc.

The UAV segment has experienced the greatest growth, driven primarily by the military community and to a lesser extent by border security, environmental monitoring, and other application domains. This picture seems to change, and it is projected that UAVs will dominate civilian and public domain applications.

The UGV segment has exploded recently in response to urgent military needs. Future UGV missions will include perimeter surveillance, logistics support, rescue operations, communications relay, among other applications.

UUVs and USVs are still in their infancy. Unlike the UAV and UGV segments, unmanned maritime systems are funded at much lower levels in the USA and worldwide. The UUV segment is maturing more rapidly due to the commercial use of these systems by the offshore oil and gas industry. Despite challenges stemming from the current world economic conditions and others due to unmanned systems “growth pains,” demand for UUVs is high and will continue to grow over the coming years.

Unmanned systems in general raise new issues of improved autonomy, communications, interoperability, propulsion and power, man-machine interfaces, and manned-unmanned teaming that will challenge the R&D community. The need to maintain simplicity in US is also an ongoing challenge.

When focusing on unmanned system autonomy, improved system autonomy requires and will require new and innovative incipient failure detection, prediction of the useful life of the failing vehicle components, and innovative fault-tolerant control strategies. There is also a need for an open architecture (OA) that will accommodate attributes of interoperability, modularity, plug and play, and flexibility across unmanned platforms. By leveraging OA and open interfaces, the US community can overcome problems associated with proprietary robotic system architectures. These advances are bound to lower platform costs, open new application domains, and offer the opportunity for the deployment of multiple (mixed) types of unmanned systems for improved mission effectiveness.

Overall, the state of the unmanned system technology in general is mature, while the unmanned aviation and unmanned aircraft segment by itself or in combination with ground and maritime vehicles will play an increasing role in both the military and civilian domains.

In the sequel, a detailed outline of each section of the handbook is given, which includes a summary of the chapters comprising each section. Chapters are summarized in the order they appear in each section of the handbook.

2.3 Outline of the Handbook Sections

2.3.1 UAV Design Principles

Section ▶[UAV Design Principles](#) includes chapters ▶[Computational and Experimental Design of a Fixed-Wing UAV](#), ▶[Payload Design of Small UAVs](#), ▶[Small UAV Design Development and Sizing](#), ▶[Systematic Design Methodology and Construction of Micro Aerial Quadrotor Vehicles](#), and ▶[Dexterous UAVs for Precision Low-Altitude Flight](#).

Chapter ▶[Computational and Experimental Design of a Fixed-Wing UAV](#) by Ryaciotaki-Boussalis and Guillaume presents a comprehensive methodology for the design and manufacture of all aspects of a prototype fixed-wing UAV, which also allows for utilizing multi-segmented flight control surfaces on the wings of such UAVs. Adding multiple segments to UAV wings creates smaller control surfaces; thus, refined adjustments to UAV performance while airborne may be accomplished.

Chapter ▶[Payload Design of Small UAVs](#) by Gruber, Kwon, Hager, Sharma, Yoder, and Pack discusses payload design issues for small UAVs, which are essential to overcome various small UAV constraints imposed by stringent weight, power, and volume. The efficacy of design principles is demonstrated with the help of the payloads for a fixed-wing UAV developed by the Center for Unmanned Aircraft Systems Research at the U.S. Air Force Academy, and it is shown that the system requirements for specific applications are closely related to those for other small UAV applications.

Chapter ▶[Small UAV Design Development and Sizing](#) by Brandt describes a rationale for selecting a low-risk UAV configuration and a methodology for sizing and optimizing the shape of both fueled and electric-powered aircraft.

Three examples of low-risk UAV configurations are shown to be essentially similar from the point of view of optimization and sizing. The sizing methodology for fueled aircraft is described in detail, and a similar one for electric-powered aircraft is described to the extent that it differs from that for fueled aircraft. Typical values of technology-related parameters used in sizing calculations are given, followed by an example of a small UAV designed using the proposed methodology.

Chapter ►[Systematic Design Methodology and Construction of Micro Aerial Quadrotor Vehicles](#) by Phang, Li, Chen, and Lee presents guidelines to optimally and systematically design and construct an ultralight micro aerial quadrotor vehicle, including details of the steps to design a stable quadrotor with not more than 50 g take-off weight and flight duration of 8 min. The methodology covers, sequentially, structural analysis of the micro quadrotor air frame, design in a 3-D virtual simulator, design of the avionic system, controller implementation via simulation, and component fabrication and implementation. Finally, performance is evaluated, tested, and confirmed with real flight missions.

Chapter ►[Dexterous UAVs for Precision Low-Altitude Flight](#) by Jiang and Voyles follows the concept of force closure (a term from the dexterous manipulation community) to introduce a new type of dexterous six degree-of-freedom (6 DOF) UAV, which provides the unique capability of being able to resist any applied wrench, or generalized force-torque, providing more precise control during low-altitude flight. The major challenge that needs to be overcome in such designs is that low-altitude flight usually introduces ground-effect disturbances and other backwash issues, while in the new field of aerial mobile manipulation, it often includes close to structures operations for inspection or manipulation purposes. The presented design differs considerably from typical helicopters or quadrotors, which cannot instantaneously resist or apply an arbitrary force in the plane perpendicular to the rotor axis, thus, making such designs inadequate for complex mobile manipulation tasks.

2.3.2 UAV Fundamentals

Section ►[UAV Fundamentals](#) includes chapters ►[Kinematics and Dynamics of Fixed-Wing UAVs](#), ►[Dynamic Model for a Miniature Aerobatic Helicopter](#), ►[Quadrotor Kinematics and Dynamics](#), ►[Dynamics and Control of Flapping-Wing MAVs](#), and ►[Dynamic Modeling and Control of an Autonomous Quadrotor](#).

Chapter ►[Kinematics and Dynamics of Fixed-Wing UAVs](#) by Dobrokhodov provides a thorough review of the fundamentals required for accurate mathematical modeling of flight of a fixed-wing UAV, including kinematics and dynamics of motion, and transformation of forces and moments acting on the airplane. The main objective is to familiarize the reader with the “kinematics–dynamics–actions” triad as it applies to a generic fixed-wing UAV. Emphasis is given to the understanding of reference frames and their dynamics as this is essential for the design of the UAV guidance, navigation, and control systems.

Chapter ▶ [Dynamic Model for a Miniature Aerobatic Helicopter](#) by Gavrillets presents a nonlinear dynamic model of a miniature aerobatic helicopter following a component buildup to derive the model using simplified analytical expressions for the component forces and moments. Key parameters are estimated based on flight-test experiments. The derived model is used to design control logic for aerobatic maneuvers performed entirely under computer control.

Chapter ▶ [Quadrotor Kinematics and Dynamics](#) by Powers, Mellinger, and Kumar presents an overview of the rigid body dynamics of a quadrotor, starting from describing the Newton-Euler equations that govern the quadrotor motion. This is followed by the derivation of a linear controller based on a linearized model of the dynamics and a nonlinear controller derived from the original dynamic model. Experimental results that illustrate the dynamics and control of small quadrotors are also presented.

Chapter ▶ [Dynamics and Control of Flapping Wing MAVs](#) by Doman, Oppenheimer, and Sigthorsson focuses on presenting methods suitable for hover-capable flapping-wing vehicles. This is a challenging problem as flapping-wing motion is characterized by seven wing motion parameters like wing stroke amplitude, symmetric and asymmetric frequency, bias, upstroke and downstroke angle-of-attack limits, and wing stroke plane inclination. A method is presented that enables the designer to estimate how the seven wing motion parameters change in response to changes in the control mechanism position, followed by a method for estimating the combined effect of fuselage motion and flapping-wing motion. Control design considerations and simulation methods are also discussed.

Chapter ▶ [Quadrotor Kinematics and Dynamics](#) by Elkaim, Pradipta Lie, and Gebre-Egziabher presents two easily reconfigurable system architectures for guidance, navigation, and control of small UAVs. The presented system architectures integrate a low-cost inertial measurement unit, a GPS receiver, and a triad of magnetometers to generate a navigation solution which, in turn, is used in the guidance and control algorithms. The full system architecture – the hardware, software, and algorithms – is included for completeness. Hardware-in-the-loop simulation and flight-test results documenting the performance of these two systems are given.

2.3.3 UAV Sensors and Sensing Strategies

Section ▶ [UAV Sensors and Sensing Strategies](#) includes chapters ▶ [Sensors for Missions](#), ▶ [Inertial Sensor-Based Simultaneous Localization and Mapping for UAVs](#), ▶ [UAV Localization Using Inertial Sensors and Satellite Positioning Systems](#), and ▶ [Data Fusion and Tracking with Multiple UAVs](#).

Chapter ▶ [Sensors for Missions](#) by Mejias, Lai, and Bruggemann sets the tone for sensors used on UAVs that are assigned complex missions. The sensor suite onboard a UAV is tightly coupled with payload capabilities, as payload dictates UAV usability and market value. However, advances in miniaturization of electronics are enabling replacement of multiprocessing, power-hungry general-purpose processors with more integrated and compact electronics that contribute to more onboard

sensors. Several common payload sensors are described along with their usefulness to solve real-world problems.

Chapter ▶ [Inertial Sensor-Based Simultaneous Localisation and Mapping for UAVs](#) by Bryson and Sukkarieh provides an overview of algorithms for inertial sensor-based simultaneous localization and mapping (SLAM) within the context of UAVs, using the extended Kalman filter (EKF) and the extended information filter (EIF) due to their ease of understanding, applicability to online implementation, and prevalence in airborne localization applications outside of SLAM.

Chapter ▶ [UAV Localisation Using Inertial Sensors and Satellite Positioning Systems](#) by Bryson and Sukkarieh provides an overview of UAV localization with a focus on aided inertial localization, that is, algorithms for fusing data from, for example, satellite positioning systems, barometric sensors, and magnetometers with inertial sensors to provide real-time position and orientation. An example implementation of aided inertial localization on a UAV is presented as a tutorial to understand key concepts in airborne localization and as a basic guide toward more complicated implementations.

Chapter ▶ [Data Fusion and Tracking with Multiple UAVs](#) by Ridley, Upcroft, and Sukkarieh describes decentralized data fusion (DDF) algorithms for a team of multiple autonomous platforms. It is shown how through the DDF algorithms, each platform can maintain a consistent global solution from which decisions may be made. The overall system design is detailed, providing insight into the overall complexity of implementing a robust DDF system for use in information gathering tasks in outdoor UAV applications.

2.3.4 UAV Propulsion

Section ▶ [UAV Propulsion](#) includes chapter ▶ [Power Managements of a Hybrid Electric Propulsion System Powered by Solar Cells, Fuel Cells, and Batteries for UAVs](#).

Chapter ▶ [Power Managements of a Hybrid Electric Propulsion System Powered by Solar Cells, Fuel Cells, and Batteries for UAVs](#) by Lee, Park, and Kim provides a very comprehensive description of the power management of a UAV hybrid electric propulsion system. It considers three electric propulsion systems with solar cells, fuel cells, and battery power sources, which are designed and constructed to share the same operation voltage range and connect to the power bus without the need for additional converters or controllers. MATLAB/Simulink-based modeling is followed for each power source, and the component models are verified with published data from the manufacturers as well as flight-test data. The component models are integrated into one power system model that is used to simulate the UAV electric propulsion system. For fuel cells and batteries, the simulation process is verified via comparison between simulation results and available UAV flight-test results. Two types of power control logic are investigated, passive and active. The passive power management simulation shows that the behavior of each power source and its integrated system are adequate for the overall UAV flight envelope. The active power management simulation yields more efficient power distribution

and better system safety than passive power management. Additional simulations of a hybrid electric power system allow for estimation of the output behavior of the power source.

2.3.5 UAV Control

Section ▶UAV Control includes chapters ▶Linear Flight Control Techniques for Unmanned Aerial Vehicles, ▶Nonlinear Flight Control Techniques for Unmanned Aerial Vehicles, ▶Adaptive Control of Unmanned Aerial Vehicles: Theory and Flight Tests, and ▶Robust and Adaptive Control Methods for Aerial Vehicles.

Chapter ▶Linear Flight Control Techniques for Unmanned Aerial Vehicles by How, Frazzoli, and Chowdhary presents an overview of linear flight control and guidance methods for UAVs, starting with a discussion of rotation matrices and UAV kinematic equations and followed by derivation of the 6 DOF UAV equations of motion. Equations of motion are then linearized, and several linear multi-loop closure techniques for UAV guidance and control are discussed.

Chapter ▶Nonlinear Flight Control Techniques for Unmanned Aerial Vehicles by Chowdhary, Frazzoli, How, and Liu discusses nonlinear and adaptive control techniques that are often used to improve UAV performance and reliability. Such techniques are actively being studied to handle nonlinear aerodynamic and kinematic effects, actuator saturations and rate limitations, modeling uncertainty, and time-varying dynamics. An overview of tools and techniques used for designing nonlinear flight controllers is presented along with a summary of Lyapunov stability theory. In detail, the nonlinear control techniques that are considered include gain scheduling, model predictive control, backstepping, dynamic inversion-based control, model reference adaptive control, and model-based fault-tolerant control.

Chapter ▶Adaptive Control of Unmanned Aerial Vehicles: Theory and Flight Tests by Kannan, Chowdhary, and Johnson focuses on adaptive control of unmanned aircraft that are underactuated systems. A 6 DOF flight control algorithm is derived that can track both position and attitude trajectories. Approximate inverse models for vehicle attitude and position dynamics are used for feedback linearization leading to an inner loop that tracks attitude and angular rate and an outer loop that tracks position and velocity commands. A single adaptive element is used to compensate for inversion errors (uncertainty) in both loops. A key challenge in realizing an adaptive control design on real aircraft is dealing with actuator magnitude and rate saturation. Such saturation elements cannot be easily captured in inverse models and leads to incorrect learning in the adaptive element during periods of saturation. A mechanism to exactly remove such incorrect learning is provided. Additionally, nonlinear reference models are introduced to mitigate the risks of the closed-loop system entering regions of the flight envelope that result in loss of controllability. The resulting adaptive controller accepts trajectory commands comprising of desired position, velocity, attitude, and angular velocity and produces normalized actuator signals required for flight control. A modification to the baseline adaptive control system is also provided that enables long-term retention

of the uncertainty approximation within the adaptive element. This architecture is validated through flight tests on several fixed-wing and rotorcraft UAVs.

Chapter ▶[Robust and Adaptive Control Methods for Aerial Vehicles](#) by Lavretsky includes a collection of reliable, efficient, robust, and adaptive control methods for aerial vehicles. It begins with a brief overview of flight dynamics models suitable for flight control design. The first control design method represents the well-understood and now-classical linear quadratic regulator (LQR) command tracker, with proportional-integral (PI) feedback connections, serving as the backbone of all other subsequent methods. The main intent is to demonstrate the design of predictable, formally justified, yet numerically efficient flight controllers, with an LQR PI baseline and with a direct model reference adaptive control (MRAC), as an augmentation to the baseline. Through extensive simulation, analysis, and actual flight testing, it is shown that (LQR PI + adaptive) – controllers provide robust stability and maintain tracking performance, when operated in the presence of “unknown unknowns” in the vehicle dynamics and in often “unfriendly” operational environment. All presented control methods are flight tested and validated on a wide range of aerial vehicles.

2.3.6 UAV Communication Issues

Section ▶[UAV Communication Issues](#) includes chapters ▶[Problem of UAV Communications](#), ▶[Cognitive Networking for UAV Swarms](#), ▶[Layered Approach to Networked Command and Control of Complex UAS](#), and ▶[Cognitive Radio Architectures for Unmanned Aircraft Systems](#).

Chapter ▶[Problem of UAV Communications](#) by Heppe summarizes the current consensus view (as of the end of 2012) regarding critical data flows for UAS, control link performance requirements, potential frequency bands that can satisfy the relevant technical as well as regulatory constraints, challenges that must be overcome to ensure reliable operation, and possible data link design principles that could lead to a safe and workable implementation. Both line-of-sight and beyond-line-of-sight data links are addressed. It is shown that the challenges to be faced and overcome are significant, but a safe and workable implementation appears to be achievable through data link diversity and other error correction and error recovery techniques.

Chapter ▶[Cognitive Networking for UAV Swarms](#) by Brown, McHenry, and Jaroonvanichkul examines the enabling role of cognitive radio technologies for UAS to access more spectrum needed for flight operations. After requirements are set, different architecture choices available to the UAS cognitive radio designer are presented, including the communication architecture, the spectrum awareness techniques for assessing what spectrum is available, and the spectrum access techniques for deciding which available spectrum to use. Information in this chapter is relevant for the development of future UAS rules and standards.

Chapter ▶[Layered Approach to Networked Command and Control of Complex UAS](#) by Elston, Stachura, Dixon, Agrow, and Frew discusses different networking hardware, protocols, and sensors that, when combined, create a diverse and complex

UAS through a layered design approach with modular supporting software. It is shown that critical software components, such as service discovery, simplify the inclusion of a diverse set of subsystems and sensors. Maintaining the modularity of these software components ensures that the system can be expanded while requiring minimal software changes. A detailed description of a system is presented, which enabled flight operations of a multi-vehicle unmanned aircraft system for performing targeted, in situ sampling of supercell thunderstorms during the 2010 VORTEX2 field campaign.

Chapter ▶ [Cognitive Radio Architectures for Unmanned Aircraft Systems](#) by Wietfeld and Daniel discusses challenges and solution approaches with respect to self-organized and robust multi-UAV systems. The focus is on designing novel service-oriented system architecture to achieve a flexible deployment of UAS. The design aspects of mobility control algorithms addressing the competing requirements of communication reliability and spatial distribution of UAV swarms are demonstrated using aerial sensor networks and ad hoc aerial relay networks. Solution approaches rely on agent-based control of UAV swarms operating autonomously in dynamically changing environments. Different communication-aware algorithms for microscopic as well as macroscopic mobility control are presented, such as Cluster Breathing, Smart Cube, Potential Fields, and Role-Based Connectivity Management. The chapter also discusses how the network and application task specific performance requirements are met with the proposed mobility control algorithms even in the cases of temporarily unavailable communication links. The self-healing capabilities therefore allow for reliable networking and control of aerial robot swarms in diverse use cases, such as emergency response, environmental monitoring, and ad hoc network provisioning.

2.3.7 UAV Architectures

Section ▶ [UAV Architectures](#) includes chapters ▶ [HDRC3: A Distributed Hybrid Deliberative/Reactive Architecture for Unmanned Aircraft Systems](#), ▶ [Classification of Multi-UAV Architectures](#), and ▶ [Operator Interaction with Centralized Versus Decentralized UAV Architectures](#).

Chapter ▶ [HDRC3: A Distributed Hybrid Deliberative/Reactive Architecture for Unmanned Aircraft Systems](#) by Doherty, Kvarnstrom, Wzorek, Rudol, Heintz, and Conte presents a distributed UAS architecture, also extendable for use with multi-platform systems, providing full integration of both low and high autonomy. It covers the full spectrum of functionalities required for operation in missions requiring high autonomy. Specific interfaces and languages are introduced, which provide seamless transition between deliberative and reactive capability and reactive and control capability. Hierarchical concurrent state machines are introduced as a real-time mechanism for specifying and executing low-level reactive control. Task specification trees are introduced as both a declarative and procedural mechanism for specification of high-level tasks. Task planners and motion planners are described which are tightly integrated into the architecture. Generic middleware

capability for specifying data and knowledge flow within the architecture based on a stream abstraction is also described. Emphasis is placed on the robust integration and interaction between these diverse functionalities using a principled architectural framework. The architecture has been empirically tested in several complex missions, some of which are also described in this chapter.

Chapter ►[Classification of Multi-UAV Architectures](#) by Maza, Ollero, Casado, and Scarlatti presents a classification of different schemes for multiple UAV cooperation considering the coupling between the vehicles and the type of cooperation. It also considers UAVs that are networked with other elements in the environment to support their navigation and, in general, their operation. The chapter provides a theoretical framework but also emphasizes practical field demonstrations involving aerial vehicles.

Chapter ►[Operator Interaction with Centralized Versus Decentralized UAV Architectures](#) by Cummings starts from the fact that there are major attempts to streamline UAV operations and reduce staffing in order to invert the current many-to-one ratio of operators to vehicles. It then introduces architectural schemes in the context of optimizing operator workload and efficiency. In centralized multiple UAV architectures, a single operator, who requires significant cognitive resources, interacts with and oversees every UAV in the network. In decentralized multiple UAV architectures, an operator interacts with an automated mission and payload manager, which coordinates a set of tasks for a group of highly autonomous vehicles. While a single operator can maintain effective control of a relatively small network of centralized UAVs, decentralized architectures are more scalable, particularly in terms of operator workload, and more robust to single points of failure. However, in terms of operator workload, the ultimate success of either a centralized or decentralized UAV architecture is not how many vehicles are in the network per se but rather how many tasks the group of vehicles generates for the operator and how much autonomy is onboard these vehicles. Task-based control of UAV architectures with higher degrees of autonomy (i.e., decentralized networks) can mitigate cognitive overload and reduce workload. Mutually exclusive boundaries for humans and computers in multiple UAV systems should not be the goal of designers for either centralized or decentralized architectures, but rather more effort needs to be spent in defining mutually supportive roles such that humans and computers complement one another.

2.3.8 UAV Health Management Issues

Section ►[UAV Health Management Issues](#) includes chapters ►[Integrated Vehicle Health and Fault Contingency Management for UAVs](#), ►[Automated Failure Effect Analysis for PHM of UAV](#), ►[Prognostics Applied to Electric Propulsion UAV](#), ►[Actuator Fault Detection in UAVs](#), ►[Experimental Validation of Fault Detection and Diagnosis for Unmanned Aerial Vehicles](#), ►[Fault Detection and Diagnosis for NASA GTM UAV with Dual Unscented Kalman Filter](#), and ►[Fault Diagnosis of Skew-Configured Inertial Sensor System for Unmanned Aerial Vehicles](#).

Chapter ► [Integrated Vehicle Health and Fault Contingency Management for UAVs](#) by Roemer and Tang presents various concepts for integrating real-time vehicle health assessment and fault contingency management technologies for UAVs. The presented Integrated Vehicle Health Management (IVHM) and Automated Contingency Management (ACM) system architecture is shown to support real-time, onboard health state assessment and fault management so that UAVs can enjoy greater autonomy and survivability during anomalous operating conditions. Selected real-time system identification and automated health assessment algorithms are presented that can readily identify the dynamics and performance limitations of degraded UAV systems. Additionally, a high-level mission adaptation approach is presented to estimate the safe flight operating envelope after the occurrence of faults. Reconfigurable fault-tolerant control techniques that directly utilize the identified UAV subsystem dynamic models have been developed and tested in simulation. Finally, proof-of-concept demonstrations are presented using NASA engine and aircraft dynamic models with simulated engine and actuator faults.

Chapter ► [Automated Failure Effect Analysis for PHM of UAV](#) by Snooke describes how model-based simulation can be employed to automatically generate the system-level effects for comprehensive sets of component failures on systems within the aircraft. The results of the simulation can be used in several ways. They can be used to produce a system-level failure modes and effects analysis (FMEA) for aircraft systems. They can be used to identify the sensors necessary to discriminate remotely between different failures on the aircraft. Once a set of sensors have been chosen for placement on the vehicle, the simulation results can also be used to generate diagnostic and prognostic software for deployment on the vehicle. Using automated FMEA, safety analysis software is more efficient than doing the same work without software and also provides a guaranteed level of performance. Using the results of this analysis can provide sensor selection and diagnostic capability while retaining some of the benefits of rule-based diagnostic systems. Alternative model-based techniques have been widely used to create diagnostic systems in a variety of domains, and these approaches are compared with the diagnostic capability provided by a failure effect-oriented technique from the perspective of the UAV application.

Chapter ► [Prognostics Applied to Electric Propulsion UAV](#) by Goebel and Saha explores the technical underpinnings of how to perform prognostics and shows an implementation on the propulsion of an electric UAV. An accurate run-time battery life prediction algorithm is of critical importance to ensure the safe operation of the vehicle if one wants to maximize in-air time. Current reliability-based techniques turn out to be insufficient to manage the use of such batteries where loads vary frequently in uncertain environments. A particle filter is shown as the method of choice in performing state assessment and predicting future degradation. The method is then applied to the batteries that provide power to the propeller motors.

Chapter ► [Actuator Fault Detection in UAVs](#) by Ducard is dedicated to actuator fault detection systems for UAVs, with two main requirements: real-time capability and modularity. After defining the terminology employed in this field, it first reviews some commonly used techniques in FDI systems, followed by presenting briefly the mathematical model of a UAV that serves as a basis for the design of two actuator

FDI systems. The first method presents and illustrates the multiple-model approach, whereas the second method presents an FDI system, which is based on a single model. Both methods are enhanced by a mechanism that actively tests actuators in order to efficiently detect and isolate actuator faults and failures. Advantages and drawbacks of each method are stated issues of robustness are discussed against model uncertainties and external perturbation. In addition, aspects of computational load are addressed. The FDI systems applied to a realistic model of an unmanned aircraft, and the performance of the methods is shown in simulation.

Chapter ► [Experimental Validation of Fault Detection and Diagnosis for Unmanned Aerial Vehicles](#) by Chamseddine, Hadi Amoozgar, and Zhang investigates the problems of fault detection, diagnosis, and fault-tolerant control for UAVs. It first presents a detailed overview of existing experimental research focusing on fixed-wing as well as rotorcraft UAVs including single-rotor and multi-rotor helicopters. It then discusses three Kalman filters employed for actuator fault detection and diagnosis, namely, the unscented Kalman filter, the two-stage Kalman filter, and the adaptive two-stage Kalman filter. The three filters are experimentally applied to a quadrotor helicopter UAV test bed, and results are shown, compared, and discussed.

Chapter ► [Fault Detection and Diagnosis for NASA GTM UAV with Dual Unscented Kalman Filter](#) by Zhang investigates a simultaneous state and parameter estimation-based fault detection and diagnosis (FDD) scheme applied to a realistic nonlinear six degree-of-freedom fixed-wing unmanned aerial vehicle (UAV) model, the NASA GTM (generic transport model). By introducing partial loss faults in actuators into the NASA GTM, a dual unscented Kalman filter (DUKF) algorithm is applied for the purpose of online estimation of both flight states and fault parameters. A Bayesian rule is then used for detection and isolation decision making. The developed FDD scheme is implemented in both the nonlinear GTM and the linear parameter-varying (LPV) representation of the nonlinear GTM. Simulation results show satisfactory results for detecting and diagnosing the control effectors faults.

Chapter ► [Fault Diagnosis of Skew-Configured Inertial Sensor System for Unmanned Aerial Vehicles](#) by Yoon, Kim, Bae, Kim, and Kim presents fault detection and isolation scheme to handle three successive faults in the skew-configured inertial sensors of an unmanned aerial vehicle. The skew-configured inertial sensors are composed of the primary inertial measurement unit and the redundant secondary inertial measurement unit. Since small unmanned aerial vehicles are restricted by cost and payload space, the secondary small and low-cost inertial measurement unit is installed with a skewed angle in addition to the primary inertial measurement unit. In the hybrid fault detection and isolation scheme, a parity space method and an in-lane monitoring method are combined to increase system tolerance to the occurrence of multiple successive faults during flight. The first and second faults are detected and isolated by the parity space method. The third fault is detected by the parity space method and isolated by the in-lane monitoring method based on the discrete wavelet transform. Hardware-in-the loop tests and flight experiments with a fixed-wing unmanned aerial vehicle are performed to verify the performance of the proposed fault diagnosis scheme.

2.3.9 UAV Modeling, Simulation, Estimation, and Identification

Section ► [UAV Modeling, Simulation, Estimation, and Identification](#) includes chapters ► [Flight Dynamics Modeling of Coaxial Rotorcraft UAVs](#), ► [Modeling of a Micro UAV with Slung Payload](#), and ► [Command and Control of Autonomous Unmanned Vehicles](#).

Chapter ► [Flight Dynamics Modeling of Coaxial Rotorcraft UAVs](#) by Wang, Cui, Chen, and Lee aims at presenting a thorough approach related to systematic modeling of UAV flight dynamics by talking about the UAV model formulation, parameter identification, and model verification. The presented methodology of UAV model formulation and parameter identification is based on modeling two kinds of coaxial helicopters. This is a challenging problem as modeling of coaxial helicopters, despite the common governing kinematic and dynamic principles, deserves special attention due to their distinctive mechanical structure and aerodynamics behavior. The modeling procedures and parameter identification technique presented in this chapter may also serve as a guideline for modeling other types of aerial vehicles.

Chapter ► [Modeling of a Micro UAV with Slung Payload](#) by Feng, Rabbath, and Su derives a mathematical and simulation model of a micro UAV carrying a payload. When the load is slung underneath a UAV by cable, the flight dynamics of the UAV are altered, which makes the stability of the UAV disturbed. The unstable oscillation degrades UAV performance, and the accurate placement of the load will be affected. Unlike external disturbances, the negative effects are related to the characteristics of the UAV and the payload. In order for the UAV to be able to adopt the change of the system dynamics and reduce the effects caused by the swing of the load, one modeling method of a micro UAV with single slung payload is addressed. The slung payload is treated as a pendulum-like mass point, and the coupling factors between the UAV and the payload are considered. The derived model may be used to estimate the negative effects acting on the UAV, and it may also be used to estimate the trajectory of the load, which allows for improving the accuracy of placement of the loads.

Chapter ► [Command and Control of Autonomous Unmanned Vehicles](#) by Scheidt describes a series of experiments, including 2011 experiments in which 16 fully autonomous unmanned vehicles, including 9 unmanned air vehicles, were used to simultaneously support mounted, dismounted, and maritime users. During these experiments, users provided abstract mission-level ISR needs to the “vehicle cloud.” These needs were interpreted by the vehicles, which self-organized and efficiently achieved the user’s objectives. The starting point is using information theory to examine UAV command and control (C2). The information-theoretic analysis provides a justification and use cases for *autonomous* UAVs. An autonomous unmanned vehicle system “organic persistent intelligence, surveillance, and reconnaissance” (OPISR) is introduced. OPISR is an autonomous unmanned vehicle system that combines the immediate response to tactical ISR needs provided by organic assets with the time-on-station, minimal logistics provided by persistent unmanned systems. OPISR autonomous vehicles collectively interpret real-time tactical intelligence, surveillance, and reconnaissance (ISR) objectives submitted by any number of disadvantaged users, gather the required ISR data, and return the

needed intelligence directly to the affected user. OPISR is an ad hoc, decentralized system that requires no central base or authority and is capable of functioning in communications-denied environment.

2.3.10 MAVs and Bio-inspired UAVs

Section ▶[MAVs and Bio-inspired UAVs](#) includes chapters ▶[Micro Air Vehicles](#), ▶[Survey of the Human-Centered Approach to Micro Air Vehicles](#), ▶[Development of Insect-Sized MAVs](#), ▶[Flapping-Wing-Propelled Micro Air Vehicles](#), ▶[Inventing a Biologically Inspired, Energy-Efficient Micro Aerial Vehicle](#), and ▶[Issues Surrounding Communications with Micro Aerial Vehicles](#).

Chapter ▶[Micro Air Vehicles](#) by R. C. Michelson addresses challenges related to the design of MAVs. Such challenges span across aerospace, electrical, mechanical, and computer engineering because of flight regime in which these tiny aircrafts operate. Aerospace designers must contend with issues surrounding low Reynolds number flight, while electrical and mechanical designers are concerned with issues of energy storage, behavior of materials at small scales, and non-scaling items. The missions at which MAVs excel demand increased levels of autonomy, forcing computer engineers to create innate onboard intelligence exhibiting high bandwidth and superior abilities to interpret obstacle-rich environments not usually encountered by larger flying machines.

Chapter ▶[Survey of the Human-Centered Approach to Micro Air Vehicles](#) by S. Michelson presents a detailed overview of some of the Human Systems Integration (HSI) and Human Factors Engineering (HFE) issues involved with MAVs. The importance of a total systems engineering approach to MAV design, how MAVs fit into commonly accepted Human Systems Integration domains, and an exposure of some emerging issues with MAVs that require further research are discussed. The unique attributes of MAVs in terms of their size and control methods, combined with the challenges of the dynamic operational environments where they are deployed, represent HFE issues exclusive to the MAV platform that require special consideration. The importance of designing for the human operator is paramount for successful outcomes with MAV platforms. Specifically highlighted are some areas where currently researched HFE issues are particularly applicable to MAVs as opposed to large-scale systems.

Chapter ▶[Development of Insect-Sized MAVs](#) by Sunada, Liu, Tokutake, and Kubo describes a prototype bio-inspired flapping MAV with flexible wings, with a specific focus on the flexible-wing aerodynamics. The flapping-wing MAV has a weight of 2.4–3.0 g and a wingspan of 10–12 cm, which is comparable to hawk moths and hummingbirds. The MAV's flexible-wing aerodynamics is analyzed by combining an *in-house* computational fluid dynamics (CFD) method and wind tunnel experiments (EXP). In addition, fixed-wing and rotary-wing MAVs with elements that enable the miniaturization of an aerial vehicle are introduced.

Chapter ▶[Flapping-Wing-Propelled Micro Air Vehicles](#) by Jones and Platzer presents a brief history of the major discoveries in the scientific exploration of

flapping-wing flight. This is followed by a short review of the basic concepts of lift generation on wings in low-speed, steady flight, which leads into a discussion of the generation of thrust due to the flapping of wings. The aerodynamics of single flapping wings in forward and hovering flight, of flapping tandem and biplane wings, and of dual wings using the clap-and-fling effects are discussed. The chapter concludes with an overview of the major characteristics of five representative flapping-wing-propelled MAVs developed to date, including models developed at the AeroVironment Company, Naval Postgraduate School, Wright State University, and Delft University.

Chapter ►[Inventing a Biologically Inspired, Energy-Efficient Micro Aerial Vehicle](#) by Ratti and Vachtsevanos introduces a novel framework for the design and control of an MAV, where the conceptual design is based on biologically inspired principles and emulates a dragonfly. The chapter addresses the design and control features of the proposed design and gives an overview on the developmental efforts toward the prototyping of the flyer. The potential applications for such a high-endurance vehicle are numerous, including air deployable mass surveillance in cluster and swarm formations. The disposability of the vehicle helps in battlefield deployment as well, where such an MAV is made available to soldiers for proximity sensing and threat level assessment. Other applications include search and rescue operations and civilian law enforcement.

Chapter ►[Issues Surrounding Communications with Micro Aerial Vehicles](#) by C. Michelson seeks to answer many of the communications link questions that MAV designers have and gives a high-level overview of the factors that affect MAV data and control links. Challenges related to communications with MAVs are because of their size and very limited payload capabilities. Limited payload capacity leads to considerable constraints on power sources, sensors, and communication systems. Power sources are by far the most weight-inefficient components on an MAV. MAV designers are forced to look elsewhere to optimize their designs. The best way to do so in lieu of focusing on improving battery technology is to optimize the systems that draw power, thereby increasing endurance. Motors, onboard processing, and communications transceivers are the largest three power consumers on MAVs today. While motors and embedded processing are important to optimize, the sheer number of available communications options may leave MAV designers unsure how to proceed. By building an MAV around its onboard communications system, designers increase reliability, endurance, and capability with little or no added cost. Care must be taken to ensure that the end result meets the power, aerodynamic, and electromagnetic requirements for the particular MAV and its particular mission.

2.3.11 UAV Mission and Path Planning

Section ►[UAV Mission and Path Planning](#) includes chapters ►[Cooperative Mission Planning for Multi-UAV Teams](#), ►[Multi-team Consensus Bundle Algorithm](#), ►[Cooperative Mission and Path Planning for a Team of UAVs](#), ►[Cooperative Task Assignment and Path Planning for Multiple UAVs](#), ►[On the Decentralized](#)

Cooperative Control of Multiple Autonomous Vehicles, ►Innovative Collaborative Task Allocation for UAVs, ►Control of Communication Networks for Teams of UAVs, ►Information-Theoretic Exploration of Unmanned Aerial Vehicle in Unknown Cluttered Environment, ►Implementing Dubins Airplane Paths on Fixed-Wing UAVs, and ►Health Monitoring of a Drone Formation Affected by a Corrupted Control System.

Chapter ►Cooperative Mission Planning for Multi-UAV Teams by Ponda, Johnson, Geramifard, and How provides an overview of three of the most common planning frameworks: integer programming, Markov decision processes, and game theory. It also considers various architectural decisions that must be addressed when implementing online planning systems for multi-agent teams, providing insights on when centralized, distributed, and decentralized architectures might be good choices for a given application and how to organize the communication and computation to achieve desired mission performance. Algorithms that can be utilized within the various architectures are identified and discussed, and future directions for research are suggested.

Chapter ►Multi-team Consensus Bundle Algorithm by Argyle, Beard, and Casbeer focuses on how the consensus-based bundle algorithm (CBBA) is incorporated into a hierarchical concept of operation, where agents are divided into teams and each team plans for its agents to service a set of tasks. This team planning is carried out in a distributed manner using the traditional CBBA. An “outer-loop” team CBBA strategy coordinates teams’ plans. The hierarchical structure of the team CBBA facilitates a manageable architecture for large numbers of unmanned agents through human-centered operations. This is because each team is managed by a human operator with the team CBBA aiding coordination between teams.

Chapter ►Cooperative Mission and Path Planning for a Team of UAVs by Oh, Shin, Kim, Tsourdos, and White addresses the cooperative mission and path-planning problem of multiple UAVs in the context of the vehicle-routing problem. Since the conventional vehicle routing algorithms approximate their path to straight lines to reduce computational load, the physical constraints imposed on the vehicle are not to be taken into account. In order to mitigate this issue, a framework is described allowing integrated mission and path planning for coordinating UAVs using the Dubins theory based on the differential geometry concepts which can consider non-straight path segments. The main advantage of this approach is that the number of design parameters can be significantly reduced while providing the shortest, safe, and feasible path, which leads to a fast design process and more lightweight algorithms. In order to validate the integrated framework, cooperative mission and path-planning algorithms for two missions are developed: (1) road-network search route-planning patrolling every road segment of interest efficiently based on the optimization and approximation algorithm using nearest insertion and auction negotiation and (2) communication-relay route planning between a ground control station and the friendly fleet satisfying the constraints on the UAV speed and the avoidance of nonflying zones.

Chapter ►Cooperative Task Assignment and Path Planning for Multiple UAVs by Moon, Shim, and Oh presents a hierarchical framework for path planning and task assignment for multiple UAVs in a dynamic environment. The discussed

path-planning algorithm is based on the visibility and shortest path principles in Euclidean space, with the A* algorithm adopted to find an admissible path in a “best-first” approach during the search process. The path planner is augmented with a potential field-based trajectory planner, which solves for a detouring trajectory around other agents or pop-up obstacles. Task assignment is achieved by a negotiation-based algorithm, which assigns a task with the lowest cost to each agent after comparing all task costs of all participating agents. These algorithms are implemented on MATLAB/Simulink, which can run with simulated vehicle models or actual UAVs through a communication network. In the simulations, the algorithms are validated to perform task assignment and path planning flawlessly. In actual flight tests, the proposed algorithms were tested with a number of fixed wing UAVs in a fully realistic situation under various reality factors such as communication loss or tracking errors. The flight test shows, even in the presence of such uncertainties and logistic factors, the algorithms were able to perform all of the given tasks without any collision with other agents or obstacles.

Chapter ► [On the Decentralized Cooperative Control of Multiple Autonomous Vehicles](#) by Hirsch and Schroeder is concerned with dynamically determining appropriate flight patterns for a set of autonomous UAVs in an urban environment, with multiple mission goals. The UAVs are tasked with searching the urban region for targets of interest and tracking those targets that have been detected. It is assumed that there are limited communication capabilities between the UAVs and that there exist possible line-of-sight constraints between the UAVs and the targets. Each UAV operates its own dynamic feedback loop, in a receding-horizon framework, incorporating local information as well as remote information to determine the task to perform and the optimal flight path of UAV over the planning horizon. This results in a decentralized and more realistic model of the real-world situation. As the coupled task assignment and flight route optimization formulation is NP-hard, a hybrid heuristic for continuous global optimization is developed to solve for the flight plan and tasking over the planning horizon. Experiments are considered as communication range between UAVs varies.

Chapter ► [Innovative Collaborative Task Allocation for UAVs](#) by Karaman, Koyuncu, and Inalhan is devoted to a large-scale distributed task/target assignment problem for a fleet of autonomous UAVs, deriving an algorithm that uses the delayed column generation approach on a non-convex supply-demand formulation for the problem. The algorithm exploits a computationally tractable distributed coordination structure, i.e., a market for tasks and targets created by the UAV fleet. The resulting structure is solved via a fleet-optimal dual simplex ascent in which each UAV updates its respective flight plan costs with a linear update of waypoint task values as evaluated by the market. Synchronized and asynchronous distributed implementations of this approximation algorithm are demonstrated in a thorough experimental study involving dynamically changing scenarios with random pop-up targets.

Chapter ► [Control of Communication Networks for Teams of UAVs](#) by Kopeikin, Ponda, and How studies the communication network that is a fundamental component of a multi-UAV system, as it enables exchanges in command and control messages and allows for remotely sensed mission data to be sent to processing

centers. Network control is critical for the system to function properly. Challenges in network communication control are presented, followed by a background discussion in wireless networking including system architecture, wireless channel performance, topology models, and information routing, as well as bounds and limitations in cooperative control and decision making because of network limitations. Several network control strategies for multi-UAV systems are discussed, including motion planning methods to control the topology and relay deployment techniques to extend the performance of the network.

Chapter ► [Information-Theoretic Exploration of Unmanned Aerial Vehicle in Unknown Cluttered Environment](#) by Yang, Gan, and Sukkarieh proposes an information-theoretic path-planning method for exploration mission in unknown environments. Instead of using traditional grid-based occupancy maps, the Gaussian process (GP) is used to build an environmental map. The GP map performs inference directly on the collected sensor datasets, which allows it to infer the probability of collision for any query point in continuous 3-D space, removing the need for maintaining a full discrete map. This GP map is fused with the rapidly exploring random tree (RRT) path planner to plan a safe path and acquire information about the unknown environments. Using mutual information as an information measure, the most informative path is chosen as the path for exploration. Simulation results show that GP map combined with RRT planner achieves exploration task successfully in unknown complex environment and present their potential implementation in UAV missions.

Chapter ► [Implementing Dubins Airplane Paths on Fixed-Wing UAVs](#) by Owen, Beard, and McLain presents a 3-D Dubins airplane model, deriving a complete architecture for implementing Dubins airplane paths on small fixed-wing UAVs. A vector-field method is then used to design a guidance law that causes the Dubins airplane model to follow straight-line and helical paths. Dubins airplane paths are more complicated than Dubins car paths because of the altitude component. Based on the difference between the altitude of the start and end configurations, Dubins airplane paths are classified as low-, medium-, or high-altitude gain. While for medium- and high-altitude gain there are many different Dubins airplane paths, it is proposed to select the path that maximizes the average altitude throughout the maneuver. The proposed architecture is implemented on a six degree-of-freedom MATLAB/Simulink simulation of an Aerosonde UAV, and results from this simulation demonstrate the effectiveness of the technique.

Chapter ► [Health Monitoring of a Drone Formation Affected by a Corrupted Control System](#) by Lechevin, Rabbath, and Maupin proposes a dynamic feature that is instrumental in achieving vulnerability assessment of a network of UAVs, whose control system is possibly affected by the diffusion of malware. The feature consists of the characterization of the transition from stability to instability with probability one. The stability of the networked UAVs can be indirectly affected by malicious attacks targeting the communication units or the control systems. The network is modeled as a discrete-time, jump, linear system whose state space variables represent the probabilities that each node receives a malware and is infected by it. The stability analysis is obtained by means of a stochastic Lyapunov function argument and yields a sufficient condition expressed as a linear matrix

inequality (LMI). This LMI involves the networked asset state space matrices and the probability that each UAV's control system is infected. An approximation to the sufficient condition is proposed so that convergence of the system trajectories could be monitored online.

2.3.12 UAV Autonomy

Section ▶[UAV Autonomy](#) includes chapters ▶[Integrated Hardware/Software Architectures to Enable UAVs for Autonomous Flight](#), ▶[Distributed Optimization of Autonomous UAVs with Event-Driven Communication](#), and ▶[UAV Guidance Algorithms via Partially Observable Markov Decision Processes](#).

Chapter ▶[Integrated Hardware/Software Architectures to Enable UAVs for Autonomous Flight](#) by Pippin describes hardware and software architecture components that enable autonomous flight operation, also presenting some recommended architectural approaches and best practices. This is deemed essential as UAVs become increasingly more autonomous and ubiquitous and system hardware and software designers need to consider system requirements that contribute to increased levels of autonomy. Autonomous UAV architectures need to be modular, open, and provide visibility for operators and testers, also embracing open and extensible architectures that can be configured and deployed for robust and safe autonomous operation. Future UAVS will operate with varying levels of human interaction and will require the ability to process sensor data onboard and to operate using high-level, autonomous behaviors. The chapter concludes with a case study of the hardware and software architecture for an autonomous UAV research platform.

Chapter ▶[Distributed Optimization of Autonomous UAVs with Event-Driven Communication](#) by Zhong and Cassandras considers problems where multiple autonomous UAVs cooperate to control their individual state so as to optimize a common objective while communicating with each other to exchange state information. Since communication costs can be significant, especially when the UAVs operate with limited energy, conditions are sought under which communication of state information among nodes can be restricted while still ensuring that the optimization process converges. An asynchronous, event-driven scheme is described which limits communication to instants when some state estimation error function at a node exceeds a threshold. This scheme guarantees convergence to an optimal even in the presence of communication delays as long as they are bounded. This approach is applied to a UAV network coverage control problem where the objective is to maximize the probability of detecting events occurring in a given region, and it is shown that it may significantly reduce communication costs, hence also prolonging the system's lifetime, without any performance degradation.

Chapter ▶[UAV Guidance Algorithms via Partially Observable Markov Decision Processes](#) by Ragi and Chong presents a path-planning algorithm to guide UAVs to track multiple ground targets based on the theory of *partially observable Markov decision processes* (POMDPs). This method shows how to exploit the generality and flexibility of the POMDP framework by incorporating a variety of features of

interest naturally into the framework, which is accomplished by plugging in the appropriate models. Specifically, this study shows how to incorporate the following features by appropriately formulating the POMDP action space, state-transition law, and objective function: (1) control UAVs with both forward acceleration and bank angle subject to constraints, (2) account for the effect of wind disturbance on UAVs, (3) avoid collisions between UAVs and obstacles and among UAVs, (4) track targets while evading threats, (5) track evasive targets, and (6) mitigate track swaps.

2.3.13 UAV Sense, Detect, and Avoid Systems

Section ►UAV Sense, Detect, and Avoid Systems includes chapters ►Development of a Regulatory Safety Baseline for UAS Sense and Avoid, ►Achieving Sense and Avoid for Unmanned Aircraft Systems: Assessing the Gaps for Science and Research, ►Automatic Traffic Alert and Collision Avoidance System (TCAS) Onboard UAS, ►Test Flights to Demonstrate Effectiveness of a Ground-Based Detect and Avoid Integration Concept, ►Scalable RADAR-Based Sense and Avoid System for Unmanned Aircraft, and ►Assessment of Detect and Avoid Solutions for Use of Unmanned Aircraft Systems in Nonsegregated Airspace.

Chapter ►Development of a Regulatory Safety Baseline for UAS Sense and Avoid by Oztekin and Wever presents a systems level approach to analyze the safety impact of introducing UAS into the NAS. Utilizing Safety Management Systems (SMS) principles and the existing regulatory structure, a methodology is outlined to determine a regulatory safety baseline for a specific area of interest regarding a new aviation technology, such as UAS Sense and Avoid (SAA). The presented methodology is then employed to determine a baseline set of hazards and causal factors for the UAS Sense and Avoid problem domain and associated regulatory risk controls.

Chapter ►Achieving Sense and Avoid for Unmanned Aircraft Systems: Assessing the Gaps for Science and Research by Davis and Cook focuses on the efforts by the Office of the Under Secretary of Defense (OUSD) for Acquisition, Technology, and Logistics (AT&L) to make progress in solving the SAA problem by the establishment of the SAA Science and Research Panel (SARP) in February 2011. The SARP consists of a panel of experts in technologies necessary to provide UAS with the ability to sense and avoid other aircraft. The panel's primary purpose is to promote partnerships between the U.S. DoD and the broader academic and scientific community on UAS NAS Integration science and research initiatives in order to identify open questions and challenges in research and science efforts that must be addressed to provide UAS with an effectual SAA capability. Further, this chapter describes the methods of identifying and closing the science and research gaps that have been identified to date.

Chapter ►Automatic Traffic Alert and Collision Avoidance System (TCAS) Onboard UAS by Meyer, Gottken, Vernaleken, and Scharer reports on the SAA capabilities of the Barracuda UAS demonstrator, on which Cassidian installed a co-operative collision avoidance system based on Traffic Alert and Collision Avoidance

System (TCAS). The demonstration is intended as a risk mitigation measure to support future UAS development efforts. The demonstration steps involve laboratory tests with real equipment in a closed-loop demonstration and the automatic execution of a collision avoidance maneuver in flight. The in-flight demonstration takes place in segregated air space involving the Barracuda demonstrator platform and a real intruder. The chapter outlines the setup of the cooperative collision avoidance system demonstrator, including the interface with the flight control system of the Barracuda and the control authority granted to the cooperative collision avoidance system demonstrator. It summarizes the Human-Machine-Interface (HMI) layout and situation awareness provided to the UAS operator and illustrates how the potential loss of data link connection is covered by the system design. The development process is detailed, including requirements definition, validation, and verification steps as well as hardware-in-the-loop demonstration in support of the in-flight demonstration. It illustrates the demonstration scenarios utilized in the flight demonstration and the results achieved. Conclusions are drawn regarding the potential of such cooperative collision avoidance systems based on TCAS, the operational implications, and the certification aspects to be solved.

Chapter ► [Test Flights to Demonstrate Effectiveness of a Ground-Based Detect and Avoid Integration Concept](#) by Udovic, de Jong, and Vielhauer presents and reports on the VUSIL program (*Validierung von UAS zur Integration in den Luftraum*) that has developed a concept for integrating UAS into uncontrolled nonsegregated airspace, which is also used by manned civil aviation. The surrounding traffic is indicated on a display in the pilot ground control station (GCS), based on radar data provided by the Air Navigation Service Provider (ANSP). In addition, the pilot is supported by means of a “detect and avoid” tool in detecting potential conflicts. This tool shows conflicts and possible avoidance maneuvers on a separate display. To validate the integration concept, two flight campaigns with two different UAS (helicopter and fixed-wing UAS) and several intruders were conducted. In the first flight campaign, the intruders for the real flying UAS were simulated. In the second flight campaign, the intruder was a manned helicopter. The concept has been validated by the results of both flight campaigns and by a safety analysis. The latter has addressed demonstrating the equivalent level of safety (ELOS) of “see and avoid” in manned aviation and “detect and avoid” as developed in the VUSIL program.

Chapter ► [Scalable RADAR-Based Sense and Avoid System for Unmanned Aircraft](#) by Moses, Rutherford, and Valavanis presents one possible solution for addressing the midair collision problem in addition to increasing the levels of autonomy of UAS beyond waypoint navigation to include preemptive sensor-based collision avoidance. The proposed solution goes beyond the current state of the art by demonstrating the feasibility and providing an example of a scalable, self-contained, RADAR-based, collision avoidance system. The RADAR hardware and collision avoidance algorithms are presented in detail. The technology described herein can be made suitable for use on a miniature (maximum take-off weight <10 kg) UAS platform. This is of paramount importance as the miniature UAS field has the lowest barriers to entry (acquisition and operating costs) and consequently represents the most rapidly increasing class of UAS.

Chapter ▶ [Assessment of Detect and Avoid Solutions for Use of Unmanned Aircraft Systems in Nonsegregated Airspace](#) by Verstraeten, Stuip, and Birgelen summarizes and assesses the most promising candidate solutions for detect and avoid (DAA) for use of UAS in nonsegregated airspace against a set of DAA requirements. A DAA solution is the combination of the sensor suite, the avoidance algorithms, and the method of operation. The functions of a DAA solution are collision avoidance and, in uncontrolled airspace, separation provision. The chapter focuses on the detection of conflicting traffic by the sensor suite. The sensor suite needs to be able to detect different classes of conflicting traffic in varying environments. Requirements for DAA solutions have to be set, and DAA systems must then be developed to meet these requirements. ATM requirements for use of UAS in nonsegregated airspace have been drafted by EUROCONTROL, including a subset of requirements that are directly or indirectly of influence on the sensor suite used to DAA other traffic. This subset is used as baseline for the establishment of DAA requirements and expanded by development of five additional generic requirements based on the main tasks of a DAA solution: (1) detection of other traffic; (2) tracking of other traffic and (3) assessing if there is a conflict; (4) if there is a conflict, determining which evasive maneuver is to be executed; and (5) executing the selected maneuver. Five candidate solutions are assessed against the requirements: a noncooperative solution, a cooperative solution, and three solutions mixing cooperative and noncooperative sensors. Overall it is concluded that it is a great challenge to develop a collision avoidance solution for UAS with a satisfactory level of safety. It is even more difficult to develop a DAA system that is capable of both collision avoidance and separation provision.

2.3.14 Networked UAVs and UAV Swarms

Section ▶ [Networked UAVs and UAV Swarms](#) includes chapters ▶ [UAV Swarms: Models and Effective Interfaces](#), ▶ [Decentralized Multi-UAV Coalition Formation with Limited Communication Ranges](#), ▶ [Coordinated Convoy Protection Among Teams of Unmanned Aerial Vehicles](#), and ▶ [UAV Routing and Coordination in Stochastic, Dynamic Environments](#).

Chapter ▶ [UAV Swarms: Models and Effective Interfaces](#) by Chapman and Mesbahi examines the modeling and design of effective control interfaces for human operators of UAV swarms. The swarm is modeled as a two-component hierarchical system consisting of the interaction dynamics among the UAVs in the swarm, referred to as the *network dynamics*, and the *UAV dynamics* itself. Human operators are assumed to be able to interface with the swarm via the network dynamics, which in turn has adopted a *leader-follower consensus model*. The system-theoretic and topological features of the network dynamics are then examined in order to design effective mechanisms for interfacing with the swarm. A performance metric is selected for reasoning about effective human swarm interaction. The role of topological features of the network is highlighted in the context of the chosen metric and is related through the effective resistance of the corresponding electrical

network. This is then followed by exploiting such topological features for designing a network rewiring protocol to maximize the metric. These topology design tools are applied to wind-gust rejection in disturbed swarming scenarios, demonstrating the viability of topology-assisted design for improved swarm performance. A network-based model reduction is also proposed to form a lower-order model of the network which is easier for the human operators to conceptualize and manage. The reduction process involves a novel partitioning scheme, dubbed *leader partition*, in order to fuse “similar” states in the UAV network and to form a graph-theoretic method for model reduction. This model reduction technique is then applied to derive improved swarming performance in the presence of wind gusts.

Chapter ► [Decentralized Multi-UAV Coalition Formation with Limited Communication Ranges](#) by Sujit, Manathara, Ghose, and Sousa discusses a communication protocol derived to search for potential coalition members over a dynamic UAV network and a strategy for a UAV coalition to track and destroy moving targets. The effects of communication ranges and delays as well as the scalability of the proposed schemes with change in number of targets and UAVs are studied. The results show that it is beneficial to search for potential coalition members over a wide diameter in the UAV network when the communication delay is less, while it is more advantageous to determine a coalition from among the immediate neighbors in the presence of significant communication delay.

Chapter ► [Coordinated Convoy Protection Among Teams of Unmanned Aerial Vehicles](#) by Egerstedt, Rahmani, and Young investigates how to control and coordinate teams of UAVs to provide protection to convoys of ground vehicles. This is a rich yet canonical problem when coordinating multiple UAVs in that coordinated movements, task assignments, and resource balancing must all be performed for a successful completion of the mission. Time optimal paths for providing convoy protection to stationary ground vehicles are presented, and these algorithms are extended to moving ground vehicles. The assignment problems, associated with dispatching UAVs from the convoy to inspect and clear potential threats, are also discussed.

Chapter ► [UAV Routing and Coordination in Stochastic, Dynamic Environments](#) by Enright, Frazzoli, Pavone, and Savla focuses on a technical approach that relies upon methods from queuing theory, combinatorial optimization, and stochastic geometry. The main advantage of this approach is its ability to provide analytical estimates of the UAV system performance on a given problem, thus, providing insight into how performance is affected by design and environmental parameters, such as the number of UAVs and the target distribution. This approach also provides provable guarantees on the system’s performance with respect to an ideal optimum. To illustrate this approach, a variety of scenarios are considered, ranging from the simplest case where one UAV moves along continuous paths and has unlimited sensing capabilities, to the case where the motion of the UAV is subject to curvature constraints, and finally to the case where the UAV has a finite sensor footprint. Finally, the problem of cooperative routing algorithms for multiple UAVs is considered, within the same queuing-theoretical framework, and with a focus on control decentralization.

2.3.15 UAV Integration into the National Airspace

Section ►UAV Integration into the National Airspace includes chapters ►Aviation Regulation, ►Human Factors of Unmanned Aircraft System Integration in the National Airspace System, ►Methodologies for Regulatory Compliance and Harmonization, ►A Certification Strategy for Small Unmanned Aircraft Performing Nomadic Missions in the US National Airspace System, ►Hazard and Safety Risk Modeling, ►Safety Risk Management of Unmanned Aircraft Systems, ►Certification of Small UAS, ►Technology Surveys and Regulatory Gap Analyses of UAS Subsystems Toward Access to the NAS, ►Concept of Operations of Small Unmanned Aerial Systems: Basis for Airworthiness Toward Personal Remote Sensing, and ►Standards and Certification of a UAS Sense and Avoid Capability.

Chapter ►Aviation Regulation by Dalamagkidis presents a brief introduction to aviation regulation. Key terms are defined, and an overview of current manned aviation regulations is provided using the U.S. Federal Aviation Regulation as an example. This includes airworthiness certification, operation rules, and airspace classes.

Chapter ►Human Factors of Unmanned Aircraft System Integration in the National Airspace System by Kaliardos and Lyall focuses on identifying human factor challenges to integrating UAS in the National Airspace System (NAS) for both pilots and air traffic controllers. The method for identifying these challenges is primarily based on the differences or “gaps” between manned aircraft in the NAS today and the unique aspects of UAS. The goal is not to generate a comprehensive list of human factors issues, but to focus on those that are traceable to fundamental characteristics of UAS and that are also considered challenging with respect to the current NAS and its regulatory framework. It is assumed that for integration into the NAS, future UAS will be much more standardized in many respects than in the past – closer to that of today’s civil manned aircraft, which are subject to rigorous design, operational, manufacturing, and training approvals by the FAA. Under such assumptions, human factors not only are central to the very definition of UAS but are also central to some of the most important challenges facing the integration of UAS in the NAS.

Chapter ►Methodologies for Regulatory Compliance and Harmonization by Marshall proposes a methodology for analysis of current regulations for applicability to UAS activities. Where the rules are in development, or merely being contemplated, a structure for harmonization with international standards is proposed. The focus is on the USA as the nation that has the most comprehensive aviation regulations, while also supporting major efforts to promulgate regulations and standards to supplement those regulations with UAS specific requirements and criteria. To provide global context, the International Civil Aviation Organization’s (ICAO’s) structure and procedures are reviewed in detail, and contrasted with State sponsored regulatory processes, with an analysis of the applicability of ICAO’s standards to UAS operations.

Chapter ►A Certification Strategy for Small Unmanned Aircraft Performing Nomadic Missions in the US National Airspace System by Stachura, Elston, Argrow, Frew, and Dixon discusses the specifics of the Certificates of Authorization (COA)

obtained for the second Verification of the Origin of Rotation in Tornadoes Experiment (VORTEX2) project and how the operations are conducted to satisfy the COA requirements. A strategy is outlined for operating these nomadic missions with small UAS within the confines of FAA regulations. This includes information on getting FAA COAs for a large area, specifically focusing on area selection, airworthiness, and emergency procedures, which are the keys to these applications.

Chapter ►[Hazard and Safety Risk Modeling](#) by Dalamagkidis presents aspects of risk modeling with a focus on UAS. It provides an overview of the current level of safety of manned aviation in terms of accident statistics. These are then mapped as target levels for UAS under the “Equivalent Level of Safety” principle to provide a glimpse at what that may entail for UAS regulations. Different methodologies are presented for estimating the risk of ground impact and midair collision accidents and how these estimates can be translated to system requirements. Guidelines are provided on the use of different risk models followed by applying a selection of them to five different UAS in two distinct scenarios, to compare the results of different choices.

Chapter ►[Safety Risk Management of Unmanned Aircraft Systems](#) by Clothier and Walker provides existing risk practitioners with a high-level introduction to some of the unique issues and challenges in the application of the safety risk management process to UAS. The scope is limited to safety risks associated with the operation of unmanned aircraft in the civil airspace system and over inhabited areas. This chapter notes the unique aspects associated with the application of the safety risk management process to UAS compared to that of conventionally piloted aircraft. Key challenges discussed include the specification of high-level safety criteria; the identification, analysis, and evaluation of the risks; and the effectiveness of available technical and operational mitigation strategies. Some solutions to these challenges are examined including those currently in practice and those still under research and development.

Chapter ►[Certification of Small UAS](#) by Leijgraaf describes certification procedures for (small) UAS. It focuses on the certification process, the requirements for the safe design of a UAS, and the organizational requirements for the company designing the UAS.

Chapter ►[Technology Surveys and Regulatory Gap Analyses of UAS Subsystems Toward Access to the NAS](#) by Stansbury and Wilson provides the necessary details on how to conduct a technology survey and regulatory gap analysis of UAS technology subsystems. Four past studies performed by Embry-Riddle Aeronautical University for the FAA’s William J. Hughes Technology Center are discussed. These studies address UAS propulsion systems, sense and avoid technologies and procedures, command control and communication, and emergency recovery and flight termination systems. A recommended process for future studies is provided.

Chapter ►[Concept of Operations of Small Unmanned Aerial Systems: Basis for Airworthiness Toward Personal Remote Sensing](#) by Stark, Coopmans, and Chen discusses the challenges of UAS integration into the NAS for a class of small UAS for personal remote sensing (PRS). This approach is centered on the three pillared foundations for NAS integration presented by the U.S. DoD Unmanned Aircraft System Airspace Integration Plan and focuses on the specific challenges that are

unique to the PRS UAS platform. This chapter presents a concept of operations for these unmanned aircrafts along with an application scenario example and concludes with a discussion of the future use of UAS in the NAS and how it is achievable through adequate regulations and standards.

Chapter ►[Standards and Certification of a UAS Sense and Avoid Capability](#) by Zeitlin centers on “certifiable” and “standardized” UAV sense and avoid (SAA) capabilities that are needed to mitigate for the lack of a remote pilot’s ability to “see and avoid” other aircraft, which is an operational requirement. Regulators certify the airworthiness of aircraft, assuring their compliance with applicable regulations. Approvals of SAA implementations constitute part of that process. Each applicant needs to develop a Project Specific Certification Plan in close coordination with the regulator. Various tools for safety analysis and configuration control support this effort. Specification of the certification basis for SAA needs close attention, since the system addresses functions traditionally allocated to an onboard pilot. Algorithms and software will receive scrutiny due to their safety roles. Standards are developed to capture common requirements in support of systems that may use some degree of unique design innovation. A standard should suggest means of demonstrating compliance to its requirements, and so doing will greatly simplify the burden for both applicant and regulator. The closest existing standard to SAA is that for the Traffic Alert and Collision Avoidance System II (TCAS II), but SAA includes the additional function of self-separation. SAA can be implemented with various technologies in a variety of architectures. SAA should provide two basic functions, each performing prescribed subfunctions. Considerable modeling and simulation will be required to validate system performance and support the safety case. The regulator may offer guidance to applicants through publication of an Advisory Circular.

2.3.16 UAV-Human Interfaces and Decision Support Systems

Section ►[UAV-Human Interfaces and Decision Support Systems](#) includes chapters ►[Human Interfaces in Micro and Small Unmanned Aerial Systems](#), ►[Human Factors Perspective on Next Generation Unmanned Aerial Systems](#), ►[Cognitive Task Analysis for UAV Design](#), ►[Display and Control Concepts for Multi-UAV Applications](#), and ►[UAV Operators Workload Assessment](#).

Chapter ►[Human Interfaces in Micro and Small Unmanned Aerial Systems](#) by Peschel and Murphy surveys the various aspects of user interfaces for a Mission Specialist role on micro and small UAS human-robot teams. Three formal human team member roles are presented, this of Flight Operations Director, Pilot and Navigator, and the Mission Specialist, in order to identify distinct team member role differences. For each category of UAV, the Mission Specialist role is evaluated to determine what type of user interface technologies are present and/or available, how the Mission Specialist currently or could interact with the user interface technology, and what are the perceived consequences of this user interface technology in the context of the three identified micro and small UAS human-robot team roles. The findings of this survey suggest that current Mission Specialist performance in micro

and small UAS may be suboptimal due to the sharing of a single Pilot and Navigator-oriented user interface or a reuse of the Pilot and Navigator user interface.

Chapter ► [Human Factors Perspective on Next Generation Unmanned Aerial Systems](#) by Goodrich and Cummings presents a comprehensive view on UAS autonomy and user interface design, motivating emerging themes in near-term and far-term UAS developments. Issues include user interface design, UAS autonomy, UAS teaming, operator workload, and payload management.

Chapter ► [Cognitive Task Analysis for UAV Design](#) by Adams provides an overview of cognitive analysis methods that can be integrated into the early system design process to account for the human element. It specifically provides an overview of task analysis, work analysis, and information flow analysis methods along with an overview of how individual methods from the different analysis classes have been combined. It is argued, and rightly so, that although UAS are complex tools requiring thoughtful designs that account for the platform itself, payloads, and human users as a holistic system, often the focus is first placed on designing the platform or payload, while the human element is considered at a later stage. As such, system design approaches do not properly consider the human cognitive limitations and do not design the system to support the human's tasks and decision making. Specific examples of applying these analyses methods to unmanned aerial systems, ranging from potential futuristic systems to research systems to deployed systems, are also reviewed.

Chapter ► [Display and Control Concepts for Multi-UAV Applications](#) by Calhoun and Draper introduces a variety of display and control concepts that may be especially applicable to multi-UAV applications. Many such technologies are being investigated at the U.S. Air Force Research Laboratory and include multimodal interfaces, temporal displays, and integrated visualization concepts. The thesis is that for any candidate technology, rigorous user-centered design and evaluation needs to be conducted, which focuses on the tasks and requirements of the specific multi-UAV application being targeted. It is also important to ensure that the operator has adequate means to first observe and then direct the automation's functioning in order to be responsive to changes in the mission, vehicle status, or operational environment. What is important to capture is that advances in automation technology are leading to the development of UAV concepts in which a single pilot will be responsible for multiple UAVs. Thus, the operator's role will become more supervisory in nature, resulting in a new interface paradigm required for effective supervisory control of multiple UAVs by a single operator.

Chapter ► [UAV Operators Workload Assessment by Optical Brain Imaging Technology \(fNIR\)](#) by Izzetoglu, Ayaz, Hing, Shewokis, Bunce, Oh, and Onaral discusses the utilization of near-infrared-based functional brain imaging systems, fNIRs, in the monitoring of a cognitive workload during UAV operation and as an objective measure of expertise development, that is, the transition from novice to expert during operator training. The challenge relates to the fact that UAV ground operators are required to acquire skills quickly and completely, with a level of expertise that builds the operator's confidence in his/her ability to control the UAV under adverse conditions. As UAVs are held to increasingly higher standards of

efficiency and safety, operators are routinely required to perform more information-dense and cognitively demanding tasks, resulting in increased cognitive workloads during operation. Functional brain monitoring offers the potential to help UAV operators meet these challenges, and research has demonstrated the utility of fNIRs for the purpose of monitoring frontal cortical areas that support executive functions (attention, working memory, response monitoring). fNIR technology allows for continuous monitoring of operators during training as they develop expertise, as well as the capacity to monitor their cognitive workload under operational conditions while controlling UAVs in critical missions.

2.3.17 Human Factors and Training

Section ►[Human Factors and Training](#) includes chapters ►[Using Best Practices as a Way Forward for Remotely Piloted Aircraft Operators: Integrated Combat Operations Training-Research Testbed](#) and ►[From Research to Operations: The PITVANT UAS Training Experience](#).

Chapter ►[Using Best Practices as a Way Forward for Remotely Piloted Aircraft Operators: Integrated Combat Operations Training-Research Testbed](#) by Rowe, Conwell, Morris, and Schill centers on work undertaken at the U.S. Air Force Research Laboratory (AFRL) in the Integrated Combat Operations Training-Research Testbed (ICOTT), where training gaps in the RPA community are extensively researched. This is the aftermath of the massive growth and increase of medium-altitude, long-endurance (MALE) remotely piloted aircraft combat air patrols, which has enforced the need to train RPA operators quickly and effectively. Using a competency-based approach, researchers, engineers, subject matter experts, and warfighters are developing a training research testbed to evaluate curriculum, training methodologies, and best practices in training simulation. The chapter addresses the evolution of training requirements for the RPA community as well as targeted solutions to support RPA training, operational readiness needs, and an exploration into the existing methods and common tools used to understand and evaluate the relative trade-off in training utility associated with alternative training systems. The overarching discussion focuses on methods for providing more rapid and efficient knowledge and skill development for RPA operators.

Chapter ►[From Research to Operations: The PITVANT UAS Training Experience](#) by Madruga Matos, Vieira Caetano, Morgado, and Borges de Sousa presents the PITVANT UAS R&D program training experience, including the different crew positions and responsibilities, the several training phases, and the systems engineering methodology behind all developments carried by the joint team of the Portuguese Air Force Academy and Porto University. The training program is described including theoretical classes, written exams, and practical instruction and evaluation. The process of developing, testing, and validating concepts of operations and the correspondent operational procedures and their training with the involvement of end users is reported. The current operational capabilities of the systems are explained, showing that the PITVANT program has achieved a level of

maturity that enables testing and validation in near real scenarios. This leads to the next step in the systems spiral development that should include, in the near future, the possibility of technology transfer to the industry.

2.3.18 UAV Logistics Support

Section ▶UAV Logistics Support includes chapters ▶UAV Logistic Support Definition and ▶UAV Logistics for Life-Cycle Management.

Chapter ▶UAV Logistic Support Definition by Chiesa and Fioriti presents considerations and hypotheses about RAMS characteristics and logistic support options for operational fleets of UAS. At first, an outline of why logistic support is relevant for UAS in comparison with other complex systems is presented, followed by considerations why reliability, safety, and maintainability are relevant aspects for UAS, explaining how such characteristics can be achieved by design. Synthetic considerations regarding transportability, a characteristic strictly connected to safety, reliability, maintainability, and logistic operations, are discussed. It is stated that considering (in addition to technical features) targets, requirements, constraints at the origin of a fleet of UAS, fleet sizing, and related system support (elements and strategies), a useful tool to face such complex problems is stochastic simulation and in particular a Monte Carlo methodology, which is also discussed.

Chapter ▶UAV Logistics for Life-Cycle Management by Karaagac, Pakfiliz, Quagliotti, and Alemdaroglu provides the reader with general knowledge on UAV Logistics for Life-Cycle Management, focusing on current logistics challenges and possible future UAV logistics trends that may improve UAV deployment. Although UAVs do not have a pilot onboard, they require nearly the same logistics support as most manned aircraft. That is, logistics support covers support for all components including the unmanned aircraft and the ground control station. The objective is to design a system with a reduced logistic footprint, enabling fast deployment and high mobility.

2.3.19 UAV Applications

Section ▶UAV Applications includes chapters ▶Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring, ▶Measurement and Exploration in Volcanic Environments, ▶Cooperative Unmanned Aerial Systems for Fire Detection, Monitoring, and Extinguishing, ▶Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States, ▶Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation, ▶Unmanned Aircraft Systems for Maritime Operations, ▶Autonomous Remote Sensing of Invasive Species from Robotic Aircraft, and ▶Cyber-Physical Systems Enabled by Small Unmanned Aerial Vehicles.

Chapter ▶Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring by Kanistras, Martins, Rutherford, and Valavanis focuses on surveying UAV-based

systems for traffic monitoring and management. Although there is voluminous research on the subject, UAVs are proven to be a viable and less time consuming alternative to real-time traffic monitoring and management as they provide the dynamic “eye-in-the-sky” solution to the problem.

Chapter ► [Measurement and Exploration in Volcanic Environments](#) by Melita, Longo, Muscato, and Giudice presents a challenging application for UAVs in volcanic environments, as they may be used to collect typical aerial measurements and data. The main forms of measurements concern the collection of visual/thermal images and gas analysis and sampling. An overview of the major worldwide projects carried out in the last few years by several research organizations is summarized and discussed.

Chapter ► [Cooperative Unmanned Aerial Systems for Fire Detection, Monitoring, and Extinguishing](#) by Merino, Martinez-de Dios and Ollero deals with the application of cooperative UAVs to forest fires. It presents a decision and control architecture for multi-UAS teams that are equipped with diverse sensors (e.g., infrared and visual cameras) and details the perception techniques (fire segmentation, geo-localization, and data fusion) that are used toward fire extinguishing.

Chapter ► [Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States](#) by Ambrosia and Zajkowski describes some of the UAS lessons learned by both the U.S. Forest Service (USFS) and the National Aeronautics and Space Administration (NASA) during their more than 10-year efforts on identifying appropriate UAS wildfire observation capabilities. The agencies, along with the UAS community, have demonstrated the use of both low-altitude, short-endurance (LASE) and medium-altitude, long-endurance (MALE) platforms for wildfire sensor intelligence gathering. Those experiences are identified along with an examination of the future direction of UAS for supporting the wildfire management community.

Chapter ► [Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation](#) by Kondak, Ollero, Maza, Krieger, Albu-Schaeffer, Schwarzbach, and Laiacker discusses the novel and very challenging applications of using UAVs for aerial manipulation and load transportation. A specific application of using autonomous helicopters equipped with robotic arms for aerial manipulation is detailed, along with field experiments demonstrating joint load transportation with multiple autonomous helicopters.

Chapter ► [Unmanned Aircraft Systems for Maritime Operations](#) by Borges de Sousa, McGuillivary, Bento, Morgado, Matos, Gomes Bencatel, and de Oliveira discusses current practices and major UAS trends for maritime operations, along with projections of future UAS maritime applications. Various aspects of UAS program management, including organization, operator responsibilities, program operation, and overall program costs are reviewed, followed by maritime aircraft missions and required capabilities. Technological trends are highlighted focusing on developments relevant to maritime UAS operations. Trends including miniaturization of sensors and computer systems, high energy density of power sources, and increased subsystem standardization and modularity will have important effects in the future, including increased system autonomy via new command and control frameworks.

Chapter ▶ [Autonomous Remote Sensing of Invasive Species from Robotic Aircraft](#) by Goktogan and Sukkarieh presents an ensemble system consisting of a helicopter, a fixed-wing UAV, and their supporting infrastructure that is used in a number of environmental research experiments focused on autonomous remote sensing, detection, classification, and management of invasive species, weeds, in Australia. In these experiments, three distinct families of weeds on three different terrains are investigated. In the first group of experiments, a helicopter UAV equipped with a high-resolution imaging payload is flown over difficult to reach water channels and wetlands for detection of aquatic weeds. The second set of experiments is performed in relatively flat rangelands to map woody infestations. The third set of experiments is focused on the airborne detection of wheel cacti on remote mountainous terrain using fixed-wing aircrafts. Successful results of these experiments suggest that robotic aircrafts in a properly designed UAS can play an important role in environmental robotic science.

Chapter ▶ [Cyber-Physical Systems Enabled by Small Unmanned Aerial Vehicles](#) by Coopmans, Stark, Jensen, Chen, and McKee provides several examples of unmanned aircraft sensing-enabled cyber-physical system (CPS) scenarios, enabling adaptive management and effective control of complex physical systems such as water distribution based on measurement of soil moisture and crop evapotranspiration, radio tag-based tracking of fish, alternative energy harvesting, mapping of invasive plant species, and airborne plume (pollution) tracking. This chapter serves as a motivator for potential widespread UAS applications where UAS are used as mobile sensors and/or mobile actuators in large-scale, closed-loop CPS.

2.3.20 Social and Ethical Implications

Section ▶ [Social and Ethical Implications](#) includes chapters ▶ [Ethics and UAVs](#), ▶ [International Governance of Autonomous Military Robots](#), ▶ [The Moral Case Against Autonomous and Semiautonomous UAVs](#), ▶ [The Moral Case for Autonomy in Unmanned Systems](#), ▶ [Moral Predators: The Duty to Employ Uninhabited Aerial Vehicles](#), ▶ [Killer Robots: Ethical Issues in the Design of Unmanned Systems for Military Applications](#), ▶ [Conclusion: Testimony on Ethical Issues of UAVs for National Academy](#), and ▶ [Future of Unmanned Aviation](#).

Chapter ▶ [Ethics and UAVs](#) by Lucas provides an overview of ethical, legal, and “robot morality” challenges arising from the military and security uses of robotics and other unmanned vehicles in a number of relevant conflict environments, including the increased domestic uses of both remotely controlled and fully autonomous unmanned aerial, maritime, and ground systems for immigration control, border surveillance, drug interdiction, and domestic law enforcement.

Chapter ▶ [International Governance of Autonomous Military Robots](#) by Marchant, Allenby, Arkin, Borenstein, Gaudet, Kittrie, Lin, Lucas, O’Meara, and Silberman aims at providing a background of some of the principal legal issues that may motivate and initiate a much-needed legal and ethical dialogue related to the use of lethal autonomous robotic technologies in the military context.

Chapter ► [The Moral Case Against Autonomous and Semiautonomous UAVs](#) by Sharkey focuses on ethical concerns about the application of armed robots in areas with mixed combatant and civilian populations, making the claim that one must scrutinize the mapping between the applications of the new technology and the current laws to ensure that those laws are preserved and followed.

Chapter ► [The Moral Case for Autonomy in Unmanned Systems](#) by Arkin centers on the need to restrain the technology itself above and beyond the human limits of the warfighters themselves, making the case for the use of ethical autonomy in unmanned systems, which is deemed necessary since warfare will continue and autonomous robots will ultimately be deployed in its conduct. In this way, one may hope that battlefield atrocities will not become progressively worse given the progression of standoff weapons and increasing use of technology.

Chapter ► [Moral Predators: The Duty to Employ Uninhabited Aerial Vehicles](#) by Strawser focuses on the supposed objections to the use of UAVs in principle, defending the view that society has a duty to protect an agent engaged in a justified act from harm to the greatest extent possible so long as that protection does not interfere with the agent's ability to act justly. Since UAVs do afford such protection an argument is made that a nation is obligated to employ UAV weapon systems if it can be shown that their use does not significantly reduce a warfighter's operational capability.

Chapter ► [Killer Robots: Ethical Issues in the Design of Unmanned Systems for Military Applications](#) by Sparrow argues that designers of unmanned systems used for military applications must consider ethical, operational, requirements and limits of such systems. This chapter identifies and discusses a number of issues and offers some analysis of the implications of each issue and how it might be addressed.

The concluding chapter in this section ► [Conclusion: Testimony on Ethical Issues of UAVs for National Academy of Sciences](#) contains a transcript of testimony presented by Lucas to a select ad hoc committee convened by the National Academy of Sciences (USA) in response to a request from the U.S. Defense Advanced Research Projects Agency (DARPA) to evaluate the "Ethical and Societal Implications of Advances in Militarily Significant Technologies That Are Rapidly Changing and Increasingly Globally Accessible." This testimony was delivered at the inaugural meeting of the committee on 31 August 2011 at the NAS Beckman Center, on the campus of the University of California – Irvine.

Chapter ► [Future of Unmanned Aviation](#), brings closure to the handbook.

2.4 Remarks

The rest of this Introduction section includes chapters related to ► [Aviation History and Unmanned Flight](#), ► [Definitions and Terminology](#), ► [Classification of UAVs](#), and ► [Military and Civilian Unmanned Aircraft](#). After going through the Introduction section, the reader will be ready to proceed to technical aspects of unmanned aviation and aircraft.

Konstantinos Dalamagkidis

Contents

3.1 Introduction 43

3.2 UAV or UAS?..... 44

3.3 Important Terms 45

3.4 List of Acronyms 51

References 55

Abstract

The field of UAVs has been plagued with different terms to describe the same thing. Even the term UAV or unmanned aerial vehicle is controversial, and it has been replaced in many places with UAS or Unmanned Aircraft System. The different terms often came from the different requirements and concepts between military and civilian systems or have regulatory/legal importance. This chapter provides definitions for some of the most used terms in the field.

3.1 Introduction

Since many names have been to describe unmanned aircraft, this chapter begins with an overview of these terms and their *raison d’être*. It then continues with some other commonly used terms and expressions, concluding with a list of acronyms that are in use in the field and throughout this handbook.

3.2 UAV or UAS?

Several names have already been used to describe unmanned aircraft. UAVs became UAS, the preferred term used by the Federal Aviation Administration (FAA). Other names included Remotely Piloted Vehicles (RPVs), a term that was used in the Vietnam War. Today the USAF has mainly substituted RPV for Remotely Piloted Aircraft or RPA, a term used to include both the aircraft and the pilot, while the United Kingdom has designated them as Remotely Piloted Air System (RPAS), to demonstrate the presence of the man in the loop to control them; see Herlik (2010) and Fishpool (2010), respectively.

An unmanned aerial vehicle (also known as a drone) refers to a pilotless aircraft, a flying machine without an onboard human pilot or passengers. As such, “unmanned” implies total absence of a human who directs and actively pilots the aircraft. Control functions for unmanned aircraft may be either onboard or off-board (remote control).

The term UAV or unmanned aerial vehicle has been used for several years to describe unmanned aerial systems. Various definitions have been proposed for this term, like

A reusable aircraft designed to operate without an onboard pilot. It does not carry passengers and can be either remotely piloted or preprogrammed to fly autonomously.
Joint Capability Group on Unmanned Aerial Vehicles (2007)

In the definition above, the characterization *reusable* is used to differentiate unmanned aircraft from guided weapons and other munition delivery systems.

A few years ago, the U.S. Department of Defense (DoD), followed by the FAA and the European Aviation Safety Agency (EASA), adopted the term UAS or Unmanned Aircraft System. This was meant to signify that UAS are aircraft and as such airworthiness will need to be demonstrated, and they are also systems consisting of ground control stations, communication links, and launch and retrieval systems in addition to the aircraft itself.

The FAA has defined an *Unmanned Aircraft* or UA as

A device used or intended to be used for flight in the air that has no onboard pilot. This includes all classes of airplanes, helicopters, airships, and translational lift aircraft that have no onboard pilot. Unmanned aircraft are understood to include only those aircraft controllable in three axes and therefore, exclude traditional balloons.

Federal Aviation Administration (2008)

As a comparison, the definition of *Unmanned Vehicle* given in the 2007–2012 Unmanned systems roadmap is also provided:

A powered vehicle that does not carry a human operator, can be operated autonomously or remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload. Ballistic or semi-ballistic vehicles, cruise missiles, artillery projectiles, torpedoes, mines, satellites, and unattended sensors (with no form of propulsion) are not considered unmanned vehicles. Unmanned vehicles are the primary component of unmanned systems.

U.S. Department of Defense. Office of the Secretary of Defense (2007)

Similarly, EASA defines UAS as

An Unmanned Aircraft System (UAS) comprises individual system elements consisting of an “unmanned aircraft”, the “control station” and any other system elements necessary to enable flight, i.e. “command and control link” and “launch and recovery elements”. There may be multiple control stations, command & control links and launch and recovery elements within a UAS.

European Aviation Safety Agency (2009)

In practice, UAS and UAV are often used interchangeably, and only when the system aspect is important (mainly for legal/regulatory reasons) does UAS have preference.

3.3 Important Terms

Accident An unplanned event or series of events that results in damages. The term mishap is often used to refer to an accident, as well. Accidents that do not occur directly, but rather as a result of other accidents, are referred to as secondary accidents. Note that the National Transportation Safety Board (NTSB) defines an accident as an occurrence incidental to flight in which, as a result of the operation of an aircraft, any person (occupant or non-occupant) receives fatal or serious injury or any aircraft receives substantial damage.

Federal Aviation Administration (2000)

Accident Probability Levels An arbitrary categorization that provides a qualitative measure of the most reasonable likelihood of occurrence of an accident resulting from personnel error, environmental conditions, design inadequacies, procedural deficiencies, or system, subsystem, or component failure or malfunction.

U.S. Department of Defense (2000)

Accident Risk Categories An arbitrary categorization of accident risk assessment values often used to generate specific actions such as mandatory reporting of certain hazards to management for action or formal acceptance of the associated accident risk.

U.S. Department of Defense (2000)

Accident Severity An assessment of the consequences of the most reasonable credible mishap that could be caused by a specific hazard.

U.S. Department of Defense (2000)

Air Traffic Control (ATC) ATC is a service provided under appropriate authority to promote the safe, orderly, and expeditious flow of air traffic.

Air Transport Association (ATA) (2008)

Air Traffic Management (ATM) ATM refers to the dynamic, integrated management of air traffic and airspace – safely, economically, and efficiently – through

the provision of facilities and seamless services in collaboration with all involved parties.

Air Transport Association (ATA) (2008)

Air Vehicle There are certain types of aircraft like moored balloons, unmanned balloons, unmanned rockets, and ultralights that are considered “vehicles” and, thus, are allowed to fly without an airworthiness certificate. More specifically most requirements regarding pilot certification, operating and flight rules, vehicle registration and marking, and maintenance certification that are normally applicable to aircraft do not apply for this category Schultz (2006), although operational restrictions are in place. For example, the following pertain to the operation of ultralight vehicles (FAR Part 103): (i) single occupant, (ii) daylight operations, (iii) recreation or sport purposes only, and (iv) no flight over congested areas in cities, towns, or open areas when crowds are present.

Aircraft An aircraft is defined in the United States Code (USC) as *any contrivance invented, used, or designed to navigate, or fly in, the air* (49 USC §40102). In general, aircraft exist in a variety of possible configurations: fixed-wing, rotary-wing or rotorcraft, helicopters, vertical takeoff and landing (VTOL) vehicles, or short takeoff and landing (STOL) ones. An aircraft may also be either heavier or lighter than air, with balloons and airships belonging to the latter category.

Airman Airman as defined in (49 USC 40102) is an individual:

1. In command, or as pilot, mechanic, or member of the crew, who navigates aircraft when under way
2. Except to the extent the Administrator of the Federal Aviation Administration may provide otherwise for individuals employed outside the U.S., who is directly in charge of inspecting, maintaining, overhauling, or repairing aircraft, aircraft engines, propellers, or appliances
3. Who serves as an aircraft dispatcher or air traffic control-tower operator

Airport The airport is defined as a landing area used regularly by aircraft for receiving or discharging passengers or cargo (49 USC 40102).

Background Risk Risks voluntarily accepted in the course of normal activities.
Range Safety Group, Range Commanders Council (2007)

Collision Avoidance Considered a last resort maneuver of an aircraft to avoid an imminent collision. Without the maneuver a collision might occur.

Small Unmanned Aircraft System Aviation Rulemaking Committee (2009)

Compounding Conservatism An analysis approach that results in extremely conservative results by making a series of conservative assumptions.

Range Safety Group, Range Commanders Council (2007)

Conflict Avoidance Activity which seeks to ensure that aircraft remain safely separated and well clear of each other as to not present a collision hazard.

Small Unmanned Aircraft System Aviation Rulemaking Committee (2009)

Conservatism As used in risk analysis modeling, conservatism is a set of modeling assumptions that exaggerates the risk by overstating event probabilities, hazard probabilities, or consequences.

Range Safety Group, Range Commanders Council (2007)

Damage An undesired outcome that may include injury, fatality, as well as physical, functional, and/or monetary loss.

Expected Fatalities The mean number of fatalities predicted to occur as a result of an operation if the operation were to be repeated many times.

Range Safety Group, Range Commanders Council (2007)

Exposure The number of personnel or resources affected by a given event or, over time, by repeated events. This can be expressed in terms of time, proximity, volume, or repetition. This parameter may be included in the estimation of severity or probability, or considered separately.

United States Air Force (1998)

Fail-Safe A design feature that ensures the system remains safe, or in the event of a failure, causes the system to revert to a state that will not cause a mishap.

U.S. Department of Defense (2000)

Fixed-Wing Aircraft Aircraft with immovable wings that require a runway to take off and land or alternatively can be catapult launched. Nevertheless, fixed-wing configurations that require very short runways (STOL) or can take off vertically (VTOL) are also available.

Flight Termination System (FTS) A flight termination system ends the flight of a vehicle and consists of the entire system on an airborne vehicle used to receive, decode, and execute a flight termination command.

Range Safety Group, Range Commanders Council (2007)

Fragmentation The breakup of an in-flight vehicle into fragments (components of the vehicle, pieces of the structure, chunks of solid propellant, miscellaneous hardware, etc.) due to explosive loads, aerodynamic and inertial loads, activation of a flight termination system, intercept with another vehicle, or impact on a surface.

Range Safety Group, Range Commanders Council (2007)

General Aviation General aviation is a term used to describe all nonmilitary and non-airline flying, encompassing everything from recreational aircraft to experimental aircraft to privately owned and operated business jets.

Air Transport Association (ATA) (2008)

Hazard Any real or potential condition that can cause injury, illness, or death of personnel, or damage to or loss of equipment or property. Hazards can be further distinguished as initiating, contributory, and primary. Initiating hazards include events and conditions that start an adverse chain of events that can lead to an accident. Primary hazards are events that directly and immediately cause an accident. Finally contributory hazards are the hazards that are not initiating or primary, although in this term is defined equivalently with hazard.

Range Safety Group, Range Commanders Council (2007)

Federal Aviation Administration (2000)

Hazard Threshold The lowest level at which adverse outcomes are expected to appear.

Range Safety Group, Range Commanders Council (2007)

Hazardous Operation Those activities which, by their nature, expose personnel or property to dangers not normally experienced in day-to-day actions.

Range Safety Group, Range Commanders Council (2007)

Helicopter A helicopter refers to an aircraft that takes off and lands vertically; it is also known as a rotary-wing aircraft with the ability to hover, to fly in very low altitudes, to rotate in the air, and to move backwards and sideways. It is capable of performing nonaggressive or aggressive flights. A helicopter may have different configurations, with a main and a tail rotor (most common), with only a main rotor, with tandem configuration, with coaxial but opposite rotary rotors, as well as with one, two, or four rotors.

Individual Risk Individual risk is the risk that a person will suffer a consequence. Unless otherwise noted, individual risk is expressed as the probability that an individual will become a casualty due to all hazards from an operation at a specific location.

Range Safety Group, Range Commanders Council (2007)

Involuntary Activity No choice was made by the person affected which placed them in a position of increased risk, or the activity participated in or the item used was one that is generally done or used by more than 99 % of the population.

Range Safety Group, Range Commanders Council (2007)

Model Aircraft A small UAS used by hobbyists and flown within visual line of sight under direct control from the pilot, which can navigate the airspace and which

is manufactured or assembled and operated for the purposes of sport, recreation, and/or competition.

Small Unmanned Aircraft System Aviation Rulemaking Committee (2009)

Mishap See *Accident*.

Mission A flight test or operation. It may include multiple vehicles or all phases of the flight beginning with lift-off/launch.

Range Safety Group, Range Commanders Council (2007)

National Airspace System (NAS) The National Airspace System refers to the common network of U.S. airspace, air navigation facilities, equipment and services, and airports or landing areas.

Air Transport Association (ATA) (2008)

Outrage Factor The components of outrage regarding public perception of imposed risk. Such components may include the source of the risk, the transparency/clarity of the risk presentation by the responsible party, the familiarity of the public with the risk, and the level of the perceived risk.

Range Safety Group, Range Commanders Council (2007)

Risk An expression of consequences in terms of the probability of an event occurring, the severity of the event, and the exposure of personnel or resources to potential loss or harm.

United States Air Force (1998)

Risk Analysis A study of potential risk under a given set of conditions. Risk analysis is an activity that includes the complete array of tasks from data gathering, identification of hazards, estimation of associated risks, and verification of results.

Range Safety Group, Range Commanders Council (2007)

Risk Assessment The process of detecting hazards and their causes and systematically assessing the associated risks.

United States Air Force (1998)

Risk Control An action designed to reduce risk by lowering the probability of occurrence and/or decreasing the severity of an identified hazard.

United States Air Force (1998)

Risk Management See *Risk Control*.

Risk Profile A plot that shows the probability of an accident causing a given number of casualties (vertical axis) vs. the number of casualties (horizontal axis). The area under the plot is a measure of the casualty expectation.

Range Safety Group, Range Commanders Council (2007)

Safety Critical A term applied to any condition, event, operation, process, or item whose proper recognition, control, performance, or tolerance is essential to safe system operation and support.

U.S. Department of Defense (2000)

Sensitivity Analysis The computation of the effect of changes in input values or assumptions (including boundaries and model function form) on the outputs. The study of how uncertainty in a model output can be systematically apportioned to different sources of uncertainty in the model input.

Range Safety Group, Range Commanders Council (2007)

By investigating the “relative sensitivity” of model parameters, a user can become knowledgeable of the relative importance of parameters in the model.

Transponder A transponder is an electronic device that “responds” to interrogation by ground-based radar with a special four-digit code that air traffic control specifically assigns to the aircraft on which it is located. Certain transponders have the ability to transmit automatically the altitude of the aircraft in addition to the special code.

Air Transport Association (ATA) (2008)

Uncertainty The absence of perfectly detailed knowledge. Uncertainty includes incertitude (the exact value is unknown) and variability (the value is changing). Uncertainty may also include other forms such as vagueness, ambiguity, and fuzziness (in the sense of borderline cases).

Range Safety Group, Range Commanders Council (2007)

Visual Line of Sight Unaided (corrective lenses and/or sunglasses exempted) visual contact with aircraft sufficient to be able to maintain operational control of the aircraft, know its location, and be able to scan the airspace in which it is operating to decisively see and avoid other air traffic or objects.

Small Unmanned Aircraft System Aviation Rulemaking Committee (2009)

Visual Observer An sUAS flight crew member who assists the sUAS PIC in the duties associated with collision avoidance. This includes, but is not limited to, avoidance of other traffic, airborne objects, clouds, obstructions, and terrain.

Small Unmanned Aircraft System Aviation Rulemaking Committee (2009)

Voluntary Activity The person affected made a choice, which placed them in an increased position of risk compared to the rest of the population. This can be occupation-related (e.g., being a UAS crew member) or not (e.g., doing recreational paragliding).

Range Safety Group, Range Commanders Council (2007)

Waiver Granted use or acceptance of an article that does not meet the specified requirement,

Range Safety Group, Range Commanders Council (1999)

Worst-Case A semiquantitative term referring to the maximum possible exposure, dose, or risk that can conceivably occur, whether or not this exposure, dose, or risk actually occurs in a specific population.

Range Safety Group, Range Commanders Council (2007)

3.4 List of Acronyms

| | |
|---------|--|
| AC | Advisory Circular |
| ACAS | Airborne Collision Avoidance Systems |
| ACL | Autonomous Control Levels |
| ADF | Australian Defence Force |
| ADS-B | Automatic Dependent Surveillance-Broadcast |
| AEG | Aircraft Evaluation Group |
| AFFSA | Air Force Flight Standards Agency |
| AGL | Above Ground Level |
| AIAA | American Institute of Aeronautics and Astronautics |
| AIS | Abbreviated Injury Scale |
| ALARP | As Low As Reasonably Possible |
| ALOS | Acceptable Level of Safety |
| AMA | Academy of Model Aeronautics |
| AMC | Acceptable Means of Compliance |
| A-NPA | Advance Notice for Proposed Amendment |
| ANSI | American National Standards Institute |
| AOPA | Aircraft Owners and Pilots Association |
| ARC | Aviation Rulemaking Committee |
| ARCAA | Australian Research Centre for Aerospace Automation |
| ARINC | Aeronautical Radio, Incorporated |
| ASTRAEA | Autonomous Systems Technology Related Airborne Evaluation & Assessment |
| ASRS | Aviation Safety Reporting System |
| ASTM | American Society for Testing and Materials |
| ATC | Air Traffic Control |
| ATM | Air Traffic Management |
| ATO | Air Traffic Organization |
| ATS | Air Traffic Service |
| ATSB | Australian Transport Safety Board |
| AUVSI | Association for Unmanned Vehicle Systems International |
| AWS | American Welding Society |
| BC | Blunt Criterion |
| BLOS | Beyond line of sight |
| CAA | Civil Aviation Authority (UK) |
| CAR | Canadian Aviation Regulations |
| CASA | Civil Aviation Safety Authority (Australia) |
| CBP | Customs & Border Patrol |
| CEV | Centre d'Essais en Vol (French Flight Test Center) |

| | |
|---------|--|
| CCUVS | Canadian Centre for Unmanned Vehicle Systems |
| CFR | Code of Federal Regulations |
| CGAR | Center of Excellence for General Aviation Research |
| COA | Certificate of Authorization |
| CoE | Center of Excellence |
| CONOPS | Concept of Operations |
| COTS | Commercial Off The Shelf |
| CRD | Comment Response Document |
| CRM | Crew Resource Management |
| CS | Certification Specification |
| CT | Conflicting Trajectory |
| DA | Decision Altitude |
| DoD | Department of Defense |
| DFS | Deutsche Flugsicherung (German Air Traffic Control) |
| DGA | Direction générale pour l'armement (French General Direction for Ordnance) |
| DHS | Department of Homeland Security |
| DLR | Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center) |
| DSP | Defense Standardization Program |
| EASA | European Aviation Safety Agency (EU) |
| EDA | European Defence Agency |
| EIA | Electronic Industries Alliance |
| EFVS | Enhanced Flight Vision System |
| ELOS | Equivalent Level of Performance |
| ELOS | Equivalent Level of Safety |
| ESDA | Electrostatic Discharge Association |
| EUROCAE | European Organisation for Civil Aviation Equipment |
| FAA | Federal Aviation Administration (U.S.) |
| FAR | Federal Aviation Regulations |
| FCS | Flight Control System |
| FHA | Function Hazard Assessment |
| FINAS | Flight In Nonsegregated Air Space |
| FIR | Flight Information Region |
| FL | Flight Level |
| FMEA | Failure Modes and Effects Analysis |
| FSB | Flight Standardization Board |
| FTA | Fault Tree Analysis |
| FTC | Federal Trade Commission |
| FTS | Flight Termination System |
| FY | Fiscal Year |
| GAO | Government Accountability Office |
| GCS | Ground Control Station |
| GI | Ground Impact |
| GPS | Global Positioning System |

| | |
|--------|---|
| HALE | High Altitude Long Endurance |
| HAZOP | Hazard and Operability Analysis |
| HIC | Head Injury Criterion |
| HIP | Head Impact Power |
| HLSC | High Level Safety Criteria |
| HSE | Health and Safety Executive (UK) |
| HSI | Human System Integration |
| IAOPA | International Council of Aircraft Owner and Pilot Association |
| IAP | Instrument Approach Procedure |
| ICAO | International Civil Aviation Organization |
| ICC | Interstate Commerce Commission |
| IEC | International Electrotechnical Commission |
| IEEE | Institute of Electrical and Electronics Engineers |
| IEST | Institute of Environmental Sciences and Technology |
| IFF | Identification Friend or Foe |
| IFR | Instrument Flight Rules |
| INOUI | Innovative Operational UAS Integration |
| INS | Inertial Navigation System |
| ISO | International Organization for Standardization |
| JAA | Joint Aviation Authorities (Europe) |
| JAR | Joint Aviation Requirements |
| JARUS | Joint Authorities for Rulemaking on Unmanned Systems |
| JAUS | Joint Architecture for Unmanned Systems |
| JAXA | Japan Aerospace Exploration Agency |
| JCGUAV | Joint Capability Group on Unmanned Aerial Vehicles |
| JGRE | Joint Ground Robotics Enterprise |
| JIPT | Joint Integrated Product Team |
| JUAS | Joint Unmanned Aircraft System |
| JUAV | Japan UAV Association |
| KIAS | Knots Indicated Air Speed |
| LOS | Line of sight |
| LSA | Light-sport aircraft |
| LUASS | Light Unmanned Aircraft System Scheme |
| MaC | Midair Collision |
| MASPS | Minimum Aviation System Performance Standards |
| MDA | Minimum Descent Altitude |
| MEP | Multi Engine Piston |
| MOPS | Minimum Operational Performance Standards |
| MRE | Multiple Reciprocating Engine |
| MRTFB | Major Range and Test Facility Base |
| MSL | Mean Sea Level |
| MTBF | Mean Time Between Failures |
| MTE | Multiple Turbine Engine |
| MTOW | Maximum Takeoff Weight |
| NAA | National Aviation Authority |

| | |
|--------|--|
| NAS | National Airspace System |
| NASA | National Aeronautics and Space Administration |
| NATO | North Atlantic Treaty Organization |
| NGO | Non-governmental Organization |
| NIST | National Institute of Standards and Technology |
| NOTAM | Notice to Airmen |
| NPA | Notice for Proposed Amendment |
| NTSB | National Transportation Safety Board |
| OED | Operational Environment Definition |
| OSA | Operational Safety Assessment |
| OSHA | Occupational Safety and Health Administration |
| PBFA | Policy Board on Federal Aviation |
| PANS | Procedure for Air Navigation Services |
| PIC | Pilot in Command |
| PLA | Peak Linear Acceleration |
| POI | Principal Operations Inspectors |
| PTF | Planning Task Force |
| RAN | Regional Air Navigation |
| R/C | Remotely Controlled |
| RCC | Range Commanders Council |
| RPAS | Remotely Piloted Aircraft |
| RPAS | Remotely Piloted Air System |
| RTCA | Radio Technical Commission for Aeronautics |
| S&A | Sense and Avoid |
| SAE | Society of Automotive Engineers |
| SARPS | Standards or Recommended Practices |
| SDA | Sense, Detect, and Avoid |
| SDO | Standards Development Organization |
| SEP | Single Engine Piston |
| SESAR | Single European Sky ATM Research |
| SET | Single Engine Turbine |
| SFAR | Special Federal Aviation Regulation |
| SFOC | Special Flight Operation Certificate |
| SMS | Safety Management System |
| SRE | Single Reciprocating Engine |
| SRMP | Safety Risk Management Process |
| SSP | State Safety Plan |
| STANAG | Standardization Agreement |
| STE | Single Turbine Engine |
| sUAS | Small Unmanned Aircraft System |
| SVS | Synthetic Vision System |
| SWIM | System Wide Information Management |
| TCAS | Traffic alert and Collision Avoidance System |
| TDZ | Touchdown Zone |
| TDZE | Touchdown Zone Elevation |

| | |
|-------|---|
| TLS | Target Level of Safety |
| TSO | Technical Standard Order |
| UA | Unmanned Aircraft |
| UAPO | Unmanned Aircraft Program Office (US) |
| UAS | Unmanned Aircraft System |
| UASSG | Unmanned Aircraft Systems Study Group |
| UAV | Unmanned Aerial Vehicle |
| USAR | UAV Systems Airworthiness Requirements |
| USC | United States Code |
| VC | Viscous Criterion |
| VFR | Visual Flight Rules |
| VHF | Very high frequency |
| VLOS | Visual Line of Sight |
| VLR | Very Light Rotorcraft |
| VOR | VHF Omnidirectional Radio Range |
| VUSIL | Validation of Unmanned Aircraft Systems Integration into the Airspace |
| WG | Workgroup |

References

- Air Transport Association (ATA), Learning center (2008), <http://learningcenter.airlines.org/>
- European Aviation Safety Agency, Airworthiness certification of unmanned aircraft systems (UAS). Policy Statement, E.Y013-01, 2009
- Federal Aviation Administration (FAA), FAA System Safety Handbook, Appendix A, Washington, D.C., 2000
- Federal Aviation Administration, Unmanned aircraft systems operations in the U. S. national airspace system. Interim Operational Approval Guidance 08-01, 2008
- M. Fishpool, International military and civilian unmanned aerial vehicle survey. Technical report, Socolofi Research, 2010
- E. Herlik, Unmanned Aerial Vehicles (UAVs) for commercial applications global market & technologies outlook 2011–2016. Technical report, Market Intel Group LLC, 2010
- Joint Capability Group on Unmanned Aerial Vehicles, STANAG 4671 – unmanned aerial vehicle systems airworthiness requirements (USAR). draft, NATO Naval Armaments Group, 2007
- Range Safety Group, Range Commanders Council, Range safety criteria for unmanned air vehicles – rationale and methodology supplement. Supplement to document 323-99, 1999
- Range Safety Group, Range Commanders Council, Common risk criteria standards for national test ranges: supplement. Supplement to document 321-07, 2007
- R. Schultz, Ultralights, LSAs and kit airplanes – what's the difference? Fla. Aviat. Bus. J. (2006) <http://www.airportjournals.com/Display.cfm?varID=0609005>. Accessed Apr 2008
- Small Unmanned Aircraft System Aviation Rulemaking Committee, Comprehensive set of recommendations for sUAS regulatory development, 2009
- United States Air Force, Operational risk management (ORM) guidelines and tools. Air Force Pamphlet 91-215, 1998
- U.S. Department of Defense, Standard practice for system safety. MIL-STD-882D, 2000
- U.S. Department of Defense Office of the Secretary of Defense, Unmanned systems roadmap 2007–2032. Report, 2007

Konstantinos Dalamagkidis

Contents

4.1 Introduction 58

4.2 Precursors of Flight and Unmanned Aircraft 58

4.3 1916–1944 60

4.4 The Machines of the Cold War 61

4.5 Modern Systems 67

References 81

Abstract

Although there are many who believe that UAVs are a recent invention going back at most two or three decades, unmanned flight has a rich history that goes back all the way to ancient times. Of course, the first systems that can qualify with the modern definition of UAVs are quite recent and mainly involve the reconnaissance drones developed and deployed during the cold war. Today, UAV systems have evolved and expanded into widely different designs like quadrotors, ducted fan, and blimps in addition to the classic fixed-wing and helicopter approaches. They have also acquired new roles beyond reconnaissance and have also found applications beyond the military domain, for example, in weather monitoring or infrastructure inspection. This chapter provides a short overview of this history.

K. Dalamagkidis
Institut für Informatik I6, Technische Universität München, Garching bei München, Germany
e-mail: dalamagkidis@tum.de

4.1 Introduction

This section presents a historical perspective on unmanned flight starting from the ancient times and reaching modern times. The purpose is not to provide a detailed historical account of manned or unmanned aviation, but an insight on how UAVs have evolved to be what they are today. The section is subdivided based on the different time periods. This division is not completely arbitrary, but it reflects the evolution of what an unmanned aircraft is, from the first ideas on flying machines in the ancient times up to the industrial revolution, to target drones and missiles, to airborne reconnaissance systems, and to the multirole systems of today.

4.2 Precursors of Flight and Unmanned Aircraft

In modern times, manned aviation appeared in the late 1700s, and it took another century for heavier than air machines to take to the skies. Unmanned aircraft followed soon after the advent of the airplane, appearing around the time of the First World War (1916). However, the idea of building “flying machines” was first conceived close to 2,500 years ago, in ancient Greece and China!

Pythagoras, Archimedes, and others studied the use of autonomous mechanisms for a variety of applications. The first known autonomous flying machine has been credited to Archytas from the city of Tarantas or Tarentum in South Italy, known as Archytas the Tarantine. Archytas has been referred to as Leonardo da Vinci of the Ancient World and was also the father of number one in number theory (Valavanis et al. 2007) and the solution for doubling the cube. He was also possibly the first engineer, designing and building various mechanisms. In 425 BC he built a mechanical bird, which he called “the pigeon,” shown in Fig. 4.1. According to Cornelius Gellius in his *Noctes Atticae*, the bird was made of wood, nicely balanced with weights, and flew using air (most likely steam) enclosed in its stomach (Gellius 1927). It is alleged that Archytas’ pigeon flew about 200 m before falling to the ground, once all energy was used. The pigeon could not fly again, unless the mechanism was reset (Guedj 1998).

During the same era in a different part of the Ancient World – China – at about 400 BC, the Chinese were the first to document the idea of a vertical flight aircraft. The earliest version of the Chinese top consisted of feathers at the end of a stick. The stick was spun between the hands to generate enough lift before released into free flight.

Over the years, the Chinese experimented with other types of flying machines such as hot air balloons, rockets, or kites. It is noteworthy that although some of these machines were used for entertainment, some of the applications were military in nature. In fact there are historical records of a “wooden hawk” that was used for reconnaissance around 450 BC, as well as a kite in the shape of a crow, which was employed during the Ming Dynasty to bomb enemy positions (Yinke 2005).



Fig. 4.1 An artist's depiction of the flying pigeon, the first documented UAV in history. It is reported that it flew for about 200 m

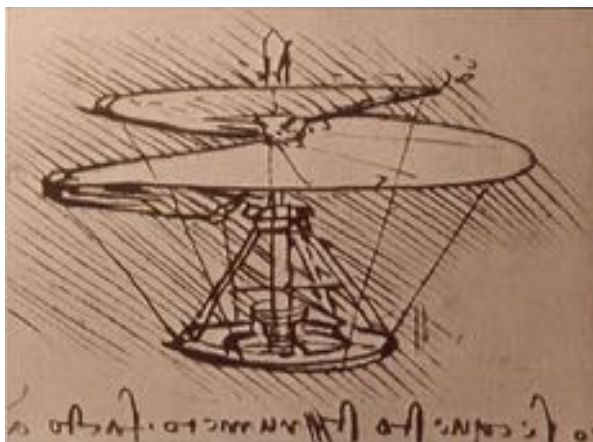
Several centuries later, Leonardo Da Vinci, in 1483, designed an aircraft capable of hovering, called aerial screw or air gyroscope, shown in Fig. 4.2. It had a 5 m diameter and the idea was to make the shaft turn, and if enough force were applied, the machine could spin and fly. This machine is considered by some experts as the ancestor of today's helicopter Hiller Aviation Museum (2004). Da Vinci also devised a mechanical bird in 1508 that would flap its wings by means of a double crank mechanism as it descended along a cable.

The first widely recognized manned flight took place in 1783 using a hot air balloon designed by the Montgolfier brothers and commemorated in Fig. 4.3. Soon after, similar attempts took place in England, and for several years ballooning dominated manned flights, until the first helicopters in the 1860s and later fixed-wing aircraft.

Many flying machines were designed between 1860 and 1909, initially focusing on vertical takeoff and landing aircraft because of the limitations of the steam-powered engines that were in use at the time. As the power to weight ratio of engines improved, these early machines were transformed to the helicopter and airplane designs that are in use today.

The main drive behind aircraft development has always been the fast and safe transportation of people and cargo. Nevertheless, the military soon realized the

Fig. 4.2 Leonardo Da Vinci's air screw, a forerunner of modern helicopter designs (Public domain photo)



potential benefits of unmanned aircraft and efforts to adapt flying machines to operate without a pilot onboard started. Such systems were initially unmanned ordinance delivery systems, what would now be referred to as “missiles” or “smart bombs.” Another use for such systems was to operate as “target drones” that assisted in the training of antiaircraft gun operators. Today UAVs have been defined as those systems that are designed to be recovered after each mission, and although they may carry weapons, the weapon is not by itself an integral component of the airframe (Newcome 2004). Nevertheless, in the early days of unmanned flight, these distinctions were not that important since the technological obstacles were the same and were even shared by manned aircraft as well.

4.3 1916–1944

In 1916, less than 15 years after the Wright brothers historical flight, the first modern unmanned aircraft was demonstrated. It was the Hewitt–Sperry Automatic Airplane, named after the two inventors that designed it. This aircraft could not have become a reality without the previous work of Sperry on gyroscopic devices that were needed to provide flight stabilization. Sperry managed to attract the interest of the U.S. Navy resulting in the development of the Curtiss–Sperry Aerial Torpedo, while at the same time the U.S. Army Air Force sponsored the Liberty Eagle Aerial Torpedo of Charles Kettering shown in Fig. 4.4 (Zaloga 2008). Due to technical problems and lack of accuracy, interest on “automatic” planes was lost, but the potential for use of remotely operated drones for target practice was soon realized.

In Britain, experiments with unmanned aircraft took place throughout the 1920s with the RAE 1921 Target. In 1933, the Royal Navy used the Queen Bee target drone (Fig. 4.5) for the first time (Newcome 2004). It was a modified version of the DeHavilland Tiger Moth biplane and was successfully employed for gunnery practice.



Fig. 4.3 The first manned flight using a hot air balloon took place in 1783 in France (Photo Credit: Bildarchiv Preussischer Kulturbesitz, Berlin)

Remote operation required the perfection of radio control, a concept proposed and demonstrated in 1895 and 1898, respectively, by Tesla (Newcome 2004). Yet again private industry spurred developments, when actor Reginald Denny opened his “Reginald Denny Hobby Shops” in 1934 and started selling radio-controlled airplanes. A few years later, he demonstrated his work for the U.S. Army that led to the development of a very successful target drone used extensively during WWII.

4.4 The Machines of the Cold War

Soon after the end of WWII, interest in reconnaissance missions increased. The descendants of Reginald Denny’s target drones became the basis of the first reconnaissance drone, the SD-1 (Newcome 2004). Also known as the MQM-57



Fig. 4.4 The USAF Liberty Eagle Aerial Torpedo, also known as the Kettering Bug after its creator Charles Kettering (Photo Credit: National Museum of the USAF)

Falconer, it was developed in the mid-1950s, and by the end of its career, close to 1,500 had been built (National Museum of the USAF 2009). The SD-1 (Fig. 4.6) was remotely operated, carried a camera, and after a 30 min flight returned to base and was recovered with parachute (Zaloga 2008).

The loss of the U-2 spy plane over the Soviet Union in 1960 gave a new push towards unmanned reconnaissance drones, and the loss of a second U-2 over Cuba 2 years later helped circumvent any doubts and funding problems (Zaloga 2008). The USAF supported the Ryan Model 147 drone that evolved into a series of models with different capabilities. Two variations of the more than two dozen available are shown in Figs. 4.7 and 4.8. They were also based on a drone design and were used for reconnaissance missions by the USA over China, Vietnam, and other countries in the 1960s and 1970s (Zaloga 2008; Newcome 2004). During this time, close to 3,500 Lightning Bugs were launched, and close to 84 % returned (Wagner 1982). It is noteworthy that these drones were credited with kills that occurred when fighters tried to shoot them down and one drone was given ace status after being responsible for the loss of five North Vietnamese MIGs (Zaloga 2008).

The Ryan Model 147, which became known as the Lightning Bug, was probably the first unmanned aircraft that can withstand today's definition of a UAV aircraft.



Fig. 4.5 The DH.82B Queen Bee drone (Photo Credit: Adrian Pingstone)



Fig. 4.6 The SD-1, also known as the MQM-57 Falconer, was the first reconnaissance drone of the US Army and remained in service until the 1970s (Photo Credit: National Museum of the USAF)

In fact, modernized versions of this drone are still being built and used to carry out missions to this day.

Meanwhile, the US Navy acquired a helicopter drone from the Gyrodine Company called the QH-50 DASH (Fig. 4.9) (Zaloga 2008). This design was preferred because it could be launched from smaller vessels. Its main mission was to launch antisubmarine torpedoes; nevertheless, it was also used for surveillance, cargo transport, and other applications. This was despite reliability issues with its electrical system that led to large number of peacetime losses (Zaloga 2008).



Fig. 4.7 The AQM-34Q, one of several variations of the Ryan Model 147 unmanned reconnaissance drone, used in the 1960s and 1970s (Photo Credit: National Museum of the USAF)



Fig. 4.8 The BGM-34C was another member of the Ryan Model 147 family. It was a multi-role drone capable of performing reconnaissance, electronic countermeasure and strike missions (Photo Credit: US Air Force)

In the late 1960s, the CIA was involved in its own hypersonic, long-range reconnaissance drone, designed to be launched from another “mother” aircraft. The D-21 Tagboard (Fig. 4.10) program was mired with technical problems, accidents, and failed missions that led to its ultimate cancellation in the early 1970s (Zaloga 2008).

Fig. 4.9 A QH-50 DASH-drone armed with torpedoes over the flight deck of the USS Hazelwood (Photo Credit: US Navy)



Fig. 4.10 The Lockheed D-21B Tagboard (Photo Credit: National Museum of the USAF)



Fig. 4.11 The DBR-1 also known as the Tupolev Tu-123 Yastreb (Photo Credit: Tupolev)

In parallel with US efforts, the Soviet Air Force developed its own reconnaissance drones. The first system was the TBR-1 based on a popular target drone, and it was soon followed by the DBR-1 (Fig. 4.11) that allowed for higher range and capabilities (Zaloga 2008). The DBR-1 was not designed to be fully recovered; instead when it reached the recovery area, it dumped fuel, ejected the nose containing the sensor package, and the rest of the airframe crashed. As a result, the DBR-1 involved high operational costs which led to its replacement in the mid-1970s by the Tu-141/143 (Fig. 4.12), a mid- and short-range, respectively, reconnaissance drones that were fitted with parachutes for recovery.

In Europe the unmanned system of choice of the time was funded by Canada and the UK and was developed by Canadair (Zaloga 2008). This resulted in the CL-89 Midge (Fig. 4.13) that was also acquired by the French and German armies. It was designed to follow a preprogrammed course, take photographs (day or night), and return to be recovered by parachute (Zaloga 2008). A more sophisticated version, the CL-289, that also featured better range was developed in the late 1970s with major funding coming now from Germany (Zaloga 2008).

Another major player in the area of unmanned aircraft was the Israeli Air Force that acquired and operated a squadron of American drones for reconnaissance purposes during the Yom Kippur War (Zaloga 2008). Later, the Israeli Aircraft Industries and Tadiran developed their own aircraft, the Scout (Fig. 4.14) and Mastiff, respectively (Zaloga 2008). The Mastiff was the basis of the very popular Pioneer system, and the Israeli designs have also influenced the construction of the Predator and Shadow UAV (Newcome 2004).



Fig. 4.12 The Tupolev Tu-143 Rays reconnaissance drone with SPU-143 launcher at the Ukrainian Air Force Museum (Photo Credit: George Chernilevsky)

4.5 Modern Systems

Modern systems are much more diverse than their precursors. Although one can readily identify the heritage of the reconnaissance drones in the popular Pioneer (Fig. 4.15) of the 1980s or the French-built SPERWER (Fig. 4.16), we have moved towards larger, more capable, and higher endurance systems like the RQ-4 Global Hawk (Fig. 4.17). Systems that can take multiple roles are also available like the MQ-9 Reaper (Fig. 4.18), which besides reconnaissance can also be used as a hunter-killer, and the Neptune (Fig. 4.19) that is used for water operations.

Although some of the aforementioned UAVs like the MQ-9 Reaper can be armed, there is now an entire class of systems being developed with combat operations as their primary mission, known as Unmanned Combat Aircraft Systems (UCAS). Although many of these systems are still in experimental stages, there are several that are already operational. Examples of UCAS include the Neuron (Fig. 4.20), the Barracuda, the Italian Sky-X (Fig. 4.21), the MiG Skat, and the BAE Mantis.

Almost all of the aforementioned systems utilize a fixed-wing design. However, there is a number of helicopter UAVs available, several of which are currently operational in military and civil applications. Some examples of helicopter UAVs



Fig. 4.13 German reconnaissance drone CL-289 (Photo Credit: Karsten Franke)



Fig. 4.14 The Israeli Aircraft Industries Scout drone. The similarity with the Pioneer UAV that was later bought by the US Armed Forces is evident (Photo Credit: Wikimedia Commons User Bukvoed)



Fig. 4.15 An RQ-2 Pioneer ready for launch during operation Desert Shield (Photo Credit: US Marine Corps)



Fig. 4.16 The French SPERWER. It is used by the military forces of several European nations (Photo Credit: David Monniaux)



Fig. 4.17 The RQ-4 Global Hawk is a high-altitude, long-endurance UAV. It was the first UAV to do a transpacific crossing (Newcome 2004) (Photo Credit: US Air Force, Master Sgt. Jason Tudor)



Fig. 4.18 The MQ-9 Reaper is an updated version of the Predator UAV. It is primarily used as a persistent hunter-killer UAV for critical time-sensitive targets and secondarily for intelligence gathering (US Department of Defense Office of the Secretary of Defense 2007) (Photo Credit: US Air Force, Staff Sgt. Brian Ferguson)

Fig. 4.19 The Neptune, a reconnaissance UAV capable of water landings (Photo Credit: US Navy)



Fig. 4.20 The Neuron is an experimental UCAS being developed by a consortium of European companies (Photo Credit: Wikimedia Commons User Tangopaso)



Fig. 4.21 The Sky-X is an Italian UCAS built by the Finmeccanica group. It first flew in 2005 (Photo Credit: Wikimedia Commons User Duch.seb)



Fig. 4.22 The A-160 Hummingbird built by Boeing/Frontier. It is a demonstrator for improvements in range endurance and controllability (US Department of Defense Office of the Secretary of Defense [2005](#)) (Photo Credit: US DoD)



Fig. 4.23 The MQ-8 Fire Scout is being developed by Northrop Grumman Corporation to support US Army and Navy requirements (Public domain photo)



Fig. 4.24 The Aerosonde Laima was the first UAV to make a transatlantic crossing (Newcome 2004) (Public domain photo)



Fig. 4.25 The Helios UAV developed by NASA and AeroVironment. During its second high-altitude flight, it reached 96,863 ft, shattering the existing world altitude record for sustained level flight for both propeller and jet-powered aircraft (Photo Credit: NASA)

include the A-160 Hummingbird (Fig. 4.22), the APID55, the Schiebel S-100, and the MQ-8 Fire Scout (Fig. 4.23).

In recent years there is an increasing interest for long-endurance UAVs that can fly for several days. The Aerosonde Laima (Fig. 4.24) is one such system and was the first to do the transatlantic crossing. The record though is held by the QinetiQ Zephyr which managed to remain airborne for a little over 2 weeks, taking into advantage its lightweight design and solar power. A large number of long-endurance systems are also used for civilian applications. For example, NASA employs the Helios, Altair, and Ikhana UAVs primarily for Earth science missions (Figs. 4.25 and 4.26).

Small UAVs have also garnered significant interest, especially since they are considered by many as entry points to the civilian market. Although their smaller sizes invariably leads to reduced payload capacities, a large number of small and miniature UAVs are in operation or active development. This is because they are versatile, portable, and easy to maintain; they can be employed for the same applications as larger UAVs on a smaller scale and at a lower cost. In fact several of these systems are considered expendable, and no recovery is attempted especially if there is any risk perceived (Fig. 4.26).

Examples of small UAVs include the Skylark, the Evolution, the Puma, and the Aerocon Inspector. In the miniature category, there exist systems that weigh less than 1–3 kg and are easily backpackable (some are even foldable). Such systems include the Cyberbug, Raven (Fig. 4.27), WASP (Fig. 4.28), BATCAM, Nighthawk (Fig. 4.29), and the Dragon Eye.



Fig. 4.26 The Ikhana is a Predator B UAV acquired by NASA and retrofitted for Earth science missions (Photo Credit: NASA)

Fig. 4.27 The Raven is a backpackable UAV for “over the hill” and route reconnaissance that evolved from the Pointer UAV. In the photo paratroopers are training with an RQ-11 Raven (Photo Credit: US Army, Sgt. Amanda Jackson)



Another UAV design concerns the lighter-than-air systems or airships, a distinct UAV category with significantly higher endurance than other systems that makes them ideal for a variety of operations. The main advantage of such systems is that they provide an airborne sensor platform that can be used for persistent area surveillance. Such systems include the Joint Land Attack Elevated Netted Sensor (Fig. 4.30), the Persistent Threat Detection System (Fig. 4.31), and the Rapid Aerostat Initial Deployment system (Fig. 4.32). The High-Altitude Airship (Fig. 4.33) is an untethered platform concept that will be able of providing satellite-like services to ground forces.

Fig. 4.28 The WASP is a MAV that can carry an EO payload and has an endurance of over 1 h (Photo Credit: NIST)



Fig. 4.29 The Nighthawk is the evolution of the BATCAM foldable UAV (Photo Credit: NIST)



In addition to the popular fixed-wing and helicopter systems, other designs are also used for UAVs (Figs. 4.34 and 4.35). These include duct-fan design (e.g., the iStar MAV, the Sikorsky Cypher, and the SELEX Galileo Spyball), counterrotating rotors (e.g., the IT-180 and the KOAX X-240), as well as mixed designs like the Eagle Eye (Fig. 4.36) and X-50 (Fig. 4.37) that combine some of the advantages of fixed-wing and helicopter designs. Finally, the CyberQuad and the AirRobot AR 100-B (Fig. 4.38) are both examples of the quad-rotor design that is especially popular in academic environments.



Fig. 4.30 Joint Land Attack Elevated Netted Sensor (JLENS) by Raytheon/TCOM capable of providing over-the-horizon surveillance (Photo Credit: US Army)



Fig. 4.31 The Persistent Threat Detection System (PTDS) is equipped with a high-resolution EO/IR payload used for surveillance (Photo Credit: US Army)

Fig. 4.32 The Rapid Aerostat Initial Deployment (RAID) is a smaller version of the JLENS, used in missions of area surveillance and force protection against small arms, mortar, and rocket attacks (Photo Credit: US Army)



Fig. 4.33 High-Altitude Airship (HAA) developed by Lockheed Martin. It is a solar-powered, untethered, long-endurance, high-altitude demonstrator (Photo Credit: US Missile Defense Agency)

Fig. 4.34 The iSTAR MAV duct-fan aircraft built by Allied Aerospace (Photo Credit: US Navy)



Fig. 4.35 The Sikorsky Cypher uses a shrouded twin-rotor design (Photo Credit: US Navy)



Fig. 4.36 The Eagle Eye is a tilt-rotor UAV developed by Bell Helicopter. It was selected by the US Coast Guard for its Deepwater program but was put on hold in 2007 (Photo Credit: US Coast Guard)



Fig. 4.37 The X-50 aircraft built by Boeing Corp. It is a technology demonstrator for the Canard Rotor Wing (CRW) configuration which combines hovering capabilities with high cruise speeds (US Department of Defense Office of the Secretary of Defense [2005](#)) (Photo Credit: US DoD)

Fig. 4.38 The AirRobot AR 100-B features autonomous landing when it is out of range or when the battery is low (Public domain photo)



References

- C. Gellius, *Attic nights*. Book 10 (Trans. by Rolfe, J. C.) (1927), http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Gellius/10*.html
- D. Guedj, *Le Theoreme du Perroquet* (Editions du Seuil, Paris, 1998)
- Hiller Aviation Museum, *History of helicopters* (2004). Online, <http://www.hiller.org/>
- National Museum of the USAF, *Radioplane/northrop MQM-57 falconer factsheet* (2009), <http://www.nationalmuseum.af.mil/factsheets/factsheet.asp?id=7684>. Retrieved Oct 2009
- L. Newcome, *Unmanned Aviation: A Brief History of UAV's* (American Institute of Aeronautics and Astronautics, Reston, 2004)
- US Department of Defense Office of the Secretary of Defense, *Unmanned aircraft systems roadmap 2005-2030*. Report (2005)
- US Department of Defense Office of the Secretary of Defense, *Unmanned systems roadmap 2007-2032*. Report (2007)
- K.P. Valavanis, G.J. Vachtsevanos, P.J. Antsaklis, Technology and autonomous mechanisms in the Mediterranean: from ancient Greece to Byzantium, in *Proceedings of the European Control Conference (ECC2007)*, Kos, 2007, pp. 263–270
- W. Wagner, *Lightning Bugs and Other Reconnaissance Drones* (Aero Publishers, Fallbrook, 1982)
- D. Yinke, *Ancient Chinese Inventions* (China Intercontinental Press, Beijing, 2005)
- S.J. Zaloga, *Unmanned Aerial Vehicles: Robotic Air Warfare 1917–2007*. No. 144 in *New Vanguard* (Osprey, Oxford/New York, 2008)

Konstantinos Dalamagkidis

Contents

5.1 Introduction 83

 5.1.1 Classification Based on MTOW and Ground Impact Risk 84

 5.1.2 Classification Based on Operational Altitude and Midair Collision Risk 85

 5.1.3 Classification Based on Autonomy 87

 5.1.4 Military Classifications 89

 5.1.5 Classification Based on Ownership 90

References 90

Abstract

Different UAV classification schemes have been proposed to help differentiate existing systems based on their operational characteristics and their capabilities. Several of these schemes are also of regulatory importance since the metrics used directly correlate with risk from ground impact or midair collision accidents. This chapter provides characteristic UAV classifications from a variety of sources, both civilian and military.

5.1 Introduction

This section aims to present different UAV classification schemes that have been proposed and/or are currently in use. UAV classification is not only used to help differentiate existing systems but has regulatory importance. This is because it is unlikely that rules will be developed that can fit all UAVs and as a result different requirements may be imposed on different UAV categories based on their

K. Dalamagkidis
Institut für Informatik I6, Technische Universität München, Garching bei München, Germany
e-mail: dalamagkidis@tum.de

characteristics. This is also true in manned aviation where, for example, AC 23.1309 already divides FAR Part 23 aircraft into four classes based on MTOW and engine type with different target levels of safety for each one.

There is a large number of metrics that have been used for UAV classification, including mean takeoff weight (MTOW), size, operating conditions, capabilities, or any combination of these and other characteristics. It should be noted that while some of these metrics have minimal effect on the safety performance requirements of the system, they are still important from an operational, commercial, legal, and possibly other points of view. Rather than an exhaustive list which would be of little value, this section presents characteristic examples of different types of classifications from the literature.

A comprehensive classification of UAV demonstrating both the wide variety of UAV systems and capabilities as well as the multiple dimensions of differentiation is presented in Table 5.1.

5.1.1 Classification Based on MTOW and Ground Impact Risk

MTOW is a good metric to classify aircraft for regulatory purposes since it correlates well with the expected kinetic energy imparted at impact, which in turn is considered to be the primary factor affecting safety of operations, as discussed in Weibel and Hansman (2004), Range Safety Group, Range Commanders Council (2007, 1999), Haddon and Whittaker (2002), Joint JAA/Eurocontrol Initiative on UAVs (2004), European Aviation Safety Agency (EASA) (2005), and Dalamagkidis et al. (2012). Of course the broad categories defined in AC 23.1309, do not cover the full spectrum of UAVs that extends down to just a few grams.

In Dalamagkidis et al. (2012), a model was used to estimate the expected number of fatalities after a ground impact. The model was based on the kinetic energy, which was in turn estimated from the MTOW of the UAV. Based on a requirement to maintain an expected number of fatalities of less than 10^{-7} h^{-1} , the classification presented in Table 5.2 was derived. It should be noted that the reliability targets in Table 5.2 were derived based on certain assumptions like a standard population density of 200 ppl/km². For operation over metropolitan areas, this number can be significantly higher and even prohibitive for large systems. On the other hand, the Micro and Mini class UAV, as defined in Table 5.2, are so light that it is almost impossible for a fatality or serious injury to occur after a ground impact. They are also unlikely to cause problems to other aviation, provided that they operate with sufficient clearance from airports, due to their typically low operating altitudes.

A similar classification, but directly based on the kinetic energy, is shown in Table 5.3. In this case the proposed classification scheme directly correlates each UAV class with certification requirements for the aircraft, its pilot, and its operator.

Another simple classification scheme based on MTOW is presented in Table 5.4. This scheme also shows the expected range and maximum operating altitude of the UAVs of each class which are also a function of MTOW.

Table 5.1 UAV categorization for differentiation of existing systems (Source: van Blyenburgh (2006))

| | Mass (kg) | Range (km) | Flight alt. (m) | Endurance (h) |
|---------------------------------------|----------------------------|------------|-----------------|---------------|
| Micro | <5 | <10 | 250 | 1 |
| Mini | <20/25/30/150 ^a | <10 | 150/250/300 | <2 |
| Tactical | | | | |
| Close range (CR) | 25–150 | 10–30 | 3,000 | 2–4 |
| Short range (SR) | 50–250 | 30–70 | 3,000 | 3–6 |
| Medium range (MR) | 150–500 | 70–200 | 5,000 | 6–10 |
| MR endurance (MRE) | 500–1,500 | >500 | 8,000 | 10–18 |
| Low altitude deep penetration (LADP) | 250–2,500 | >250 | 50–9,000 | 0.5–1 |
| Low altitude long endurance (LALE) | 15–25 | >500 | 3,000 | >24 |
| Medium altitude long endurance (MALE) | 1,000–1,500 | >500 | 3,000 | 24–48 |
| Strategic | | | | |
| High altitude long endurance (HALE) | 2,500–5,000 | >2,000 | 20,000 | 24–48 |
| Stratospheric (Strato) | >2,500 | >2,000 | >20,000 | >48 |
| Exo-stratospheric (EXO) | TBD | TBD | >30,500 | TBD |
| Special task | | | | |
| Unmanned combat AV (UCAV) | >1,000 | 1,500 | 12,000 | 2 |
| Lethal (LET) | TBD | 300 | 4,000 | 3–4 |
| Decoys (DEC) | 150–250 | 0–500 | 50–5,000 | <4 |

^aVaries with national legal restrictions

5.1.2 Classification Based on Operational Altitude and Midair Collision Risk

Although MTOW provides a good basis to classify aircraft based on the risk they present to people and property after a ground impact, UAV classes based on altitude may also be of interest since they will dictate to a degree collision avoidance requirements (see Zeitlin 2009). A simple classification was proposed in Dalamagkidis et al. (2012) and is outlined below and presented in Table 5.5:

1. *Very low altitude (VLA/LOS)* operating in Class G airspace and typically in altitudes less than 400–500 ft with the operator always in visual contact with the aircraft
2. *Very low altitude (VLA/BLOS)* as above but with the possibility that the aircraft is flown beyond the line of sight of the operator

Table 5.2 Proposed UAV classification based on MTOW and ground impact risk. T_{GI} is the minimum time between ground impact accidents (Source: Dalamagkidis et al. (2012))

| Category | | | | |
|----------|----------|----------------|------------------|--|
| Number | T_{GI} | MTOW | Name | Notes |
| 0 | 10^2 | Less than 1 kg | Micro | Most countries donot regulate this category since these vehicles pose minimal threat to human life or property |
| 1 | 10^3 | Up to 1 kg | Mini | These two categories roughly correspond to R/C model aircraft |
| 2 | 10^4 | Up to 13.5 kg | Small | |
| 3 | 10^5 | Up to 242 kg | Light/ultralight | Airworthiness certification for this category may be based either on ultralights (FAR Part 103), LSA (Order 8130), or even normal aircraft (FAR Part 23) |
| 4 | 10^6 | Up to 4,332 kg | Normal | Based on MTOW these vehicles correspond to normal aircraft (FAR Part 23) |
| 5 | 10^7 | Over 4,332 kg | Large | These vehicles best correspond to the transport category (FAR Part 25) |

Table 5.3 Proposed classification of UAVs based on MTOW by the CAA of New Zealand (Source: Civil Aviation Authority of New Zealand (2007))

| Class | Maximum energy | UAV | Requirements | |
|-------|----------------|---|------------------------------|--|
| | | | Operator | UAV pilot |
| 1 | Up to 10 kJ | No certification requirement | Regulated by UAV association | Must have pilot license when flying beyond line of sight or in controlled airspace |
| 2 | Up to 1 MJ | Flight permit required | Certificated by CAA | Pilot license, type rating and in some cases instrument rating |
| 3 | Over 1 MJ | Type and airworthiness certificates, maintenance release and continuing airworthiness | Certificated by CAA | Pilot license, type rating and in some cases instrument rating |

3. *Medium altitude (MA)* operating in Class A through E airspace

4. *Very high altitude (VHA)* operating in Class E airspace above FL600

Another proposed classification that can be considered to be indirectly correlated with the midair collision risk is presented in Table 5.6. This classification uses the maximum deviation from the commanded flight path and its duration as the key

Table 5.4 Classification of UAVs based on MTOW (Source: Industrianlagen-Betriebsgesellschaft mbH (2001))

| Class | MTOW (kg) | Range category | Typical max altitude (ft) |
|-------|-----------|----------------|---------------------------|
| 0 | ≤25 | Close range | 1,000 ft |
| 1 | 25–500 | Short range | 15,000 ft |
| 2 | 501–2,000 | Medium range | 30,000 ft |
| 3 | >2,000 | Long range | Above 30,000 ft |

Table 5.5 Proposed classification based on class of airspace used

| Class | Airspace class | S&A | Transponder | 2-way ATC communication |
|----------|----------------|-----------------------|-----------------------|---------------------------|
| VLA/LOS | Class G | Not required | Not required | Not required ^a |
| VLA/BLOS | Class G | Required ^b | Required ^b | Not required ^a |
| MA | Class A–E | Required | Required | Required |
| MA/A | Class A | Required ^b | Required | Required |
| VHA | Above FL600 | Required ^b | Required | Required ^b |

^aCommunication with ATC before operation may still be required^bMay be waived for certain types of operations or under certain conditions**Table 5.6** Source: Industrianlagen-Betriebsgesellschaft mbH (2001)

| Category | Grade of deviation from planned/required flight path | Explanation/definition |
|----------|--|--|
| I | No deviations | |
| II | Minor deviations | Deviations in altitude of not more than 100 ft. Lateral deviations of not more than a nautical mile. UAV is able to correct deviation within 10 s. |
| III | Remarkable/considerable deviations | Deviations in altitude of not more than 500 ft. Lateral deviations of not more than one nautical mile. UAV is able to correct deviation within 30 s. |
| IV | Extreme deviations | Deviations in altitude of more than 500 ft. Lateral deviations of more than one nautical mile. UAV is not able to correct deviation within 30 s. |

differentiation parameters. Of course, the different categories presented can also impose different requirements on pilot certification, onboard control systems, etc.

5.1.3 Classification Based on Autonomy

Another way to categorize UAVs that is also of interest for certification purposes is based on their level of autonomy. Already UAVs are exhibiting autonomy in certain functions, and this trend is expected to increase, especially if one pilot is to control

Table 5.7 Autonomous control levels
(Source: Clough (2002a))

| ACL | Level descriptor |
|-----|--|
| 0 | Remotely piloted vehicle |
| 1 | Execute preplanned mission |
| 2 | Changeable mission |
| 3 | Robust response to real-time faults/events |
| 4 | Fault/event adaptive vehicle |
| 5 | Real-time multi-vehicle coordination |
| 6 | Real-time multi-vehicle cooperation |
| 7 | Battlespace knowledge |
| 8 | Battlespace cognizance |
| 9 | Battlespace swarm cognizance |
| 10 | Fully autonomous |

more than one aircraft at the same time, a possibility discussed in Protti and Barzan (2007). In 2005 the autonomous control levels (ACL) were proposed to measure autonomy. More specifically, ten (10) such levels were proposed in Clough (2002a) that were based on requirements like situational awareness, analysis, coordination, decision making, and operational capability. A list of the ACL is presented in Table 5.7.

Each ACL is based on three aspects characterizing autonomy, namely, the level of independence from human involvement, the complexity of the mission, and the complexity of the environment (Huang 2007). Of course, some of the distinctions between the ACL defined in Table 5.7 may not be of value for regulatory purposes, and some of them are not applicable for civil UAV. A simpler classification that takes into account only the level of human involvement and is compatible with the four operational modes proposed in Federal Agencies Ad Hoc Autonomy Levels for Unmanned Systems (ALFUS) Working Group (WG) (2004) is provided below:

- Remotely piloted: A certified pilot remotely controls the system either within LOS or with feedback from the UA sensors.
- Remotely operated (semiautonomous): The UA is given high-level commands (waypoints, objects to track, etc.), and its performance is monitored by a trained operator. In this case the flying is performed by the UA itself, but all the decision making is delegated to a human.
- Fully autonomous: The UA is given general tasks and is capable of determining how to accomplish them, even at the face of unforeseen events. It can also monitor its health and take remedial action after the occurrence of faults.

It should be noted that as autonomy increases, new regulatory issues will arise. For example, Clough (2002b) and Protti and Barzan (2007) mention that highly autonomous systems may exhibit non-deterministic behavior, something that will be likely prohibited based on current regulatory approaches. It is also evident that issues of liability for highly autonomous systems may also pose questions that will need to be addressed.

Table 5.8 NATO UAV classification guide from the Sep. 2009 JCGUAV meeting (Source: The Joint Air Force Competence Centre (2010))

| Class | Category | Normal operating altitude | Normal mission radius | Example platforms |
|----------------------------|---------------|---------------------------|-----------------------|---|
| Class I (less than 150 kg) | Small >20 kg | Up to 5,000 ft AGL | 50 km (LOS) | Luna, Hermes 90 |
| | Mini 2–20 kg | Up to 3,000 ft AGL | 25 km (LOS) | Scan Eagle, Skylark, Raven, DH3, Aladin, Strix |
| | Micro <2 kg | Up to 200 ft AGL | 5 km (LOS) | Black Widow |
| Class II (150–600 kg) | Tactical | Up to 10,000 ft AGL | 200 km (LOS) | Sperwer, Iview 250, Hermes 450, Aerostar, Ranger |
| Class III (>600 kg) | Strike combat | Up to 65,000 ft AGL | Unlimited (BLOS) | |
| | HALE | Up to 65,000 ft AGL | Unlimited (BLOS) | Global Hawk |
| | MALE | Up to 45,000 ft AGL | Unlimited (BLOS) | Predator B, Predator A, Heron, Heron TP, Hermes 900 |

5.1.4 Military Classifications

There are several military UAV classifications in use. The NATO JCGUAV presented in September of 2009 a classification guide based on MTOW; see Table 5.8. All UAVs are divided into three classes: Class I for those weighing less than 150 kg, Class II for those in the range 150–600 kg, and Class III for those over 600 kg. More information can be found in The Joint Air Force Competence Centre (2010). Class I is subdivided into small (20–150 kg), mini (2–20 kg), and micro. Class III is also subdivided but based on the operational role of the UAV.

The JUAV CoE has defined its own categories that depend on the operational characteristics and other UAV attributes. These categories include tactical, operational, and strategic UAV that have different scope and operate under different commands (U.S. Department of Defense, Office of the Secretary of Defense 2007). Also defined are six levels of domestic use as shown in Table 5.9. A different categorization based on airworthiness requirements is shown in Table 5.10. For all categories, airworthiness and operator qualifications will need to be demonstrated. In addition to that, Cat I aircraft is limited to LOS operations.

Table 5.9 Domestic use UAV levels and corresponding system attributes as defined by the JUAV CoE (Source: U.S. Department of Defense, Office of the Secretary of Defense (2007))

| Level | Airspeed (KIAS) | Weight (kg) | Operating altitude (ft) |
|-------|-----------------|-------------|-------------------------|
| 0 | ≤250 | ≤0.9 | ≤1,200 |
| 1 | ≤250 | 0.9–9 | ≤3,000 |
| 2 | ≤250 | 10–594 | ≤18,000 |
| 3 | ≤250 | 595–5,625 | ≤18,000 |
| 4 | ≥250 | ≤5,625 | ≤18,000 |
| 5 | Any | ≥5,625 | ≥18,000 |

Table 5.10 Military UAV categories and relevant UAV regulations (Source: U.S. Department of Defense, Office of the Secretary of Defense (2007))

| Category | FAA regulation | Airspace usage | Airspeed limit (KIAS) |
|-------------------------------|----------------------------|--------------------------------|-----------------------|
| Cat I – R/C model aircraft | None (AC 91-57) | Class G | 100 ^a |
| Cat II – Nonstandard aircraft | FAR Parts 91, 101, and 103 | Class E,G, and non-joint-use D | 250 ^a |
| Cat III – Certified aircraft | FAR Part 91 | All | None |

^aProposed

5.1.5 Classification Based on Ownership

Finally UAVs – like other aircrafts – can be categorized based on their ownership as *public* or *state* when they are owned and operated by public entities like federal agencies or local law enforcement and *civil* when they are owned by industry or private parties; see Hempe (2006).

References

Civil Aviation Authority of New Zealand, Unmanned aerial vehicles. Wellington, New Zealand, 2007

B. Clough, Metrics, schmetrics! How do you track a UAV’s autonomy? in *Proceedings of the AIAA 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles*, Portsmouth, 2002a

B.T. Clough, Unmanned aerial vehicles: autonomous control challenges, a researcher’s perspective, in *Cooperative Control and Optimization*, chap. 3, ed. by R. Murphey, P.M. Pardalos (Kluwer, Dordrecht/Boston, 2002b), pp. 35–53

K. Dalamagkidis, K. Valavanis, L. Piegel, *On Integrating Unmanned Aircraft Systems into the National Airspace System: Issues, Challenges, Operational Restrictions, Certification, and Recommendations, Intelligent Systems, Control and Automation: Science and Engineering*, vol. 36, 2nd edn. (Springer, Dordrecht/New York, 2012)

European Aviation Safety Agency (EASA), A-NPA, No. 16/2005, policy for unmanned aerial vehicle (UAV) certification, Köln, Germany, 2005

Federal Agencies Ad Hoc Autonomy Levels for Unmanned Systems (ALFUS) Working Group (WG), *Autonomy Levels for Unmanned Systems (ALFUS) Framework – Version 1.1*. NIST, Gaithersburg, MD, USA, 2004

D.R. Haddon, C.J. Whittaker, Aircraft airworthiness certification standards for civil UAVs. UK Civil Aviation Authority, 2002

- D. Hempe, Unmanned aircraft systems in the United States. Presented to the US/Europe international safety conference, Vienna, 2006
- H.M. Huang, Autonomy levels for unmanned systems (ALFUS) framework: safety and application issues, in: *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, Gaithersburg, 2007, pp. 48–53
- Industrieanlagen-Betriebsgesellschaft mbH, CARE innovative action – preliminary study on integration of unmanned aerial vehicles into future air traffic management. Final report, 2001
- Joint JAA/Eurocontrol Initiative on UAVs, A concept for European regulations for civil unmanned aerial vehicles (UAV). Final report, 2004
- M. Protti, R. Barzan, UAV autonomy – which level is desirable? – which level is acceptable? Alenia aeronautica viewpoint, in *Platform Innovations and System Integration for Unmanned Air, Land and Sea Vehicles (AVT-SCI Joint Symposium)*, Florence, 2007, pp. 12–1–12–12
- Range Safety Group, Range Commanders Council, Range safety criteria for unmanned air vehicles – rationale and methodology supplement. Supplement to document 323-99, 1999
- Range Safety Group, Range Commanders Council, Common risk criteria standards for national test ranges: supplement. Supplement to document 321-07, 2007
- The Joint Air force Competence Centre, Strategic concept of employment for unmanned aircraft systems in NATO, Kalkar, Germany, 2010
- U.S. Department of Defense, Office of the Secretary of Defense, Unmanned systems roadmap 2007–2032. Report, 2007
- P. van Blyenburgh, UAV systems: global review. Presented at the Avionics'06 conference, Amsterdam, 2006
- R.E. Weibel, R.J. Hansman, Safety considerations for operation of different classes of UAVs in the NAS, in *Proceedings of the AIAA 4th Aviation Technology, Integration and Operations Forum and AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, Chicago, 2004
- A. Zeitlin, UAS S&A standards: challenges and progress, in *UAS Yearbook 2009/2010* (UVS International, Paris, France, 2009), pp. 134–136

George J. Vachtsevanos and Kimon P. Valavanis

Contents

6.1 Introduction 94

6.2 Military Unmanned Systems 96

6.3 Civilian Unmanned Systems 99

6.4 Discussion 101

6.5 Conclusions 102

References 103

Abstract

This chapter is based on information retrieved from the included references at the end of the chapter. It gives a brief overview of the global unmanned aircraft market, followed by a review of the military and civilian unmanned systems sectors. It aims at registering the current state of the art in the unmanned aviation arena projecting what lies ahead.

G.J. Vachtsevanos (✉)
Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA
e-mail: gjv@ece.gatech.edu

K.P. Valavanis
John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA
e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

6.1 Introduction

The unmanned aircraft arena has seen unprecedented levels of growth over the past 20 years with military applications dominating the field. However, the 2013 year was the critical and the turning point year for using unmanned aircraft in civilian and public domain applications for the following three important reasons:

- On 20 June 2013, the European Remotely Piloted Aircraft Systems (RPAS) Steering Group, ERSR, which was set up by the European Commission in July 2012 and was composed by a group of stakeholders of the main organizations and experts interested in the integration of RPAS into the European aviation system, handed over to the European Commission the “*Roadmap for the Integration of Civil Remotely Piloted Aircraft Systems into the European Aviation System*.” This was the result of the mandate received by the ERSR to establish a roadmap for the safe integration of civil RPAS into the European aviation system, aiming at an initial RPAS integration by 2016. This roadmap aims at facilitating decisions taken by the involved organizations, providing transparency and efficiency in planning different initiatives, and supporting the coordination of related activities in Europe.
- On 7 November 2013, the U.S. Federal Aviation Administration (FAA) presented the “*Roadmap for Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS)*,” which was the result of the joint effort between the FAA and the UAS Aviation Rulemaking Committee (ARC). The purpose of this roadmap is to outline the tasks and considerations that need to be completed in order to enable UAS integration into the NAS. This roadmap is the aftermath of, and in accordance with, the Congressional mandate in the *FAA Modernization and Reform Act of 2012*, Pub. L. 112-95. It is expected that this roadmap will be updated accordingly in subsequent publications, incorporating lessons learned and findings and also revising target dates toward efficient integration of UAS into the NAS.
- On 30 December 2013, the FAA announced the selection of six UAS research and test site operators across the USA that are tasked to study a broad range of issues and challenges related to unmanned aircraft. The six test sites can operate until 13 February 2017, with Congress expected to consider extending the test sites beyond that date. This announcement was followed by the statement that “The FAA is committed to the safe and efficient integration of UAS into the NAS, thus enabling this emerging technology to safely achieve its full potential.”

Further, the International Civil Aviation Organization (ICAO), which is a special agency of the United Nations (UN), has long been committed to “the safe and orderly development of international civil aviation throughout the world,” being the organization that also sets standards and regulations necessary for aviation safety, security, efficiency, and regularity, as well as aviation environmental protection.

The above actions from the ERSR and the FAA along with the backing of ICAO have cautiously opened the door to utilizing unmanned aircraft in civilian and public domain applications. A “domino effect” is expected to accelerate research

and development; increase funding; create new jobs; generate provable novel and cutting-edge technologies for increased safety and security of unmanned aircraft; contribute to developing functional and operational “standards” for unmanned aviation; enforce closer collaboration and coordination among national and international interested parties; bring closer academia, government, industry, and the private sector; and engage interested parties to discussing and eventually resolving ethical, legal, and privacy issues, to say the least and name but a few opportunities.

It has been estimated that the international UAV market will be worth up to \$80 billion by 2020. The USA is the world’s largest operator of unmanned aircraft and the front runner in producing and using technology in military services since the 1950s, also expected to continue to represent the biggest market as a result of requirements, spending on the systems, and research and development funding into the technology. The USA was the highest spender on UAVs in 2009 with the North American region (including Canada) spending \$2.7 billion. The results from the continued deployment of UAS in Afghanistan and the surrounding area as well as Iraq have also contributed to their increased use and procurement. The U.S. Department of Defense spent more than \$17 billion up to 2013 for new UAV systems and technology with \$5.0 billion on UAV research and development. The international market for UAVs/UAS has also grown in recent years with other countries worldwide, recognizing the capabilities and widespread utility of unmanned systems and investing heavily in these new technologies. It is anticipated that, as a result of the previous announcements, spending and investment will increase and the market numbers will shortly be revised.

Before continuing with a review of the military and civilian use of unmanned aircraft, and since several terms are used to describe such systems, a brief explanation is provided to clarify the context in which such names are used.

The terms UAVs and UAS have distinct definitions and it should be clear why, at first, UAVs became UAS: A UAV refers to a pilotless aircraft, a flying machine without an onboard human pilot or passengers. As such, “unmanned” implies total absence of a human who directs and actively pilots the aircraft. Control functions for unmanned aircraft may be either onboard or offboard (remote control). The term UAS was introduced by the U.S. Department of Defense, followed by the FAA and the European Aviation Safety Agency (EASA). This term was meant to signify that UAS are aircraft and as such airworthiness will need to be demonstrated and they are also systems consisting of ground control stations, communication links, and launch and retrieval systems in addition to the aircraft itself. Similar distinctions are true for RPA versus RPAS. Other names include remotely piloted vehicles (RPVs), a term that was used in the Vietnam War, while the U.S. military also calls them RPA a term used to include both the aircraft and the pilot, while the UK and ICAO have designated them as RPAS, to demonstrate the presence of the man in the loop to control them. The ICAO recommended definition for RPAS is “A set of configurable elements consisting of a remotely-piloted aircraft, its associated remote pilot station(s), the required command and control links and any other system elements as may be required, at any point during flight operation.”

6.2 Military Unmanned Systems

There exist many different types of unmanned aircraft depending on size, payload, and endurance. There is increased interest for Mini UAVs (MUAVs) and Micro Air Vehicles (MAVs) mostly used for short-range close reconnaissance tasks (mostly by Special Forces) and in urban environments, usually operated at company or battalion level. Lightweight MUAVs/MAVs are basically launched by hand or by a bungee catapult. MUAVs/MAVs can usually be disassembled and be carried in a backpack. Power is mostly provided by lithium polymer, lithium sulfide, or zinc-air batteries, with range typically being no more than 20 km and endurance of up to 2 h. Application wise, the MUAV/MAV and the ground control station (GCS) usually have line-of-sight (LOS) air-to-ground radio relay or microwave links due to the short-range limitation. Although MUAVs/MAVs have extensively been used by the military deployed in Iraq and Afghanistan, they are also suitable for civilian use being relatively inexpensive to purchase and operate. As such, using swarms or multiple MAVs is projected to be a key area in future UAV development that will enable more effective intelligence, surveillance, and reconnaissance (ISR) due to the increased availability of multiple sensors during a mission.

Larger UAVs/UAS operated in Afghanistan and Iraq have demonstrated their usefulness in Counter-Improvised Explosive Device (C-IED) tasks. Vertical Take-Off and Landing UAVs (VTOL UAVs) are particularly seen as suitable for C-IED with the air vehicle capable of hovering at a distance to find and locate Improvised Explosive Devices (IEDs). VTOL UAVs are additionally seen as useful in urban environments, for maritime ISR tasks from naval vessels, and in areas where normal fixed-wing UAV operations might be difficult due to terrain, threats, or insufficient support personnel deployed at forward bases to recover the UAV. Militaries have largely procured Tactical UAVs (TUAVs) to provide ISR support to ground forces with ranges typically up to 250 km and an endurance of up to 15 h. TUAVs either are catapult launched or have automatic takeoff and landing (ATOL) capabilities. The USA also uses and refers to a class of UAV systems smaller than the TUAV but larger than the MUAV as the small UAS (SUAS or sUAS).

Medium-altitude long-endurance (MALE) and high-altitude long-endurance (HALE) unmanned aircraft offer low-cost alternatives to manned aircraft for strategic reconnaissance tasks, with some having an armed capability to undertake close air support (CAS). GCS operators use beyond-line-of-sight (BLOS) satellite data links, typically Ku-band, to control and receive data from these MALE/HALE UAVs and C-band LOS data links for shorter-range missions, takeoff, and landings. MALE/HALE UAVs provide persistent ISR at long ranges being equipped with electro-optical/infrared (EO/IR) sensors and ground surveillance radar systems without the limitations of manned aircraft, particularly the physiological effect on aircrew flying long endurance missions. Combined solar/electric or hydrogen-powered HALE UAVs could have an endurance of months or years. Some unmanned HALE platforms like solar-powered stratosphere UAVs and lighter-than-air (LTA) airships are likely to be more cost-effective alternatives to manned weather aircraft, satellites, and other space-based platforms. LTA platforms are particularly cheaper

alternatives to conventional MALE/HALE UAVs. The average LTA airship unit cost is estimated to be \$3.0 million with flying operating costs about \$50 per hour, compared to up to \$10,700 for the MQ-9 Reaper or the Heron 1 UAV that includes the costs of fuel, maintenance, and deployment of personnel.

Unmanned combat aerial vehicles (UCAVs) are expected to replace and complement manned strike aircraft in the future, providing an intelligence, surveillance, target acquisition, and reconnaissance (ISTAR), Suppression of Enemy Air Defenses (SEAD), and deep penetration strike in environments where there is a high threat from enemy air defenses. However, at this stage, technologies have not matured to the point to enable UCAVs to replace manned air superiority fighter aircraft. Very low-observable (VLO) or low-observable (LO) stealth characteristics must be incorporated into the UCAV's airframe to reduce its radar cross section (RCS), infrared (IR) signature, and radio frequency (RF). Weapons and sensors are carried internally to further reduce the UCAV's RCS. In comparison to their manned aircraft equivalents, UCAVs promise to be much cheaper to develop and procure, although support costs might be high to maintain airframe coatings and materials that give the aircraft its stealth capabilities.

A potential future UAV application is for delivering cargo and supplies to frontline troops or in a humanitarian aid operation, as well as for casualty or medical evacuation in areas considered too risky to operate manned land vehicles or helicopters. The USA and Israel are currently investigating UAV technologies in these roles, which may as well transfer to the civilian domains.

Another type of military UAV is the so-called attack UAV. This type of UAVs is fitted with a high-explosive warhead that is flown by a GCS operator and then loiters over a target area using their onboard sensors to identify and then attack a target. The main attack UAV is the Harop manufactured by Israel Aerospace Industries (IAI). The delta-wing Harop carries a 23 kg warhead. The European missile company MBDA has also developed an attack MAV designated the Tactical Grenade Extended Range (Tiger) that carries a warhead less than 0.5 kg.

The U.S. Army has experienced a significant rise in the use of UAVs with nearly 4,000 air vehicles now in service compared to 54 a decade ago. The U.S. Army's Aviation Center at Fort Rucker, Alabama, is now home to the U.S. Army Unmanned Aircraft Systems Center of Excellence (UAS CoE), which is tasked with integrating and coordinating Army UAV use as well as defining current and future UAV concepts for the service. The "U.S. Army Roadmap for UAS 2010–2035," published by UAS CoE in April 2010, defines the service's long-term UAV procurement plans and how these will be integrated and operated. At battalion level, the U.S. Army operates the RQ-11B Raven MUAV manufactured by AeroVironment located at Monrovia in California.

The electrically powered 1.9 kg Raven entered U.S. Army service in 2006. It has a length of 0.9 m and a 1.37 m wingspan. The Raven can carry a 0.18 kg payload comprising a 5× zoom EO camera and IR camera equipped with a laser rangefinder that can designate targets at a range of 25 km. It has a maximum speed of 83 km/h, a range of up to 10 km, and an endurance of 90 min. U.S. Army Raven training is undertaken by the 2nd Battalion, 29th Infantry Regiment, and 197th Infantry Training

Brigade at Fort Benning in Georgia. Preliminary UAV training is undertaken at the UAS Training Battalion (UASTB) at Fort Huachuca, Arizona. The Raven is also operated by the Netherlands, Italy, and Spain, and it has also been leased in the past by the UK. The U.S. Army has spent some \$232 million on R&T and the procurement of the Raven, totaling more than 2,000 air vehicles with a total requirement for 3,000. About \$37.6 million was spent on 312 air vehicles in 2011 with orders to continue through to 2014. The Raven is also in service with the U.S. Marine Corps (USMC) with more than 1,020 acquired since 2008 at a cost of about \$71 million, with 16 more air vehicles procured in 2011 at a cost of \$32.5 million. The U.S. Army's SUAS Product Improvement Program (SUAS PIP) plans upgrades to the existing Raven UAS, such as new modular sensor suites for enhanced C-IED and CBRN tasks, an increased range of 15 km, and interoperability with unmanned ground vehicles (UGVs) and unmanned ground systems (UGS).

After canceling the Future Combat Systems (FCS) program and the acquisition of further MUAVs/MAVs, the US Army now focuses its efforts on acquiring systems like the AeroVironment Wasp and the larger Puma AE (All Environment) as well as the gMAV (Gasoline Micro Air Vehicle) VTOL MAV. Nano-UAVs and other MAVs are also planned with the U.S. Army previously testing in 2004 the 0.8 lb Tactical Mini UAV (Tacmav) manufactured by Applied Research Associates (ARA). The Tacmav is now marketed by ARA as the Nighthawk. The electric-powered 0.43 kg Wasp has a 0.72 m wingspan, an endurance of 45 min, and a 5 km range. It was developed by AeroVironment with the Defense Advanced Research Projects Agency (DARPA) with more than 200 air vehicles acquired by the USAF Special Operations Command (AFSOC) as part of its Battlefield Air Targeting Micro Air Vehicle (BATMAV) program in 2006. Full rate production of an unspecified number of Wasp IIIs was approved by AFSOC in 2008. The USMC also ordered the MAV in 2007 at a cost of \$19.3 million being assigned to units at platoon level. The Puma AE is a hand-launched electric 5.9 kg MUAV with a wingspan of 2.8 m. It has an endurance of 2 h and a range of 15 km. It shares a GCS with the Raven and the Wasp. It was selected by U.S. Special Operations Command (USSOCOM) in 2008 to meet its \$35 million All Environment Capable Variant (AECV) requirement for a cost-effective shipborne UAV. It entered service in 2009 replacing an earlier version of the Puma. The gMAV is manufactured by Honeywell Defense & Space at Albuquerque, New Mexico, as the T-Hawk (Tarantula-Hawk) and is largely used for C-IED tasks with the US Army and joint EOD units. It is designed to operate in urban environments, and 90 systems (180 air vehicles) have been acquired by the U.S. Navy. The 11 kg ducted fan gMAV is 0.38 m in length and has a 0.33 m diameter. It can carry a 0.9 kg payload of interchangeable gimballed EO or IR sensors as well as avionics and RF radio pods. The gMAV is powered by a 4.4 hp (3 kW) heavy fuel boxer twin engine, providing a maximum speed of about 74 km/h and the ability to fly in a 37 km/h wind. It has an endurance of 50 min, a range of 10 km, and a maximum service ceiling of 10,000 ft. About 83 gMAV Block I have been delivered since 2007 with plans for up to 166 to be acquired. Block II/III featuring a gimballed sensor payload has since been fielded by units in Afghanistan with the type due to equip Infantry Brigade Combat Teams (IBCT) from 2011. A larger 18 kg

version powered by a heavy fuel engine has also been developed for the U.S. Army's Brigade Combat Team Modernization (BCTM) Increment two program with first flight in 2012. The U.S. Army operates at brigade level the AAI RQ-7B Shadow 200 TUAV, which was delivered in 2004–2008. Some \$350 million has been spent on the acquisition of the Shadow for the U.S. Army since 2007. The 208 kg Shadow 200 features a twin-boom and inverted V-tail. It is 3.6 m in length with a height of 1.0 m and a wingspan of 6.2 m if equipped with an enhancement kit. It can carry a payload of up to 36 kg that includes a 35 lb IAI Tamam Plug-in Optronic Payload (POP) POP-300 IR/EO sensor with laser pointer/rangefinder that is capable of detecting targets up to 10 km. The Shadow is powered by a single 38 hp (28 kW) UEL AR741 Wankel engine that provides a maximum speed of 203 km/h or a cruise speed of 166 km/h. Range is 110 km with an endurance of up to 9 h and maximum altitude of 15,000 ft. The Shadow has a Tactical Automatic Landing System (TALS) or can be launched from a hydraulic catapult launcher. The complete Shadow UAS comprises four air vehicles, two GCS and ground data terminals, one portable GCS, and four RVTs. Three air vehicles are transportable by an air vehicle transporter (AVT). The fourth is used as a spare. Some 48 Shadow UAVs were ordered in 2007 by the USMC to replace the RQ-2 Pioneer with deliveries being completed in 2009. In January 2010, AAI was awarded a \$39 million contract for the supply of two air vehicles for the U.S. Army and one for the USMC with deliveries in March 2011.

Growing military UAV use has mostly been driven by the USA's operational experience of using these platforms in most theaters, Israel's adoption of the systems for intelligence and reconnaissance tasks, as well as usage in the Kosovo War, in Afghanistan, and Iraq. UAV technology has now much improved and become more reliable, although there do still remain some problems. Nevertheless, it has been forecast that a large percentage of manned military aircraft missions will be undertaken by UAVs in the near future. It is also projected that the military UAS market will remain of higher value than the civilian market for the foreseeable future due to a higher allocation of funding drawn from national government defense budgets, although tightened spending could have an effect on some programs and procurement. Industry has provided additional funding toward research and development costs of national UAV programs or has developed their own systems with the hope of attracting of domestic and international orders.

6.3 Civilian Unmanned Systems

Possible civilian UAV applications include scientific research, search and rescue, emergency response, traffic control tasks, infrastructure support, aerial photography, forest protection and wildfire monitoring, environmental monitoring, energy and electrical facility monitoring, pipeline inspection, and coast guard support, to name but a few possible applications. Further, UAVs may also be used for crisis management operations during and after natural disasters like hurricane Katrina and Sandy, or after terrorist attacks, to survey disaster zones or to look for survivors. The use of UAVs to transport civilian air cargo could be a lucrative area in the future.

In the short term, earth observation and surveillance are expected to account for the biggest share of the civilian UAV market.

Increasing opportunities within this market has led to the establishment of numerous small and medium enterprises (SMEs) developing low-cost systems for civilian applications. However, there remains the danger of duplication as too many companies are formed offering similar systems. Europe in particular now offers the most diverse choice of UAVs compared to any other region in the world, but may struggle to find sufficient demand worldwide to guarantee that they will be able to focus on UAV technology in the long term. The development of low-cost systems will be a major factor in sales. To ensure low-cost UAV development, some major companies have opted to convert off-the-shelf manned platforms into UAVs. The BAE Systems Herti, for example, is based on the J6 Fregata motor glider developed by Poland's J&AS Aero Design. The company has additionally used the MT-03 autogyro, which is produced by RotorSport, UK, in partnership with Germany's AutoGyro, as a basis for its Ampersand UAV. France's Sagem has converted the German Stemme S15-VT powered glider into the patroller. The USA's Aurora Flight Services, Russia's Irkut, and Israel's Aeronautics Defense Systems have all opted to convert the Austrian Diamond DA42 Twin Star twin-engine aircraft into a UAV or optionally piloted vehicle (OPV) that can be flown manned or unmanned. Italy's Selex Galileo, meanwhile, has been working with the Italian UAV company Unmanned Technologies Research Institute (UTRI) to develop the Short Take-Off and Landing OPV (STOL OPV) surveillance platform based on the two-seat Xenon gyroplane produced by Poland's Celier Aviation.

The level and the degree of use of some VTOL UAVs, TUAVs, and MALE UAVs for law enforcement and ground and border surveillance will basically be determined by funding from national governments with a leased fee-for-service arrangement more likely in an effort to keep costs down. An outright UAV purchase might not be seen as financially viable if taking into account that a complete UAS like the BAE Systems Herti will not cost much less to buy than a single Eurocopter EC145 – a platform being regularly chosen for police air support tasks. A manned platform like a helicopter is likely to remain the preferred choice as it provides more versatility, although a UAV has longer endurance time, lower operating costs, and a smaller logistics footprint. Outsourcing where a private contractor provides UAV services is already a growing trend among cash-strapped European militaries.

Focusing on law enforcement tasks and in addition to the surveillance role, some UAVs could be fitted with a Long Range Acoustic Device (LRAD) loudspeaker and lightweight nonlethal weapons to reduce the risk to police officers on the ground. France's Tecknisolar Seni test flew its Bourdon MUAV in 2004 fitted with a 1.1 kg Verney-Carron Flash-Ball nonlethal defense weapon that fires 44 mm rubber balls, plastic rounds, tear gas, or dye-ball markers. France's SMP Technologies plans to incorporate a 0.5 kg taser stun gun, which it manufactures into an I-Drone quadrotor UAV for law enforcement tasks. Austria's Schiebel has already flown its Camcopter S-100 VTOL UAV with a LRAD in a military test.

There is currently a wide array of UAV applications in the civilian world, and many more are envisioned for the future. Among them, UAVs monitoring the Arctic,

deployed to survey areas affected by earthquakes, monitoring marine mammals, use for civilian scientific research or environmental monitoring, weather/atmospheric data collection, oceanographic data collection, agricultural monitoring, and high-altitude geological mapping of magnetic, radiological, and gravimetric data and more, among many others.

6.4 Discussion

There is no doubt that the international market for UAVs or UAS or RPA or RPAS has grown in recent years. The use of these systems instead of manned platforms has particular benefits for an operator, namely, lower procurement costs than for manned aircraft and a small logistics footprint. There is additionally a low risk to human life. Military use is currently the biggest market for UAVs and likely to do so for the foreseeable future, despite growing interest in the systems for civilian uses. Europe and the rest of the world were initially fairly slow to adopt UAVs with the USA and Israel using systems operationally from the 1970s. Earlier UAV technology was unreliable with flight control systems and sensor technology, endurance, and weather conditions limiting usage. UAV technology has now much improved and become more reliable, although there do still remain some problems.

Military UAVs are generally operated in restricted airspace away from civilian air traffic, but serious problems are posed when both military and civilian operators start to use the systems in congested airspace, including possible midair collision. Current major efforts are trying to put in place general harmonization rules for UAV operations within civilian or controlled airspace with each state responsible for its own air regulations. Several regional efforts have been launched to draw up guidelines for UAV use in controlled airspace.

Despite progress, there is still need for new technology to ensure that UAVs can operate safely in civilian airspace. One important area is the development of sense and avoid (S&A) technology that will enable the UAV to autonomously identify a manned aircraft and then make corrections to its flight path to avoid a midair collision. Integrating UAVs into congested airspace remains one of the main areas of current studies that will draw on existing and future UAV technology.

There are still other problems to overcome, particularly for the military, to ensure that UAVs can remain operable, such as the need to develop secure data links to prevent possible hostile jamming and data sent from the UAV's sensors from getting into the wrong hands. The USA and Israel have already had problems with enemy forces intercepting unsecure data broadcast from UAVs. There is additionally the problem of nations and hostile forces developing and gaining anti-UAV technology. The USA's Raytheon has already successfully used its Laser Phalanx naval close-in weapon system to shoot down a UAV in tests. There is additionally the problem of the loss of data links between the GCS and the UAV, which has been the most common cause of crashes for the past decade. There is particularly a need to allocate sufficient satellite bandwidth to control UAVs and to provide a means of transmitting communications and real-time data from sensors to ground control stations and

other receiving facilities. The USA has had problems with the allocation of enough bandwidth that has led it to use commercial satellite operators in addition to its own military satellite systems. The London Satellite Exchange (LSE), for example, is one commercial operator that has been providing Ku-band links for UAVs for several countries, including France. Research is being undertaken to address some of these issues. In Europe, for example, a study is being conducted by the European Space Agency (ESA) on European and Canadian satellite technologies. The Advanced Research on Telecommunication Satellite Systems (ARTES) program also includes planned studies into satellite communications (satcom) systems for civilian UAVs. A separate study dubbed SIGAT has also worked on identifying appropriate military radio spectrum for UAV integration into general air traffic in Europe. UAVs featuring autonomous flight capabilities could lower bandwidth use with the operator just responsible for guidance to the target area or making the decision to release weapons.

Some of these potential roles for UAVs could, however, be controversial, and their use by law enforcement agencies might not be fully supported by the public. There are also issues about the export and proliferation of UAVs. The informal Missile Technology Control Regime (MTCR) prevents the proliferation of certain UAVs, particularly systems that could be used to deliver weapons of mass destruction. It has restricted the sale of MALE systems, as has the USA's own export laws. Several UAV manufacturing nations, such as China, are not part of the MTCR that has led to concerns that the technology could end up in the hands of terrorist organizations. MTCR has, however, restricted sales of UAVs to nations outside Europe. The USA's General Atomics Aeronautical Systems Inc (GA-ASI) has been keen to export of an unarmed version of the Predator MALE UAV, dubbed the Predator XP.

6.5 Conclusions

Although UAVs have been and are a familiar presence in war zones, the picture is changing and roadmaps have been produced to integrate UAVs into the global airspace. The civilian sector is attracting, of course, the interest of UAV OEMs and users, worldwide. In some cases, the civilian market is driven by specific needs such as border patrol, forest fire monitoring, rescue operations, etc. In other cases, the need is more generic and the UAVs may find multiple application domains, like traffic monitoring, police surveillance, etc. UAV applications are set to explode in the commercial market once airspace regulations are clearly defined and published. The Association for Unmanned Systems International (AUUSI) and UVS International hold annual shows and conferences lobbying governments for greater access to civilian skies.

The airspace that belongs to manned systems still presents a controversial barrier to the civilian UAV market segment as regulatory agencies had limited significantly UAV access to the national airspace. For example, the FAA's line-of-sight restriction and other certification requirements have been a major roadblock. However, the

2013 year is a turning point year as roadmaps for integrating UAS into the national airspace have been published both in the USA and Europe.

Regardless, we are witnessing a growing role of relatively low-cost, nearly expendable UAVs in military and civilian operations. The use of these systems, instead of manned platforms, is providing particular benefits to the operator, namely, lower procurement costs and a small logistics footprint. UAVs are currently a growth industry in the aviation sector, with scores of new companies competing for a slice of the market. We are about to witness major changes as UAV technologies are becoming more reliable and safer, and their utility is expanding rapidly, accompanied by demand and the rapid growth of the industry. Reliability is likely to be a key issue for UAVs aimed at civilian use as the industry continues to lobby aviation regulators to gain access to skies that for the most part remained off-limits. A major challenge that needs to be addressed and solved before widespread use of UAVs relates to privacy, legal, and ethical issues.

References

- Army Unmanned Aircraft System Operations, FM-3-04-155, 14 May 2008
- FY2009–2034 Unmanned Systems Integrated Roadmap, Office of Secretary of Defense, 6 Apr 2009
- <https://extranet.acq.osd.mil/uwir/>
- Joint Concept of Operations for Unmanned Aircraft Systems, Joint Requirements Oversight Council, Nov 2008
- Joint Direct Support of Aerial Intelligence Surveillance Reconnaissance, (JDSAIRS) Initial Capabilities Document (ICD), 6 Aug 2010
- W. Lewis, A. Boydston, Qualification and reliability of complex electronic rotorcraft systems, in *AHS Specialists' Meeting on Systems Engineering*, Hartford, 15–16 Oct 2009
- Operational Environment 2009–2025, TRADOC, Aug 2009. (v5)
- Sense and Respond Logistics: Co-Evolution of an Adaptive Enterprise Capability: DoD Office of Force Transformation, Feb 2004
- Sense & Respond Logistics Technology Roadmap, Office of the Deputy Under Secretary of Defense, Mar 2009
- The U.S. Army Unmanned Aircraft Systems Strategic Vision 2007, Unmanned Futures Branch, CRD
- The U.S. UAS Future's White Paper, 1 Oct 2009
- Unmanned Aircraft System Airspace Integration Plan, Office of the Secretary of Defense, Draft Version 1.7, 30 Oct 2009
- Unmanned Ground Systems Roadmap, Robotic Systems Joint Project Office, July 2009
- Unmanned Ground Systems Roadmap, Robotic Systems Joint Project Office, Addendum, July 2010
- Unmanned Systems Integrated Roadmap, FY2011–2036, DOD Reference Number: 11-5-3613
- U.S. Army Roadmap for UAS, 2010–2035

Section II

UAV Design Principles

Daniel J. Pack and Graham Drozeski

Kimon P. Valavanis and George J. Vachtsevanos

UAV Design Principles presents the basic design principles of UAVs and the components that comprise a complete unmanned aircraft system, avoiding in depth technical details. It aims at creating the “overall picture” with respect to UAV functionality and, consequently, to indirectly motivate the reader about the challenges that need to be overcome in order to make UAVs fully operational, reliable, and safe.

► **Computational and Experimental Design of a Fixed-Wing UAV** by Ryaciotaki-Boussalis and Guillaume presents a comprehensive methodology for the design and manufacture of all aspects of a prototype fixed wing UAV, which also allows for utilizing multi-segmented flight control surfaces on the wings of such UAVs. Adding multiple segments to UAV wings creates smaller control surfaces; thus, refined adjustments to UAV performance while airborne may be accomplished.

► **UAV Handbook: Payload Design of Small UAVs** by Gruber, Kwon, Hager, Sharma, Yoder, and Pack discusses payload design issues for small UAVs, which are essential to overcome various small UAV constraints imposed by stringent weight, power, and volume. The efficacy of design principles is demonstrated with the help of the payloads for a fixed wing UAV developed by the Center for Unmanned Aircraft Systems Research at the U.S. Air Force Academy, and it is shown that the system requirements for specific applications are closely related to those for other small UAV applications.

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

► **Small UAV Design Development and Sizing** by Brandt describes a rationale for selecting a low-risk UAV configuration and a methodology for sizing and optimizing the shape of both fueled and electric-powered aircraft. Three examples of low-risk UAV configurations are shown to be essentially similar from the point of view of optimization and sizing. The sizing methodology for fueled aircraft is described in detail, and a similar one for electric-powered aircraft is described to the extent that it differs from that for fueled aircraft. Typical values of technology-related parameters used in sizing calculations are given, followed by an example of a small UAV designed using the proposed methodology.

► **Systematic Design Methodology and Construction of Micro Aerial Quadrotor Vehicles** by Phang, Li, Chen, and Lee presents guidelines to optimally and systematically design and construct an ultralight micro aerial quadrotor vehicle, including details of the steps to design a stable quadrotor with not more than 50 g take-off weight and flight duration of 8 min. The methodology covers, sequentially, structural analysis of the micro quadrotor air frame, design in a 3-D virtual simulator, design of the avionic system, controller implementation via simulation, and component fabrication and implementation. Finally, performance is evaluated, tested, and confirmed with real flight missions.

► **Dexterous UAVs for Precision Low-Altitude Flight** by Jiang and Voyles follows the concept of force closure (a term from the dexterous manipulation community) to introduce a new type of dexterous six degrees-of-freedom (6 DOF) UAV, which provides the unique capability of being able to resist any applied wrench, or generalized force-torque, providing more precise control during low-altitude flight. The major challenge that needs to be overcome in such designs is that low-altitude flight usually introduces ground-effect disturbances and other backwash issues, while in the new field of aerial mobile manipulation it often includes close to structures operations for inspection or manipulation purposes. The presented design differs considerably from typical helicopters or quadrotors, which cannot instantaneously resist or apply an arbitrary force in the plane perpendicular to the rotor axis, thus, making such designs inadequate for complex mobile manipulation tasks.

Collectively, the five chapters provide a plethora of ideas and design principles that prepare the reader to capture more advanced technical challenges.

Computational and Experimental Design of a Fixed-Wing UAV

8

Helen Ryaciotaki-Boussalis and Darrell Guillaume

Contents

| | | |
|-------|--|-----|
| 8.1 | Odyssey UAV Mission | 110 |
| 8.1.1 | Introduction | 110 |
| 8.1.2 | Objective of the Design and Development of the UAV | 111 |
| 8.1.3 | Mission Requirements of the UAV | 111 |
| 8.2 | Odyssey UAV Design Specifications | 112 |
| 8.2.1 | Initial Sizing | 112 |
| 8.2.2 | Aerodynamic Design Process | 114 |
| 8.2.3 | Geometry and Configuration | 115 |
| 8.2.4 | Airfoil Selection | 115 |
| 8.2.5 | Structural Requirements | 120 |
| 8.2.6 | Engine Selection | 120 |
| 8.3 | Odyssey UAV Software Analysis | 122 |
| 8.3.1 | Structural Analysis: Femap | 122 |
| 8.3.2 | Composite Stress Analysis | 122 |
| 8.3.3 | Modeling | 123 |
| 8.3.4 | Deflections | 125 |
| 8.3.5 | Normal Mode Shapes and Mode Frequencies | 125 |
| 8.3.6 | Stress and Strain | 126 |
| 8.3.7 | Computational Fluid Dynamics | 126 |
| 8.4 | FOSS System Application to the Odyssey UAV | 127 |
| 8.4.1 | Introduction to Structural Health Monitoring | 127 |
| 8.4.2 | Computational Model | 130 |
| 8.4.3 | Sensitivity Analysis | 132 |

H. Ryaciotaki-Boussalis (✉)

Department of Electrical and Computer Engineering, NASA University Research Center
(SPACE), California State University, Los Angeles, CA, USA

e-mail: hboussa@calstatela.edu; hboussa@exchange.calstatela.edu

D. Guillaume

Department of Mechanical Engineering, NASA University Research Center (SPACE),
California State University, Los Angeles, CA, USA

e-mail: dguilla@exchange.calstatela.edu

| | | |
|------------|---|-----|
| 8.4.4 | Location of Fiber Line and Effects of Loading | 132 |
| 8.4.5 | Number of FBG Sensors | 134 |
| 8.4.6 | Sensor Integration and Loading | 134 |
| 8.4.7 | Analysis and Results | 136 |
| 8.5 | Flight Simulation | 136 |
| 8.5.1 | X-Plane Simulation Model | 136 |
| 8.5.2 | Autonomous Design | 137 |
| References | | 140 |

Abstract

This chapter presents a detailed example for the design and manufacture of all aspects of a prototype unmanned aerial vehicle (UAV). It further allows for the utilization of multi-segmented flight control surfaces on the wings of these UAVs. Adding multiple segments to UAV wings creates smaller control surfaces. By introducing smaller control surfaces, a wing can make refined adjustments to UAV performance while airborne. This unique technique will (i) apply localized correcting forces to the UAV, (ii) reduce structural deformation, (iii) minimize drag contribution due to control surface actuation, (iv) suppress and control structural resonance due to lift forces and vibrational modes, (v) reduce the weight of the structure, and (vi) improve the endurance of flights.

8.1 Odyssey UAV Mission

8.1.1 Introduction

In recent years, UAVs have played a major role in the security of the nation and have gained popularity over traditional full-size piloted aircrafts. Further, the use of UAVs has become a favored solution for important tasks requiring air operations. Examples of areas for air operations include academic research, earth environmental studies, fire surveillance, and reconnaissance.

This chapter describes the thought process and techniques required to design and develop a fixed-wing UAV. It presents a case study for the design of a particular UAV underway at the Structures, Propulsion, And Control Engineering (SPACE) – a NASA-sponsored University Research Center (URC) of Excellence at the California State University, Los Angeles (CSULA).

The main objective was to develop a multi-mission UAV, named “Odyssey,” which could be adapted to different mission requirements. The Odyssey UAV (Fig. 8.1) incorporates a flight control system (FCS) into a twin boom aircraft configuration, which is designed to be capable of carrying a payload of up to 14 lb, remaining in flight for up to 3 h and operating at a maximum altitude of 10,000 ft above sea level. The design encompasses the major areas of aircraft design, which include aerodynamics, structures, avionics, and propulsion.

Three scale prototypes were constructed and flown by the SPACE Center team as part of the concept verification, allowing the team to experiment and learn from a



Fig. 8.1 Odyssey UAV

number of manufacturing techniques prior to the construction of the Odyssey UAV. This UAV was also used to develop and apply novel techniques for real-time structural health monitoring (SHM). The integration of a system for SHM of an aircraft allows for reduction of weight while maintaining a high level of confidence in UAV design. These SHM data are used for fault detection, while a control system is used to assure system reconfiguration.

8.1.2 Objective of the Design and Development of the UAV

During the previous years, the NASA University Research Center of Excellence (URC) has concentrated gaining the experience, skills, and manufacturing techniques for the development of small, lightweight UAVs that have excelled in the area of endurance. The Odyssey UAV project was focused on the design of a multi-mission/multipurpose air system that can operate autonomously with a maximum payload carrying capacity of 14 lb.

8.1.3 Mission Requirements of the UAV

The mission requirements were selected for the design of a multipurpose UAV that is capable of wide variety of missions with minimal structural modifications. The following are the specific requirements that have been used to design the Odyssey UAV:

Endurance: 3 h

Payload: 14 lb maximum

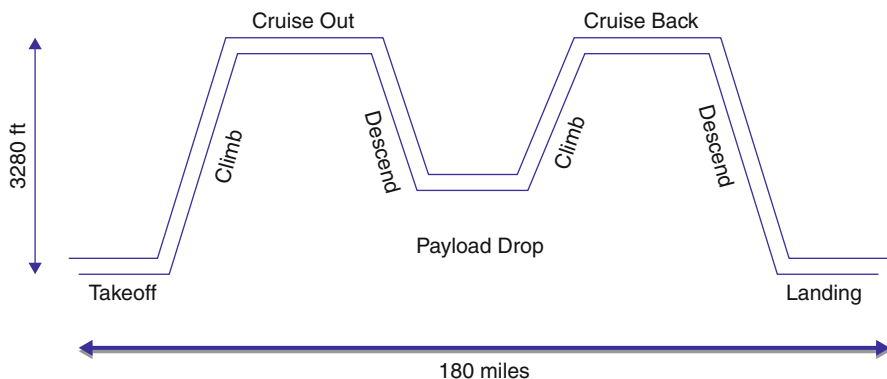


Fig. 8.2 Mission profile

Cruise speed: 81.4 ft/s

Cruising altitude: 3,000 ft (above sea level)

Max. altitude: 10,000 ft (above sea level)

The endurance capabilities of the air-frame were a trade-off between the payload weight and the mission. The mission profile shown in Fig. 8.2 was adopted for the development of the Odyssey UAV. However, the final mission will be determined by NASA once the UAV is developed.

As shown in Fig. 8.2, the UAV was designed to take off within a distance of about 300 ft, climb, reach cruise altitude, and maintain a speed of 81.4 ft/s. The UAV will then cruise for 3 h, covering 180 mi, and then finally will decent to a predetermined way point for a payload drop.

8.2 Odyssey UAV Design Specifications

8.2.1 Initial Sizing

The first step in designing Odyssey was to perform a feasibility study by identifying existing UAVs with similar geometry, size, and mission requirements. Using available data from existing UAVs, a parametric study was performed to determine an initial wingspan, power requirement, and UAV maximum speed. The following table shows the UAVs and corresponding parameters that were considered for this study (Table 8.1).

Figures 8.3–8.5 graphically show a comparison of the existing UAV's weight versus cruise speed, weight versus wingspan, and weight versus horsepower.

These figures were used to determine the first estimate of the UAV sizing including (1) gross weight, (2) wingspan, and (3) engine size. Specifically, the UAV gross weight was estimated to be between 100 and 150 lb, the wingspan estimated to be between 11 and 16 ft and the engine power estimated to be of at least 7 HP.

Table 8.1 UAV parametric study

| UAV | Weight (lb) | Wingspan (ft) | Cruise speed (mi/h) | HP | Ceiling (ft) |
|-------------|-------------|---------------|---------------------|------|--------------|
| Aerosonde | 28.8 | 5.6 | N/A | 1.7 | |
| ScanEagle | 39.6 | 10.2 | 82 | 1.3 | 16,400 |
| Insight | 44 | 10.2 | 82 | | 19,496.3 |
| Integrator | 129.8 | 15.7 | 98.4 | 8 | 19,680 |
| Aerolight | 88 | 13.1 | 84.7 | | 9,997.4 |
| Orbiter | 14.3 | 7.2 | 51 | | 17,994.1 |
| Silver Fox | 26.8 | 7.9 | 63.8 | 1.2 | 11,995 |
| Manta | 51.7 | 8.8 | 65.6 | | 15,973.6 |
| Coyote | 14.1 | 4.8 | 92.9 | | 19,906.3 |
| Killerbee | 30.8 | 10 | | 11 | 15,088 |
| Tern | 129.8 | 11.4 | | 12 | 9,997.4 |
| Mako | 129.8 | 12.8 | | 12 | 9,997.4 |
| Gull 24 | 39.6 | 8.9 | | 4 | |
| Skylark II | 77 | 13.8 | | 5.4 | |
| Neptune | 129.8 | 7 | 101.1 | 15 | 7,996.6 |
| Viking 100 | 149.6 | 12 | | 16 | |
| Gull 44 | 204.6 | 16.4 | | 19 | |
| Luna x 2000 | 81.4 | 13.7 | 63.8 | 6.7 | 13,120 |
| D1 | 74.8 | 10.8 | 86.5 | 3.5 | 4,998.7 |
| Shadow 200 | 169.4 | 12.1 | | 38.2 | 14,996.2 |

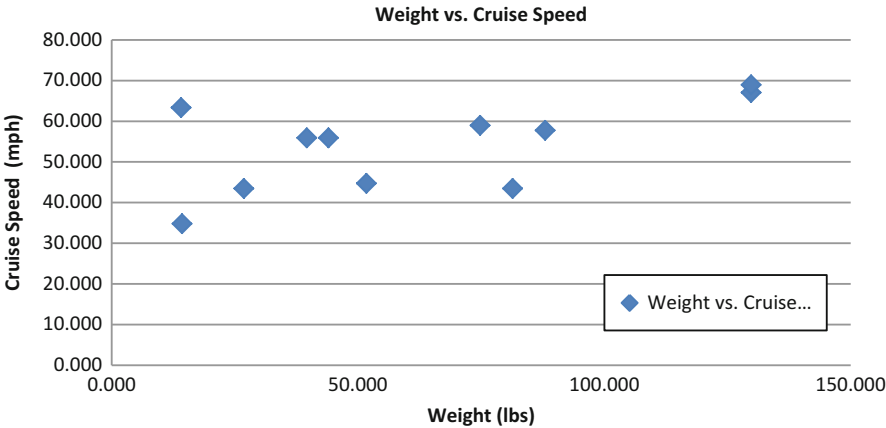


Fig. 8.3 Weight versus cruise speed

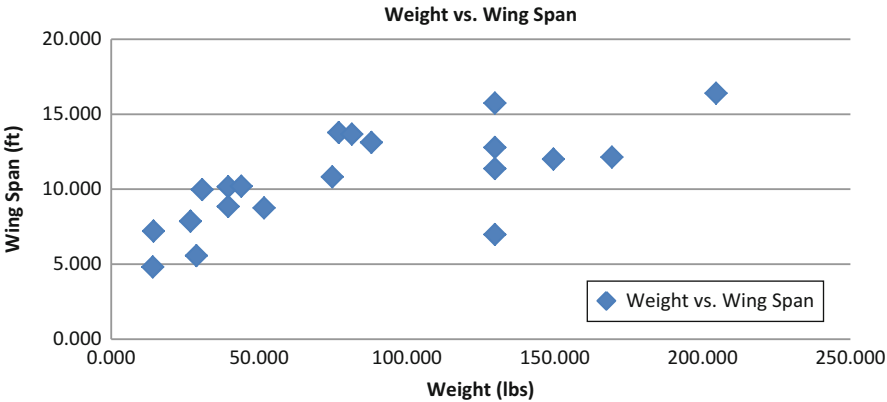


Fig. 8.4 Weight versus wing span

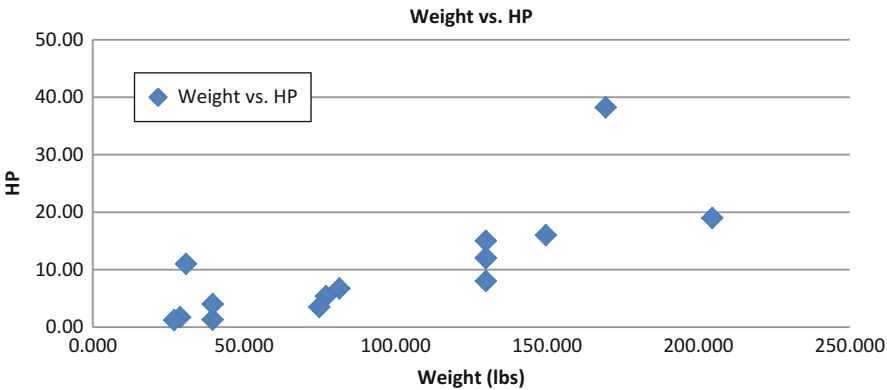


Fig. 8.5 Weight versus horsepower

8.2.2 Aerodynamic Design Process

The traditional aerodynamic analytical relationships were utilized for developing the aerodynamic UAV model by the following steps: Step 1: Make selections such as the airfoils used for the main wing and tail, the configuration of the plane such as pusher or puller, the span, location of wings relative to each other and chord length. Step 2: Perform analysis on the geometry of the airfoils selected for the aircraft. Once the geometry of the aircraft is selected, it is possible to estimate lift-to-drag ratios, static margin, and performance capabilities. The degree of refinement increases as the number of finalized parameters is increased. The increase in refinement begins to reveal the cost of construction and the types of manufacturing techniques that will be required. This is the approach that was used to design and manufacture the Odyssey UAV. Step 3: Use the Athena Vortex Lattice (AVL) computational fluid dynamics software package to model the aerodynamic characteristics of the proposed UAV

structure. Step 4: Compare chosen aerodynamic characteristics to actual flight data. These aerodynamic characteristics should provide baseline aerodynamic parameters for state-space model to be used in the development of an autonomous controller. Following the steps mentioned above allows for the completion of the UAV design.

For UAV design, a decision must be made between increased maneuverability and stability of the aircraft. For the aerodynamic design of the Odyssey UAV, priority was given to an increase in stability and payload rather than an increase in maneuverability. To achieve this, a twin boom pusher assembly with a rectangular wing was selected. Since the payload of the Odyssey UAV includes a forward-facing camera, the pusher configuration also facilitates undisturbed imaging.

8.2.3 Geometry and Configuration

The Odyssey UAV design has a rectangular main wing that has a 15 ft wingspan, a mean aerodynamic chord of 1.75 ft, and an aspect ratio of 8.57. In order to simplify the design and analysis, the main wing does not change geometry along its span nor is it swept. The aircraft is estimated to have a gross weight of 140 lb. The airfoil selected for the main wing is an Eppler214, and the airfoil for the horizontal tail and vertical stabilizers is NACA0012. These airfoil types were selected so that the UAV will exhibit a high lift at low speeds and provide adequate moments for stable pitch. Other aircraft specifications are shown in Tables 8.2–8.4 below.

The control surface sizing was based on the results of the stability studies conducted in AVL. Based on the results, the sizes of the control surfaces for the aircraft are shown in Table 8.5.

Figure 8.6 shows a representation of the UAV main geometry which includes the main wing and the horizontal and vertical tails. The yellow areas represent the control surfaces.

8.2.4 Airfoil Selection

This section describes the process in determining final geometry of the UAV based on CFD analysis and empirical calculations. As mentioned in the previous section, the airfoil types were selected so that the UAV will exhibit a high lift at low speeds and provide adequate moments for stable pitch.

Several experimental studies were performed in order to identify the final airfoil selection for the wing and horizontal tail/vertical tails. The airfoil types for these wing and tail combinations included in these studies were the Eppler214 and NACA0012, Clark Y and NACA2412, and Eppler214 and NACA2412. The Eppler12 and Clark Y were chosen for the main wing, while the NACA0012 and NACA2412 were chosen for the horizontal and vertical tails.

Since the interaction of the wing and tail is vital for the performance of the aircraft, the airfoil selection was based on the following design criteria:

- Lift/drag (L/D) must be greater than 15.

- Good trim conditions (which implies good pitch stability).

Table 8.2 Wing specifications

| | |
|---------------------------|--------------------------------|
| Wing (Eppler214 airfoil) | |
| Weight | $W = 140 \text{ lb}$ |
| Wing loading | $W/S = 5.33 \text{ lb/ft}^2$ |
| Aspect ratio | $AR = 8.57$ |
| Taper ratio | $C_t/C_r = 1$ |
| Wing area | $S = 26.25 \text{ ft}^2$ |
| Wingspan | $b = 15$ |
| Root chord | $C_r = 1.75 \text{ ft}$ |
| Tip chord | $C_t = 1.75 \text{ ft}$ |
| Average chord length | $C = 1.75 \text{ ft}$ |
| Quarter chord sweep angle | $L = 0^\circ$ |
| Leading-edge sweep | $L = 0^\circ$ |
| Span efficiency | $e = 0.505$ |
| Thickness ratio | $t/c = 0.137$ |
| Total surface area | $S_{surf} = 67.5 \text{ ft}^2$ |

Table 8.3 Vertical stabilizer specifications

| | |
|--|------------------------------|
| Horizontal stabilizer (NACA0012 airfoil) | |
| Span | $b = 6 \text{ ft}$ |
| Root chord | $C_r = 1 \text{ ft}$ |
| Tip chord | $C_t = 1 \text{ ft}$ |
| Planform area | $S = 6 \text{ ft}^2$ |
| Taper ratio | $l = 1$ |
| Aspect ratio | $AR = 6$ |
| Thickness ratio | $t/c = 0.12$ |
| Sweep angle | $L = 0^\circ$ |
| Total surface area | $S_{surf} = 12 \text{ ft}^2$ |

Table 8.4 Horizontal stabilizer specifications

| | |
|--|---------------------------------|
| Vertical stabilizer (NACA0012 Airfoil) | |
| Span | $b = 2.25 \text{ ft}$ |
| Root chord | $C_r = 1 \text{ ft}$ |
| Tip chord | $C_t = 0.675 \text{ ft}$ |
| Ave. chord | $C_a = 0.8375$ |
| Planform area | $S = 0.675 \text{ ft}^2$ |
| Taper ratio | $l = 0.675$ |
| Aspect ratio | $AR = 7.500$ |
| Sweep angle (LE) | $L = 15.52^\circ$ |
| Total surface area | $S_{surf} = 4.500 \text{ ft}^2$ |

Table 8.5 Control surface specification

| | Flaps | Aileron | Elevator | Rudder |
|-------------|-------|---------|----------|--------|
| Chord (in.) | 6.825 | 6.825 | 3.6 | 3.9 |
| Span (in.) | 30 | 20 | 64.8 | 27 |

Fig. 8.6 Representation of UAV geometry and control surfaces

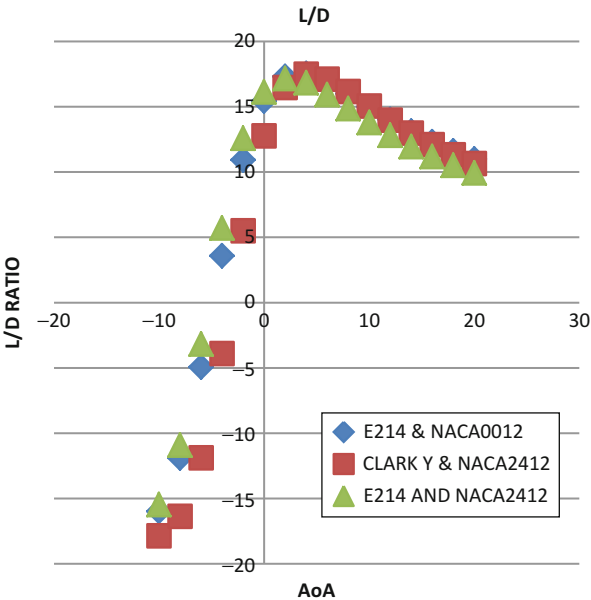
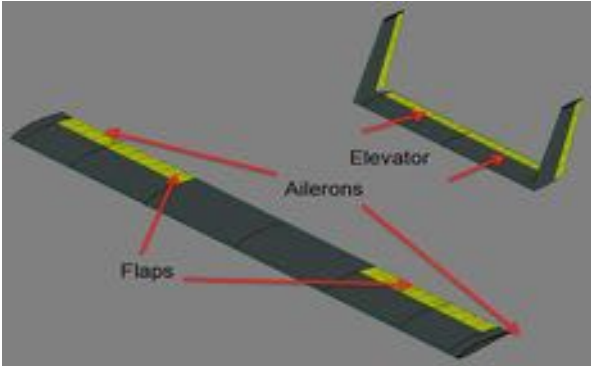


Fig. 8.7 Angle of attack versus lift-to-drag ratio

Figure 8.7 shows the different lift-to-drag ratios for the selected airfoil combinations, and all performed with an L/D ratio greater than 15. Figure 8.8 shows the pitch stability of the aircraft with the different airfoil combinations. It can be seen that Clark Y and NACA2412 and E325 and NACA2012 have negative pitch stability. Since the objective of the Odyssey UAV is to be naturally stable during steady flight, this negative pitch stability is not desired. The selected airfoil combination that provided the most desired performance for the Odyssey UAV was the NACA00012 for the main wing and the Eppler214 for the tails.

The final set of experiments for this study was to evaluate the drag performance of the different airfoil combinations. Figure 8.9 demonstrates that the Eppler214 and NACA0012 combination has the lowest drag.

Fig. 8.8 Pitch stability

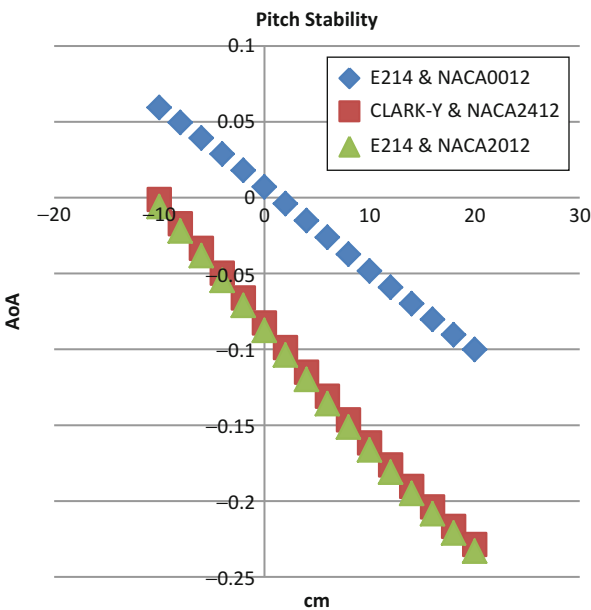
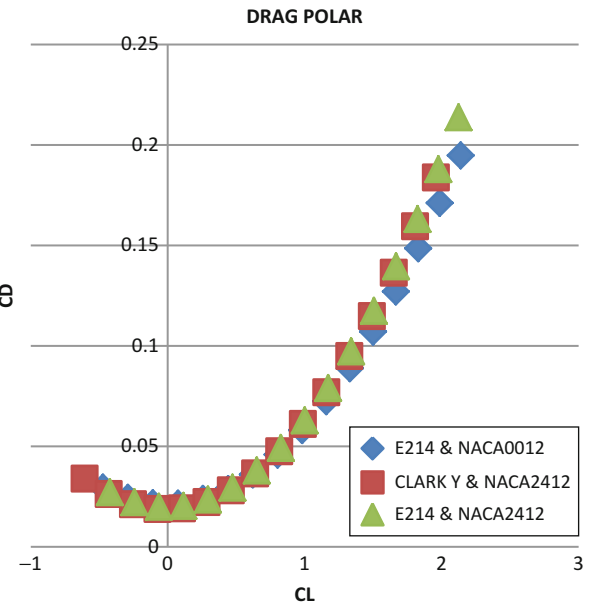


Fig. 8.9 Drag polar



Based on the analysis that was performed, the optimal airfoil combination selected for the Odyssey UAV was the Eppler214 and NACA0012 since this combination provides (1) optimal L/D, (2) the lowest drag polar, and (3) the most stable aircraft performance. The following Table 8.6 shows the aerodynamic data obtained from both empirical studies and CFD analysis.

Table 8.6 Aerodynamic coefficients from empirical studies

| | | | |
|-------------------------------------|---------------|--------|----------------------|
| Skin friction coefficient | C_{fe} | 0.0055 | |
| Zero lift parasite drag coefficient | $C_{D,O} =$ | 0.0184 | |
| Span efficiency | $e =$ | 0.5 | |
| Induced drag factors | $k_3 =$ | 0.0735 | |
| | $k_2 =$ | 0 | |
| | $k_1 =$ | 0.0245 | |
| | $K =$ | 0.0980 | |
| Air density (sea level) | $p =$ | 0.0024 | slug/ft ³ |
| Total engine power | $P =$ | 9.80 | HP |
| Prop efficiency | $H_{pr} =$ | 0.70 | |
| Total power available | $P =$ | 6.86 | HP |
| Speed | $V =$ | 81.40 | ft/s |
| Max curve slope | $Cl_{max} =$ | 1.06 | |
| Stall speed | $V_{stall} =$ | 57.20 | ft/s |
| Maximum lift-to-drag ratio | $L/D =$ | 18.8 | |

These coefficients are determined from both analytical studies and CFD analysis using AVL. The AVL software uses the vortex lattice method to estimate the lift and drag forces on the modeled aircraft. This software package was used during the design of the Odyssey UAV to ensure that the design provided adequate lift for both the UAV structure and the 14 lb payload. The software was also used to determine aircraft stability and to provide other necessary aerodynamic parameters.

The CFD study also determined that the wing and the horizontal tail must have an incidence angle of 2° . The necessary angles of attack for the main wing were found by optimizing the lift-to-drag ratios at cruising conditions. In addition to this, AVL was used to perform studies to investigate the overall performance of the aircraft at different speeds and altitudes. Optimizing an aircraft design minimizes fuel consumption, minimizes drag, and reduces the loads that the structure experiences.

By applying CFD to an aircraft design, performance of the UAV design can be simulated. The simulated performance can be used to determine if off-the-shelf components, such as an engine, can be used while maintaining design parameters of the aircraft. If off-the-shelf parts can be utilized rather than custom parts, both time and cost of manufacturing are reduced. For the Odyssey, the simulation showed that an off-the-shelf engine met the aircraft requirements and a Desert Aircraft model DA-120 engine was purchased.

Another important parameter in aircraft design is the static margin. This is the distance between the neutral point (i.e., the entire aircraft's aerodynamic center) and the center of gravity divided by the length of the chord. The static margin is used in determining the longitudinal stability of an aircraft. A positive static margin on an aircraft means that the center of gravity is in front of the neutral point, while a negative static margin on an aircraft means that the center of gravity is behind the neutral point. As the static margin increases, the longitudinally

stability increases. However, as the static margin increases, maneuverability decreases. As stated previously, the stability of the Odyssey UAV was determined to be more important than its maneuverability, so this UAV was designed with a static margin of 15 %.

8.2.5 Structural Requirements

The structural design of a UAV is based on the aerodynamic loads generated during each stage of the mission which include takeoff, climb, cruise, diving, and landing. These loads are determined from CFD and form the flight envelope of the Odyssey UAV as seen below in Fig. 8.10.

Based on the maneuverability requirements of the Odyssey UAV design, the maximum acceleration loading was set to 3.0 g. This is the largest acceleration that Odyssey UAV will experience in the flight envelope. Thus, the aircraft structure was designed to meet the 3.0 g loading condition while maintaining a factor of safety of 2.0. Initial weight estimates obtained from a finite element model (FEM) indicated that an aircraft structure designed to meet these loads would weigh approximately 140 lb.

8.2.6 Engine Selection

Once the aerodynamic analysis is complete, the propulsion system can be selected using the following considerations: (1) the size of the propulsion system must be able to meet the minimum thrust requirements necessary to overcome the drag of the aircraft and achieve lift with an estimated maximum headwind; (2) the type of propulsion system used for an aircraft, such as piston engines, turboprop, and turbofan, turbojet, will influence the performance characteristics of the aircraft; (3) the amount of thrust output by the engine compared to the weight of the engine needs to be maximized; and (4) the thrust-specific fuel consumption (TSFC) which compares the ratio of the rate of fuel consumed to the thrust produced needs to be minimized. The temperature air surrounding the aircraft affects the performance of the aircraft since decreased air density leads to decreased lift and thrust. An aircraft designed with high maneuverability and high-speed requirements will require high thrust requirements (low engine efficiency), while an aircraft designed with simple cruise requirements will have low fuel consumption (high engine efficiency). For UAV propulsion systems designed for a specific mission, trade-offs must be made between thrust and efficiency.

Piston engines are internal combustion engines that convert the reciprocating motion of pistons to rotational motion to spin a propeller. These engines are ideal for cruising speeds since they are built to operate at subsonic speeds and provide low thrust-specific fuel consumption. A propeller-driven engine has low maneuverability and cannot produce as much thrust as newer jet engines, but the low TSFC for

Fig. 8.10 V-n diagram:
operational

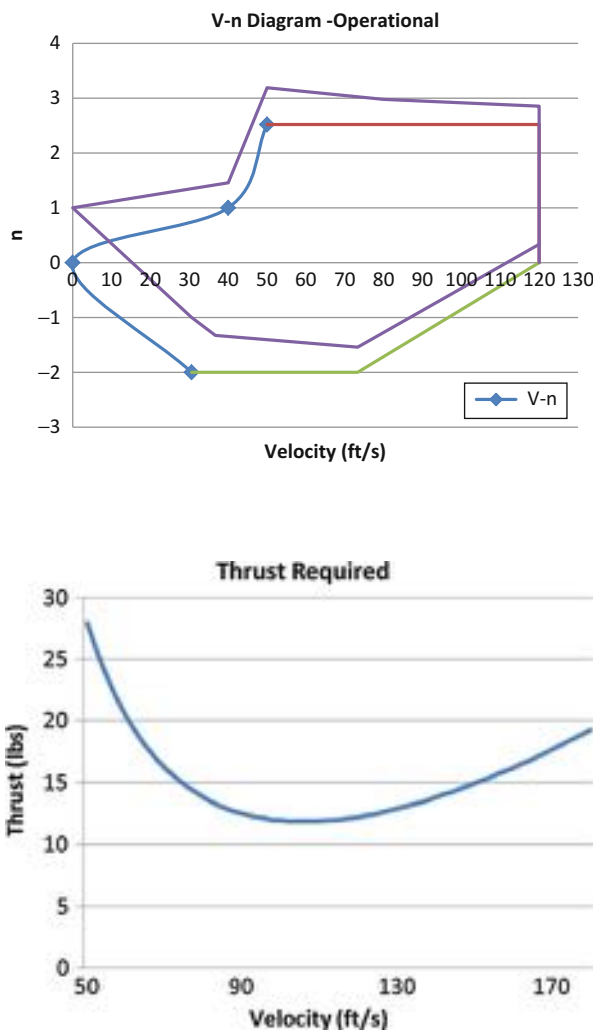
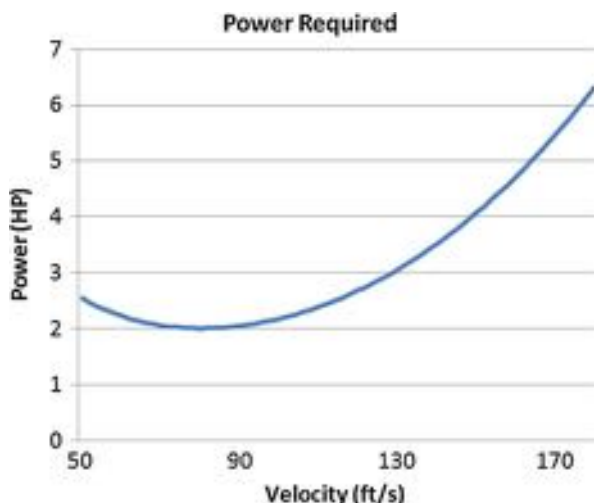


Fig. 8.11 Thrust required

propeller-driven engines is ideal for endurance missions. Alternatively, airbreathing jet engines, such as turbojet or turbofan, are another type of internal combustion engine which turbines instead of pistons to compress air. These airbreathing engines are capable of producing high thrust but are inefficient and heavy.

Since both stability of the aircraft and flight endurance were the main mission requirements of the Odyssey UAV, a piston-type engine was chosen. Figure 8.11 shows the thrust requirements of the piston engine at various speeds, while Fig. 8.12 shows the amount of power required at various speeds. The fuel tank for Odyssey UAV is custom made, and it is capable to hold a fuel capacity that would allow for a maximum 3 h of flight time.

Fig. 8.12 Power required

8.3 Odyssey UAV Software Analysis

8.3.1 Structural Analysis: Femap

A Nastran FEA model was developed to determine the allowable stresses within the aircraft skin and internal structures. For simplicity, only structural elements of the aircraft are modeled. All panels were modeled using 2-D orthotropic materials and laminate properties to determine membrane forces, bending moments, and transverse shear loads. The fuselage and wing skins are composed of sandwich panels made from composite materials. The material properties of the face materials were then defined within the FEM using core material properties specified by the manufacturer. Geometry for the finite element model was imported from Solid Works, and the imported surfaces were meshed with the laminate properties to simulate the sandwich panels.

8.3.2 Composite Stress Analysis

The structure's margin of safety was determined utilizing the Tsai-Hill Failure Index (FI) to predict the failure in composite materials. The FI is defined as

$$FI = \frac{\sigma_1^2}{X^2} - \left(\frac{1}{X^2} + \frac{1}{Y^2} - \frac{1}{Z^2} \right) \sigma_1 \sigma_2 + \frac{\sigma_2^2}{Y^2} + \frac{\tau_{12}^2}{S^2}$$

where

σ_1 = calculated plane stress of the sandwich panel in direction 1

σ_2 = calculated plane stress of the sandwich panel in direction 2

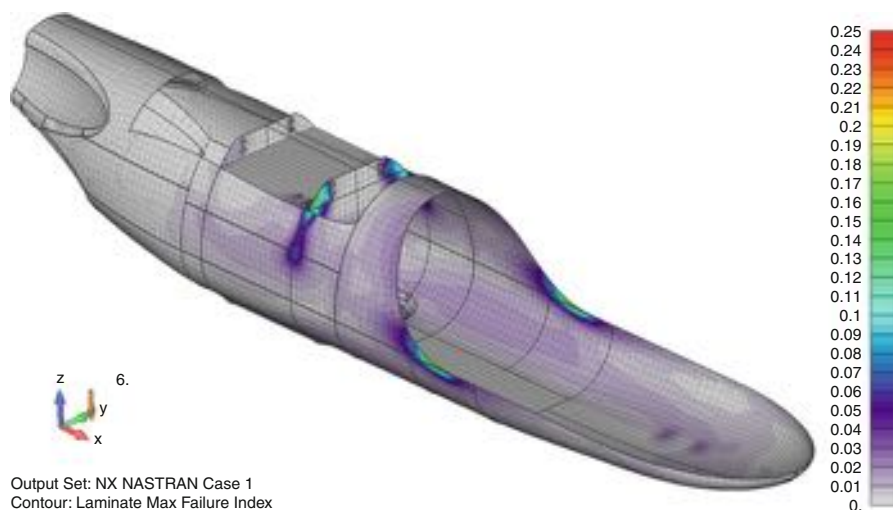


Fig. 8.13 Fuselage stress contours (Tsai-Wu failure index)

τ_{12} = calculated in-plane shear stress of the sandwich panel

X = normal allowable stress of the sandwich panel in the X direction

Y = normal allowable stress of the sandwich panel in the Y direction

S = in-plane shear allowable of the panel

8.3.3 Modeling

All internal loads were applied using mass elements when the laminate properties of the fuselage skin, bulkheads, and internal components were modeled. All mass elements are attached to the structure using interpolation elements (rigid RBE3) in the FEM model. The function of the RBE3 is to attach the force or mass of a single dependent node to the average independent node and will not increase the stiffness of the model. RBE3 can also be considered as a “force link,” which distributes the forces of the independent nodes to the dependent nodes by interpolation. The fuselage structure load was applied by specifying the aircraft accelerations in the FEA model as seen in Fig. 8.13. An engine thrust load of 60 lb was also applied to the engine assembly mass element, and the fuselage was constrained at the interface between the bulkheads and the main spar of the wing.

The structure of the wing-boom-tail assembly and all internal ribs were modeled with the use of laminate properties. For the wing-boom-tail assembly (Fig. 8.14), all lift loads were applied using nodal loads. A distributed lift load, which decreased toward the wing and tail tips, was calculated by incorporating a maximum gust load of 3.0 g and the aircraft accelerations obtained from the FEA model. The moments

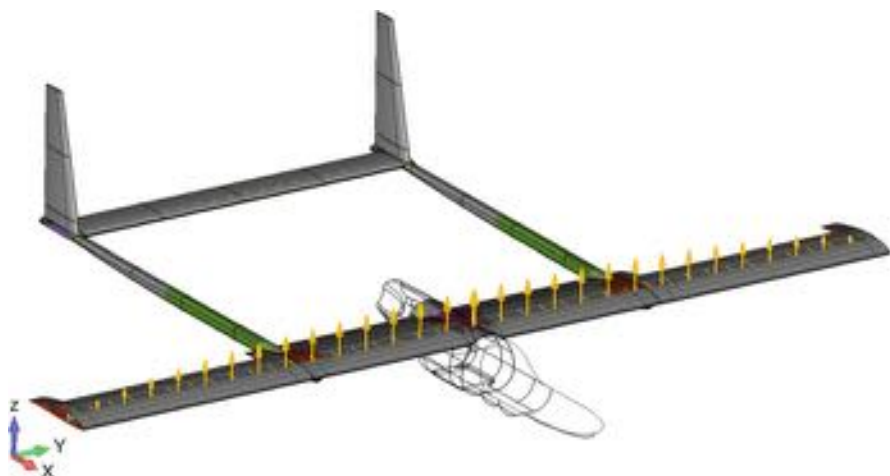


Fig. 8.14 EA analysis of wing structure

Fig. 8.15 Composite wing structure



about the quarter chord of the main wing were also applied to the model by using nodal loads. The wing is constrained at the interface between the bulkheads of the fuselage and the main spar of the wing.

Figure 8.15 shows the main load-bearing structure within the wing which is a composite spar that has a box beam configuration.

The spar is composed of alternating layers of both bidirectional and unidirectional carbon fiber. The layers of unidirectional carbon fiber are orientated in the spanwise direction to maximize the tensile and compressive strength of the carbon fiber in bending. The bidirectional layers of carbon fiber are orientated in a 45° angle with respect to the spanwise direction to increase the strength of the spar under torsional loading. Regions where the spar experiences the least bending moments are manufactured with fewer layers of carbon fiber. The loading transferred from the tail through the booms add additional local stresses. To alleviate large stress concentrations, the spar and local sandwich panels were reinforced with additional layers of carbon fiber (Fig. 8.16).

The base main wing was optimized and reinforced by increasing the number of layers of carbon fiber to counteract the large bending moments experience near the root chord. As the bending moment decreases with distance from the root, the number of layers of carbon fiber is decreased to prevent the risk of peak stress concentrations. Based on previous structural analysis, additional reinforcement was required in regions where the central wing, the wing tip extensions, and the booms connect.

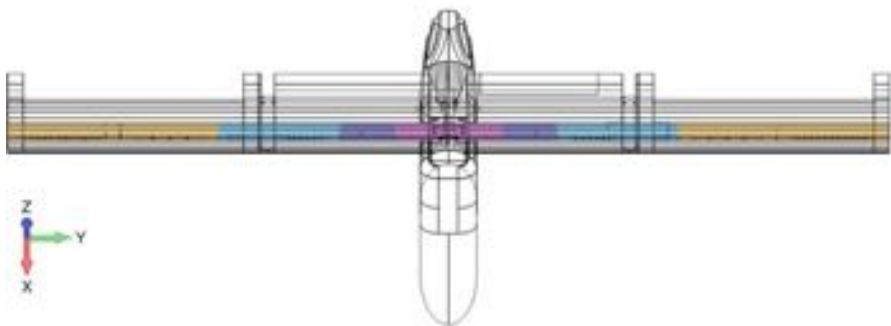


Fig. 8.16 Carbon fiber layers for spar



Fig. 8.17 Simulated deflections caused by in-flight loads

8.3.4 Deflections

The results of the FEM show that the aircraft in a 3.0 g maneuver is capable of withstanding an ultimate aerodynamic load of 720 lb and can have expected wing tip deflections up to 1.69 in. (Fig. 8.17).

8.3.5 Normal Mode Shapes and Mode Frequencies

A structure modal analysis must be performed since uncontrolled vibrations can be catastrophic for the aircraft. Based on the finite element modeling shown in Fig. 8.18, the first structural bending mode frequency of the Odyssey UAV main wing was determined to be 8.25 Hz.

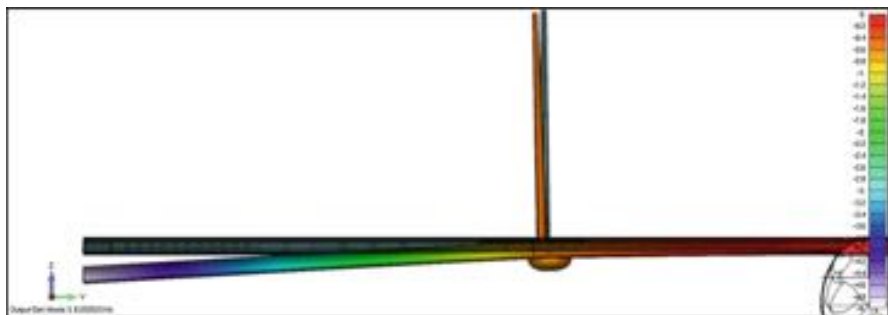


Fig. 8.18 Analysis of the first structural bending mode frequency

8.3.6 Stress and Strain

The FEM of the Odyssey UAV wing-boom-tail was used to predict the stress and strain fields in the structure while under load. The greatest stress concentration occurs near the root of the wing, while the peak stress concentration occurs near where the fuselage attaches to the wing (Fig. 8.19).

The direction of composite material needs to be specified during manufacturing. The skin material used in the Odyssey UAV is a bidirection carbon fiber weave which has equal stiffness and strength in two directions. However, the spar is composed of unidirectional carbon fiber, which is stiffer and stronger in the tow direction. FEM allows the user to specify the tow direction of the composite material. Since the largest contribution to stress on the wing structure is due to bending, it is critical to analyze the strain contours in the wingspan direction (Fig. 8.20).

8.3.7 Computational Fluid Dynamics

The Athena Vortex Lattice was used as the CFD software to calculate the aerodynamic parameters and the flight dynamics of the Odyssey UAV. It is a free resource software package that is an effective tool to approximate and refine the flight-dynamic characteristics of a custom-built rigid-body UAV.

The two major approximations made by the Athena Vortex Lattice include the following: (1) the software treated the fuselage as a slender body and did not consider certain drag features (booms and the main landing gear) of the Odyssey UAV model, and (2) the software used an average skin friction drag coefficient. The actual skin friction drag coefficient requires extensive wind tunnel testing, and it typically makes a small contribution to the overall drag coefficient of the aircraft. Regardless of these approximations, the software can provide a reasonable aerodynamic model that can be refined from flight testing.

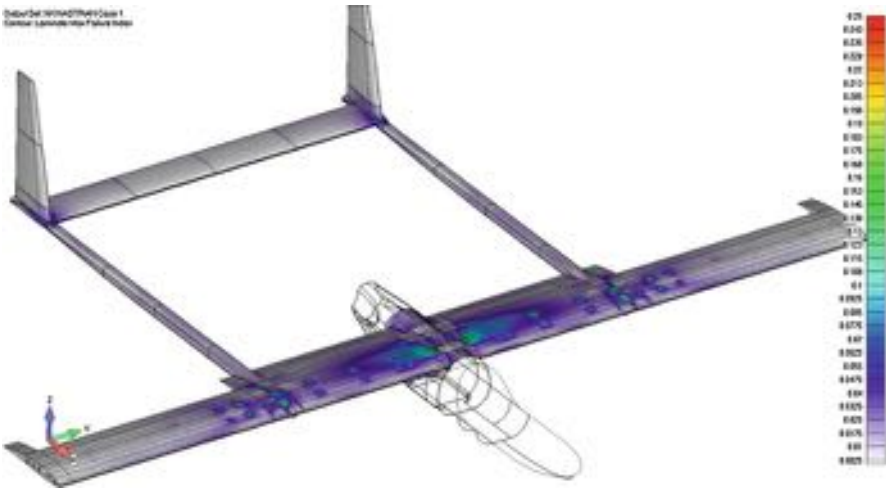


Fig. 8.19 Wing assembly stress contours (Tsai-Wu failure index)

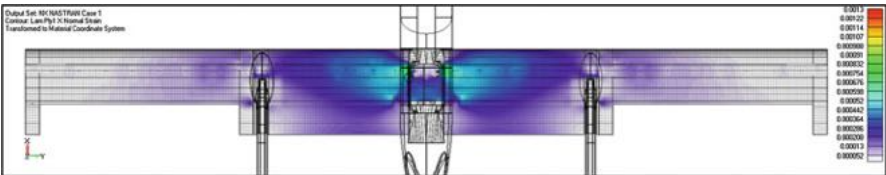


Fig. 8.20 Strain contour in the span direction

Despite of its limitations, AVL is still a powerful tool in regard to designing and building custom aircraft and was crucial to the development of the Odyssey. The software can provide lift loading at varying angles of attack, flight stability characteristics, dynamic system matrices, and other parameters necessary to design, manufacture, and control a UAV (Fig. 8.21).

8.4 FOSS System Application to the Odyssey UAV

8.4.1 Introduction to Structural Health Monitoring

The integration of a system for structural health monitoring of an aircraft allows for reduction of weight while maintaining a high level of confidence in UAV design. In addition, a real-time SHM system with a novel robotic/UAV-to-human interface will reduce the risk of in-flight breakups by providing crucial flight data to the ground control station (GCS), including structural deformations, stresses, and loading. These SHM data will also be used for fault detection, while the proposed control system algorithms will assure system reconfiguration. Traditional sensing

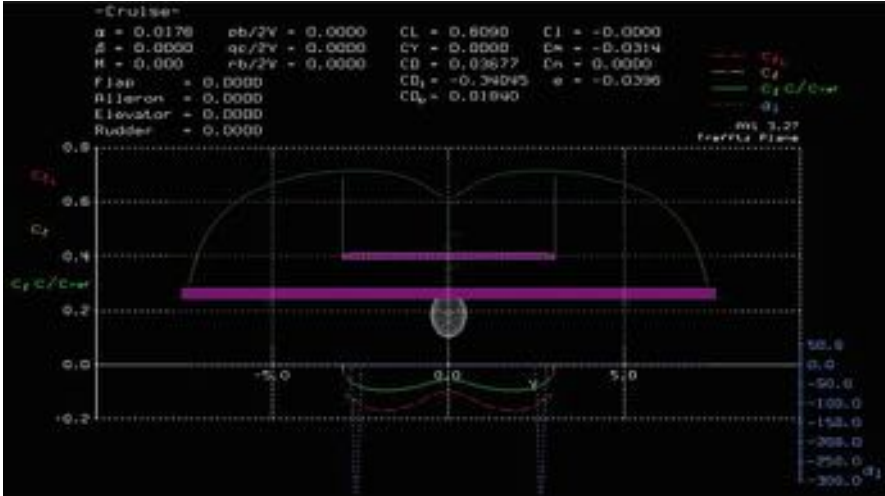


Fig. 8.21 Odyssey AVL model

instrumentation – including strain gauges, accelerometers, and thermocouples – tend to be bulky and heavy, limiting their application to a few sensors, usually near the wing root. Recent improvements in fiber optic strain sensing technology (FOSS) have enabled the use of embedded fiber Bragg gratings (FBGs), which can provide numerous distributed strain and temperature measurements from a variety of structural elements. A FBG is a photo-induced refractive plane that is etched onto a fiber-optic cable that reflects a low power signal when light pulses travel through it. As a FBG is stretched, undergoing strain, the spacing of the refractive plane changes as well. This change in the refractive plane spacing causes a change in the reflected signal's index of refractivity. The change in the index of refractivity is directly proportional to the strain felt by the fiber-optic line. By using a Bragg wavelength (λ_b) as a point of reference to track strain over time, the relationship for strain becomes

$$K\varepsilon = \frac{\lambda - \lambda_b}{\lambda_b}$$

where K is a proportionality constant based on the properties of the fiber-optic cable and λ is the wavelength being monitored by the FOSS system. Because of their accuracy, light weight, small size, and flexibility, these fiber-optic sensors are ideal for aircraft with strict weight and size limitations. Given this ability to provide a large number of sensor measurements, there is the potential to develop and implement methods of monitoring, which have a higher accuracy because of the increased sensor density. Thus, a higher accuracy of monitoring can be achieved by this large number of sensor measurements, which facilitates control and system reconfiguration. A strain-based method, known as the displacement transfer functions (DTF), has been developed by the NASA Dryden Flight Research

Center. The strain-based shape prediction algorithm, also known as the displacement transfer function, is derived from classical beam theory (Euler-Bernoulli) which relates the measured strain of the structure to the theoretical curvature, slope, deflection, and cross-sectional twist angle equations for uniform cantilever beams. The Euler-Bernoulli equation is the general differential equation that describes the relationship between load and deformation of a beam:

$$EI \frac{d^4 y(x)}{dx^4} = q(x)$$

where the variable x is the distance of the beam starting at the wing root and ending at the end of the beam, $y(x)$ is the displacement of a beam as a function of distance from the root, $q(x)$ is the loading function analogous to the lift generated by the wing, E the elastic modulus, and I is the second moment of area that is calculated with respect to the centroidal axis perpendicular to the applied loading. The moment-strain relationship of the classical beam differential equation is used to implement the shape prediction algorithm and states that the curvature, second derivative of displacement, of a beam is proportional to the applied bending moment:

$$\frac{d^2 y}{dx^2} = \frac{M(x)}{EI}$$

By relating the bending stresses to the definition of stress and strain in Hooke's law, it can be shown that the curvature of the beam is directly proportional to the strain felt by the beam during bending:

$$\sigma(x) = \frac{M(x)c}{I} \quad \& \sigma(x) = E\varepsilon \xrightarrow{\text{yields}} \frac{d^2 y(x)}{dx^2} = \frac{\varepsilon(x)}{c}$$

where c is the distance to the neutral axis of which the beam is bending.

By utilizing the high spatial resolution of the strain measurements, it is possible to numerically integrate the second-order differential by use of finite difference method, without accumulating a large amount of error. By integrating the curvature equation and applying boundary conditions, the slope equation of the n th strain location can be written in the general form:

$$\tan \theta_n = \frac{\Delta l}{2c} (\varepsilon_{n-1} + \varepsilon_n) + \tan \theta_{n-1} = \frac{\Delta l}{2c} \left[\varepsilon_0 + 2(\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \dots + \varepsilon_{n-1}) + \varepsilon_n \right]$$

$$\tan \theta_n = \frac{\Delta l}{2c} \left[\varepsilon_0 + 2 \sum_{j=1}^{n-1} \varepsilon_j + \varepsilon_n \right]$$

To obtain the displacement of the wing using the shape prediction algorithm, it is necessary to integrate the slope equation with the respect to the x -axis. Utilizing the high spatial resolution of the strain measurements, it is possible to numerically integrate the first-order differential by use of finite difference method, without

accumulating a large amount of error. The displacement of the beam at any given strain sensing station is approximated by

$$y_0 = 0 \quad \& \quad \tan \theta_0 = 0$$

$$y_1 = \frac{(\Delta l)^2}{6c} (2\varepsilon_0 + \varepsilon_1) + \Delta l \cdot \tan \theta_0 + y_0$$

$$y_2 = \frac{(\Delta l)^2}{6c} (2\varepsilon_1 + \varepsilon_2) + \Delta l \cdot \tan \theta_1 + y_1$$

After applying boundary conditions, the displacement equation of the n th strain location can be written in the general form:

$$y_n = \frac{(\Delta l)^2}{6c} \left[2\varepsilon_0 + 3(\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \dots + \varepsilon_{n-1}) + \varepsilon_n \right]$$

$$+ \Delta l (\tan \theta_1 + \tan \theta_2 + \tan \theta_3 + \dots + \tan \theta_{n-1})$$

$$y_n = \frac{(\Delta l)^2}{6c} \left[2\varepsilon_0 + 3 \sum_{i=1}^{n-1} \varepsilon_i + \varepsilon_n \right] + \Delta l \sum_{i=1}^{n-1} \tan \theta_i$$

This method coupled with the Dryden developed FOSS technology allows for real-time deformation shape prediction. The FOSS technology has shown great potential for applications on various other structures, including smaller-scale and fixed-wing UAVs, as well as space structures and is being used on the Odyssey UAV. Excessive deformations of these structures can potentially induce damage to the wing, which may lead to flight instabilities and even catastrophic in-flight breakups. For fixed wings, the FOSS system has the potential to allow for the prevention of aerodynamic flutter by monitoring resonance frequencies and by providing crucial status of the vehicle's overall structural integrity.

Currently, a computational model has been created and used to investigate and assess the viability of the deflection algorithm as a displacement-shape prediction tool for potential monitoring and control applications. The geometric and physical properties of the FEA model have been addressed, and shape deflection algorithms were developed by subjecting the model to three different loading cases. A number of sensitivity analyses have been investigated and include the placement of the fiber lines on the wing, the density of sensors along the lines, the measurement uncertainty, and the effects of loading on the wing.

8.4.2 Computational Model

The finite element model was created to represent an actual experimental swept plate that was designed, fabricated, and instrumented at NASA's Dryden Flight Research Center. Fiber-optic strain sensors were placed at the top and the bottom surfaces of

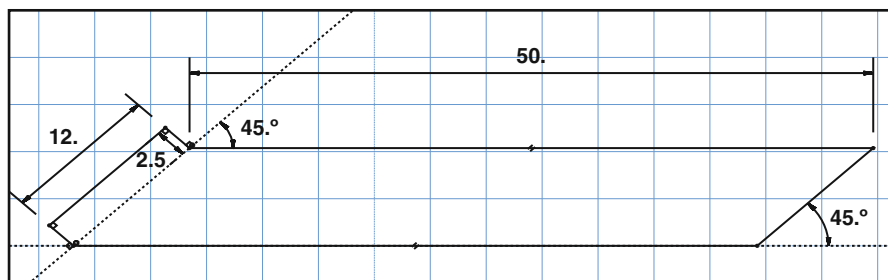


Fig. 8.22 Swept plate dimensions (inches)

Fig. 8.23 Swept plate sensor layout



the experimental swept plate, and strain data were collected from the plate when it was subjected to various loading conditions. The swept plate is made of 6061-T6 aluminum with a Young's modulus of $E = 10$ Msi. The swept plate is 12 in. wide and 0.19 in. thick and has a span of approximately 51 in. The plate is swept horizontally at a 45° angle from the fixed end (Fig. 8.22).

Three fibers are located at the top and the bottom surfaces, for a total of six fibers on the plate. Two fiber lines are placed 0.5 in. from the leading and the trailing edges, while the third fiber line is located in the middle of the plate (Fig. 8.23). There are a total of 102 fiber gratings along each fiber line, and strain measurements are available from each.

A combination of Femap and Nastran packages were used to create and analyze the computational models. The FEA models were created such that their physical properties and geometric constraints represent the actual experimental model.

Three loading cases have been considered. The uniform-load (UL) case consists of a total of 12 point loads, 6 along each the leading and trailing edges of the wing. The trailing-edge load (TE) case consists of six equally spaced point loads that are applied to the trailing edge of the wing. Finally, the single-point (SP) load case consists of a single point load located at the corner of the plate located at the tip of the trailing edge, opposite the root. The trailing-edge and single-point loading cases are investigated to assess the torsional load effects on the performance of the deflection prediction algorithm. In the case of the leading-edge loads, the 6 lb point loads are applied closest to the fixed end of the plate. In the case of the uniform load, all point loads are 2 lb each. In the case of the single-point load, the point load is 11 lb.

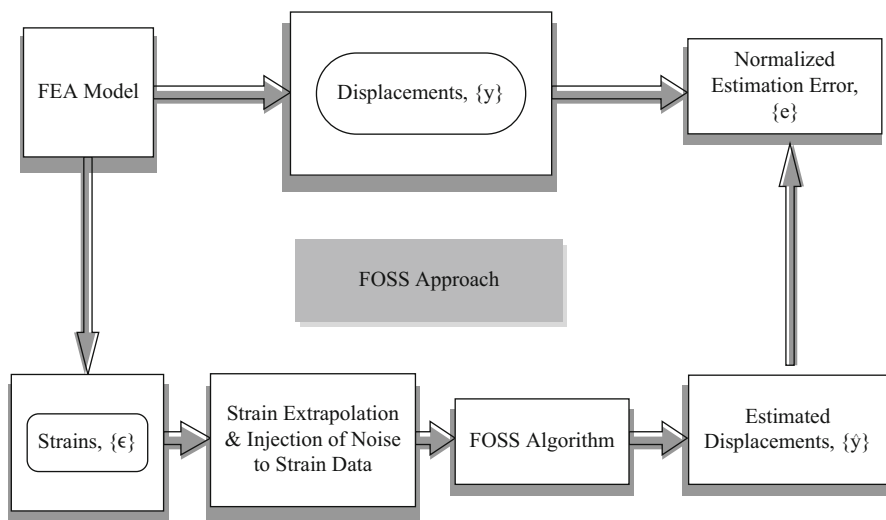


Fig. 8.24 FOSS approach sensitivity analysis

8.4.3 Sensitivity Analysis

A simulation was performed where three fiber lines were placed on both the top and bottom of the swept plate. The fibers were placed 0.5 in. parallel to the leading and trailing edges and one directly in the middle of the plate. Real-life situations were considered when assessing the effectiveness of the deflection prediction algorithms. Placement of the fibers on the wing and spacing of the sensing stations will drastically affect the accuracy of the predictions.

Strain and deflection data were extracted from the finite element model in order to compare the results of the algorithm. Because the algorithm requires the root deflection and strain information, the data was extrapolated to the root using curve fitting techniques and the known locations of the sensing stations. The strain data may be manipulated to add various levels of uncertainty as needed for the analysis. The deflections were estimated using the FOSS DFT transformations.

For each study, the displacements have been estimated, and the corresponding normalized estimation errors, e , were calculated. This process is shown in Fig. 8.24.

8.4.4 Location of Fiber Line and Effects of Loading

The top plate fibers were considered when assessing the effects of fiber location and torsion from loading. The results of the study on the fiber locations and loading conditions are shown in Fig. 8.25. Each case shows the error in deflection prediction for a particular fiber line under three loading conditions. The first loading condition

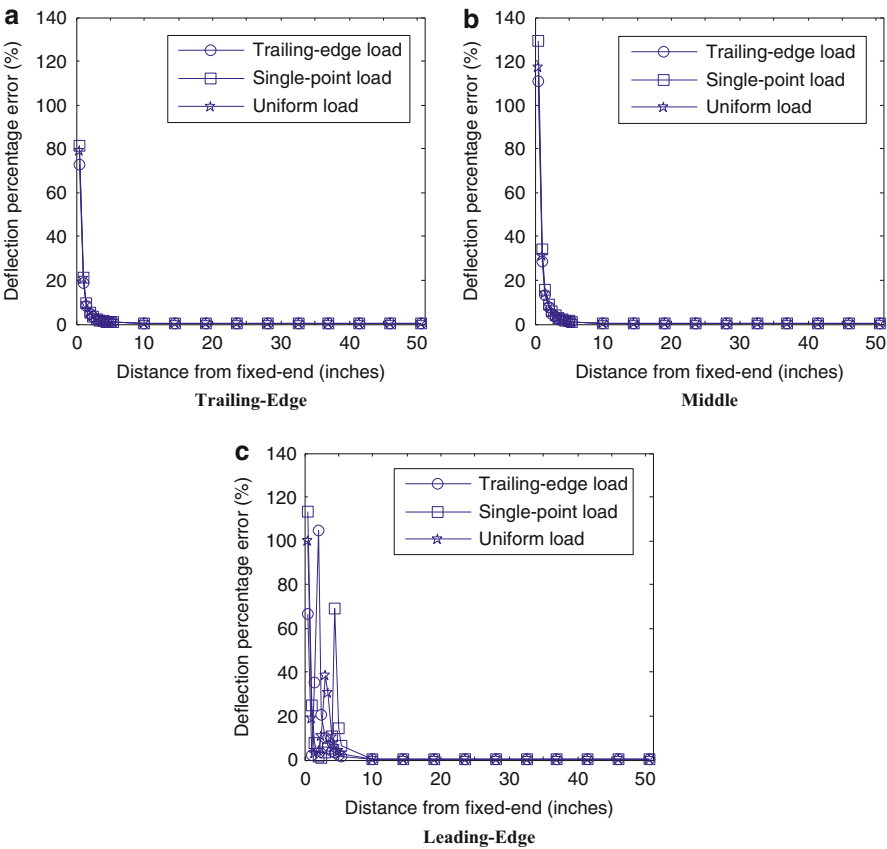


Fig. 8.25 Loading study-fiber line

is the trailing-edge load case, followed by the single-point load and uniform-load cases. Figure 8.25a shows the results for the trailing-edge fiber line, Fig. 8.25b shows the results for the middle fiber line, and Fig. 8.25c shows the results of the leading-edge fiber line.

The deflection prediction algorithm obtained promising results for both the trailing-edge and middle fiber lines. However, the results of the leading-edge fiber line showed high errors near the root of the wing for all three loading conditions. Such high prediction errors and erratic behavior can pose large problems for monitoring applications during flight. As expected, the higher percentage errors are located near the fixed end of the swept wing, where deflections are minimal, while the prediction percentage errors quickly improve toward the tip of the wing.

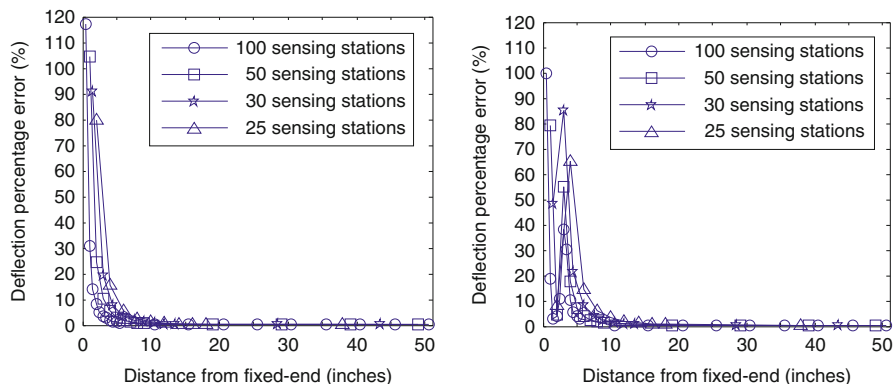


Fig. 8.26 Sensor spatial resolution comparison: uniform loading, middle fiber

8.4.5 Number of FBG Sensors

Fiber-optic technology using FBG provides the potential for providing high spatial resolution. In a real-life application, increasing the spatial resolution of the fiber-optic sensors provides increased accuracy in capturing strain values. However, increasing the spatial resolution raises computational considerations. The degree of resolution needs to be increased in order to obtain the desired accuracy while maintaining a reasonable computational demand. In order to assess the effects of spatial resolution, four cases were considered using a sensor spacing of 0.5, 1, 1.5, and 2 in. This resulted in a total number of sensing stations per fiber of roughly 100, 50, 30, and 25 sensing stations, respectively. Figure 8.26 shows the results of the uniform loading case on the middle fiber. As expected, increasing the spatial resolution greatly increases the accuracy of the algorithm. Additionally, the percentage errors increase at the free end as the spatial resolution decreases.

8.4.6 Sensor Integration and Loading

A Femap/Nastran FEA model was constructed and included the structural elements of the aircraft skin and internal structures, while the nonstructural items were excluded. All panels and the spar have been modeled using both 2-D orthotropic materials with laminate properties. For the wing-boom-tail assembly, all lift loads were applied by using point loads. The lift load was calculated by incorporating a maximum aerodynamic load of 3.0 g. The applied lift load was a distributed load that decreases toward the wing tips, classified as an elliptical load. The moments about the quarter chord of the main wing were also applied to the model by use

Fig. 8.27 Femap model with loading

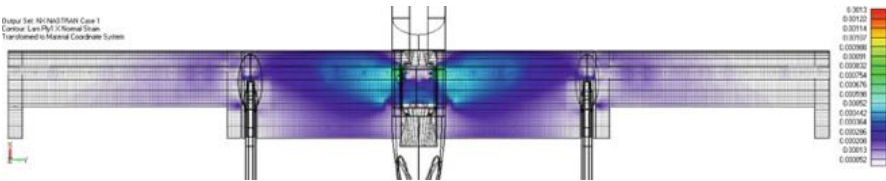
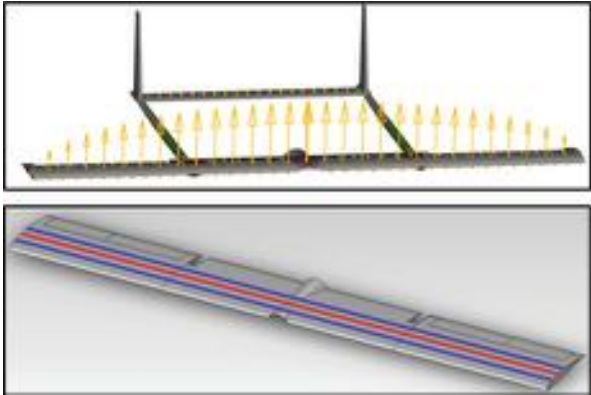


Fig. 8.28 UAV wing strain direction

of nodal loads, and a distributed load was applied to the tail. The wing-boom-tail structure load was applied by specifying the aircraft accelerations in the FEA model. The wing was constrained at the interface between the bulkheads of the fuselage and the main spar of the wing.

For this design study, strain data have been extracted from the FEA model of the UAV wing from along the quarter chord and on either side of the quarter chord. These nodes represent the potential placement of the fiber-optic line on the top of the UAV wing (Fig. 8.27). The direction of the strain is along the span of the wing and can be seen in Fig. 8.28.

Unlike a simple aluminum plate, the UAV wing is not a homogenous structure. It is constructed using various materials and has complex changes in geometry. These abrupt changes in geometry or material properties can be observed from strain measurements. While the overall behavior of the strain from the root of the wing to the tip of the wing agrees with what would be expected in a cantilever beam-like structure, there are abrupt discontinuities in the strain at locations where the materials or geometry in the wing changes. This could be seen near 40 in. from the wing root. These areas are crucial to ensure the structural integrity of the wing.

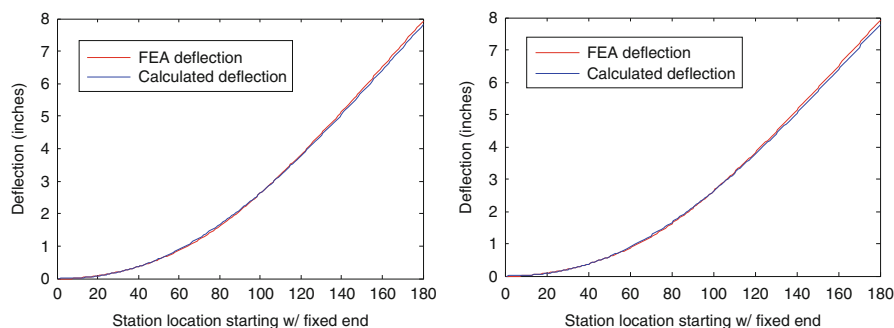


Fig. 8.29 Comparison of deflections for top right and left wing fibers

8.4.7 Analysis and Results

The analysis was performed by extracting the strain data along the span of the wing. Each side of the wing, left and right, was treated as an individual cantilever beam for the purpose of the DTF. Deflection and slope at the root of each fiber were assumed to be zero. In order to obtain evenly spaced strain sensing stations, the locations, strains, and deflections from the FEA model of the UAV were extracted and interpolated at evenly spaced intervals using a polynomial curve fitting function in MATLAB before using in the deflection equations and performing error analysis. Lastly, the half-beam depth was assumed to be equal along the span of the wing.

Figure 8.29 shows the comparison of the calculated deflection along the quarter chord of the UAV wing using the FEA strain data and the DTF deflection equations. Thus, the displacement equations accurately predict the shape of the wing for both the right and left side of the wing.

Using the results from the FEM, the fiber-optic lines have been implemented onto the Odyssey UAV. Two lines of sensors have been routed along the span of the UAV, and a series of diagonals run in between the two lines. The 2 lines of optic fiber that run in the spanwise direction have approximately 360 strain readings per fiber line and will capture strain due to bending. As shown in Fig. 8.30, the diagonal lines are aligned in the 45° orientation and will be utilized to obtain strain due to torsion.

8.5 Flight Simulation

8.5.1 X-Plane Simulation Model

By connecting the Odyssey UAV flight control system hardware to the X-Plane flight simulator, real-time hardware-in-the-loop (HIL) simulations were conducted to verify the flight behavior of the design as well as the optimized theoretical gains.

The geometry of the UAV was simulated using the Plane Maker software and then uploaded into the X-Plane virtual environment (Fig. 8.31). The aerodynamic



Fig. 8.30 Fiber optics on odyssey UAV

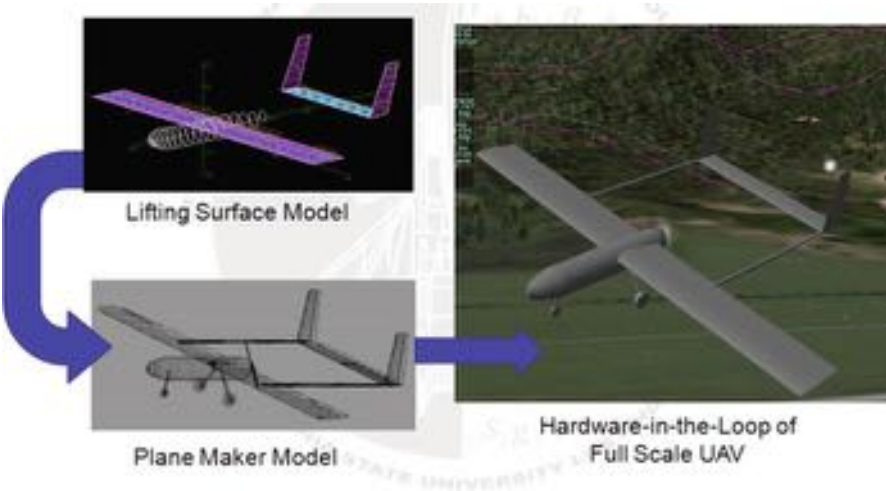


Fig. 8.31 X-plane simulation model

forces and moments calculated from this blade element theory model is a good representation of the aircraft in real environmental conditions. Specifically, wind speed, gust speed, shear angles, and turbulence were obtained in X-Plane as shown in Fig. 8.32 to analyze the dynamic stability of the UAV design.

8.5.2 Autonomous Design

8.5.2.1 Introduction to Flight Control System

If time and resources are not a major concern, designing a custom UAV flight control system (FCS) is a viable option. This allows personalized selection of sensors,

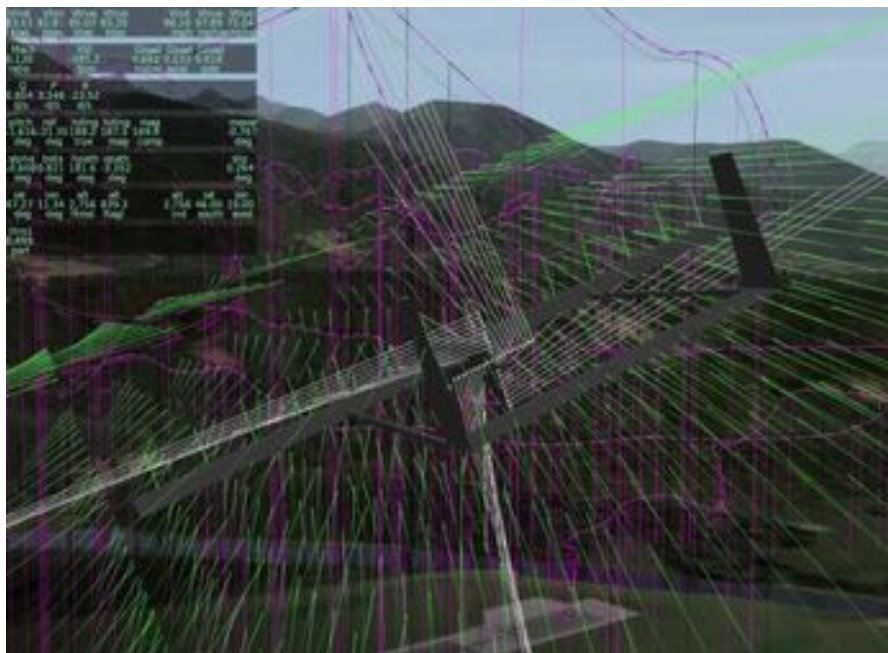


Fig. 8.32 Blade element physics with environmental disturbance

actuators, and processors as well as complete control over the underlying software architecture. The disadvantage of designing a custom FCS is that its design and implementation will have to proceed in pace with the design and the manufacturing of the UAV. Last-minute changes to design, manufacturing, or mission objectives may compromise previous avionic layouts or sensor precision/accuracy requirements in the FCS. In addition, an enormous amount of time is required to code, debug, integrate, and test the finalized FCS first in a simulation environment and then in actual flight.

To save time and recourses, an option to custom designing the FCS is the integration of an “off-the-shelf” FCS/autopilot system such as Cloud Cap Technology’s Piccolo Autopilot or the ArduPilot Mega (APM) FCS into the UAV. The advantage of this choice is time and effort saved on design/layout/programming, but the disadvantage is the possibility of experiencing difficulties from software problems or hardware failures of a system that may have no transparency of the underlying hardware and software architectures.

The “off-the-shelf” APM FCS was selected over the Piccolo Autopilot for the Odyssey UAV since there is more transparency in the APM hardware and software architectures. Further, the APM is more suited for control since it allows the setting of PID (proportional-integral-derivative) controller gains for the servos based on empirically and analytically derived UAV models.

8.5.2.2 Aerodynamic State-Space Models

Various depending system model implementations may need to be implemented based on the mission requirements of the UAV. For high-speed high-maneuverability aircraft, a nonlinear system model may be necessary. However, for aircraft that may need to have considerably large changes in flight conditions, multiple linear models may be needed with the addition of an active gain scheduler if one controller is incapable of meeting the demands of all expected flight regimes. The plant model of an aircraft can be derived analytically based on the well-known equations of motion and linearized about an operating point. Alternatively, there are existing software packages, such as the AVL that will provide a linear plant model based on design data such as the mass distribution and geometry of the aircraft. The disadvantage of this approach is the unknown underlying assumptions in the creation of such a model.

To address the uncertainty in the automatically created plant model delivered by AVL, an alternative approach was used to extract the aircraft's aerodynamic stability and body-axis coefficients and build new state-space (SS) models under the following assumptions: (1) the longitudinal and lateral degrees of freedom (DOF) are uncoupled, and (2) the aircraft's motion will mostly consist of small deviations from its equilibrium flight condition. Equations below show the SS structure assumed and the terms calculated based on available sensor data and assuming the FCS board is placed right at the center of gravity (CG) of the aircraft.

$$\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{w} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -g \\ Z_u & Z_w & u_0 & 0 \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix} \Delta \delta_e$$

$$\begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \phi \end{bmatrix}$$

$$\begin{bmatrix} \Delta \dot{v} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & 0 & (Y_r - u_0) & -g \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix}$$

$$\begin{bmatrix} \Delta v \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix}$$

The uppercase letters denote aerodynamic derivatives/interactions between subscripted states or control surfaces. The states (u,v,w) represent orthogonal velocity components in a body-centered right-hand system with positive (x,y,z)

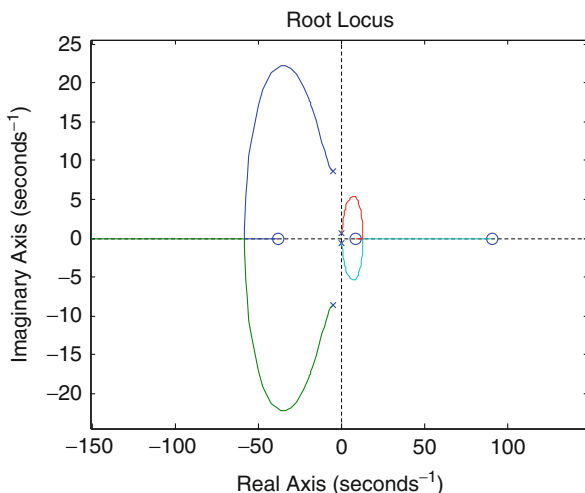


Fig. 8.33 Root locus for Δu with respect to δe

measured along the nose, right wing, and undercarriage of the aircraft, respectively. The angular states ($p, q, r, \theta, \phi, \psi$) denote angular rates of change in pitch, roll, yaw, pitch angle, bank angle, and yaw angle, respectively. The elevator, ailerons, and rudder are denoted by δe , δa , and δr , respectively.

Stability analysis should be performed to determine whether the aircraft has any uncontrollable, unobservable, or unstable modes once the model has been derived. For example, Fig. 8.33 shows the root locus for the longitudinal acceleration (Δu) with respect to elevator input (δe) for the Odyssey's half-scale prototype. Further analysis showed that the SS models are both controllable with only one unstable mode corresponding to the lateral spiral mode (not uncommon for most aircraft).

This chapter has shown the design process of a fixed-wing UAV using a case study. At the time of publication, this aircraft was designed, simulated, and constructed. The maiden flight of the Odyssey will be in about 4 months.

Acknowledgments The authors want to acknowledge and thank the technical support provided by the NASA Dryden Flight Research Center, all the faculty, and the students who have contributed in the past and present to the design and development of the Odyssey UAV at the Structures, Propulsion, And Control Engineering (SPACE) – a NASA sponsored University Research Center (URC) of Excellence at the California State University, Los Angeles (CSULA).

References

- J.D. Anderson Jr., *Aircraft Performance and Design* (Mcgraw Hill, Boston, 1999)
- A. Derkevorkian, J. Alvarenga, J. Bakalyar, L. Richards, S. Masri, H. Boussalis, Evaluation of strain-based deformation shape estimation algorithms for control and monitoring applications, in *2012 SPIE Symposium on SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring*, San Diego, Mar 2012

- A. Derkevorkian, J. Alvarenga, J. Bakalyar, L. Richards, S. Masri, H. Boussalis, Computational experimental studies of deformation shape prediction algorithms for control monitoring applications, in *5th European Conference on Structural Control – EACS 2012*, Genoa, 18–20 June 2012
- M. Emmons, S. Karnani, S. Trono, K. Mohanchandra, W. Richards, G. Carman, Strain measurement validation of Embedded fiber bragg gratings. *Int. J. Optomechatronics* **4**(1), 22–33 (2010)
- Y. Fan, M. Kahrizi, Characterization of a FBG strain gage array embedded in composite structure. *Sens. Actuators A: Phys.* **121**, 305 (2005)
- B.P. Ferdinand, J.E. Russell, D.T. John, *Mechanics of Materials*, 4th edn. (McGraw Hill, New York, 2003)
- W. Ko, W. Richards, Method for real-time structure shape-sensing. U.S. Patent 7520176, pp. 1–13, Apr 2009
- W. Ko, W. Richards, V. Tran, *Displacement Theories for In-Flight Deformed Shape Predictions of Aerospace Structure*. National Aeronautics and Space Administration (NASA), 214612, 2007
- K.S.C. Kuang, R. Kenny, M.P. Whelan, W.J. Cantwell, P.R. Chalker, Embedded fiber Bragg grating sensors in advanced composite materials. *Compos. Sci. Technol.* **61**, 1379–1387 (2011)
- D. Lee, M. Mitrovich, A. Friedman, G. Carman, L. Richards, Characterization of fiber optic sensors for structural health monitoring. *J. Compos. Mater.* **36**(11), 1349 (2002)
- T.H.G. Megson, *Aircraft Structures for Engineering Students*, 3rd edn. (Butterworth Heinemann, Amsterdam/London, 1999)
- T.E. Noll, J.M. Brown, M.E. Perez-Davis, S.D. Ishmael, G.C. Tiffany, M. Gaier, *Investigation of the Helios prototype aircraft mishap*, Vol. I, Mishap Report. National Aeronautics and Space Administration (NASA) I, Jan 2004
- D.P. Raymer, *Aircraft Design: A Conceptual Approach*. AIAA Education Series, 3rd edn. (AIAA, Reston, 1999)
- W.L. Richards, Characterization of embedded fiber optic sensors in advanced composite materials for structural health monitoring, in *Proceedings of SPIE 5390*, pp. 505–512, 2004. (About Stability Analysis Using XFLR5 by A. Deperrois – Nov 2010)
- N.L. Robert, *Machine Design An Integrated Approach*, 3rd edn (Prentice Hall, Upper Saddle River, 2006)
- SPACE Center website, <http://www.calstatela.edu/orgs/space/>
- C.C. Thomas, *Design of Aircraft* (Prentice Hall, Upper Saddle River, 2003)
- K. Wood, T. Brown, R. Rogowski, B. Jensen, Fiber optic sensors for health monitoring of morphing airframes: I. Bragg grating strain and temperature sensor. *Smart Mater. Struct.* **9**(2), 163 (2000)
- G. Zhou, L. Sim, Damage detection and assessment in fiber-reinforced composite structures with embedded fiber optics Sensors – review. *Smart Mater. Struct.* **11**, 925 (2002)

Scott Gruber, Hyukseong Kwon, Chad Hager, Rajnikant Sharma, Josiah Yoder, and Daniel Pack

Contents

| | | |
|-------|---|-----|
| 9.1 | Introduction | 144 |
| 9.2 | Payload Mission Requirements | 145 |
| 9.3 | Preliminary Payload Design Budgets | 146 |
| 9.3.1 | The Power Budget | 147 |
| 9.3.2 | The Weight Budget | 148 |
| 9.3.3 | Volume Allocation | 149 |
| 9.4 | Subsystems Design | 149 |
| 9.4.1 | Communications Subsystem | 149 |
| 9.4.2 | Single Board Onboard Computer Subsystem | 155 |
| 9.4.3 | EO Sensor Subsystem | 158 |
| 9.4.4 | Power Subsystem | 160 |
| 9.5 | Other Payload Design Considerations | 162 |
| | References | 163 |

Abstract

This chapter discusses the payload design issues for small unmanned aerial vehicles (UAVs). Details of several payload design principles to overcome various small UAV constraints imposed by stringent weight, power, and volume are discussed. Throughout the chapter, the efficacy of these principles is demonstrated with the help of the payloads for a fixed wing UAV developed by the

S. Gruber (✉) • H. Kwon • C. Hager • R. Sharma • J. Yoder
Academy Center for Unmanned Aircraft Systems Research, Department of Electrical and Computer Engineering, United States Air Force Academy, Colorado Springs, CO, USA
e-mail: scott.gruber@usafa.edu; hyukseong.kwon@usafa.edu; chad.hager.ctr@usafa.edu; rajnikant.sharma.ctr.in@usafa.edu; josiah.yoder.ctr@usafa.edu

D. Pack
Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX, USA
e-mail: daniel.pack@utsa.edu

Center for Unmanned Aircraft Systems Research at the U.S. Air Force Academy, using this example payload design, a UAV is to be used to autonomously search, detect, localize, and track ground targets. The system requirements for the example application are closely related to those for other small UAV applications.

9.1 Introduction

Clearly, a small UAV has limited payload volume and weight capacity. These limitations drive several additional constraints: electronic components must avoid electromagnetic interference (EMI) interactions, a risk increases by their close proximity; power is limited by the space and weight available for energy storage devices such as batteries; total payload operational time is limited by the available; power processor speed is limited by the weight and power available, and sensor capabilities such as video frame rate and image resolution are limited by available weight allocation.

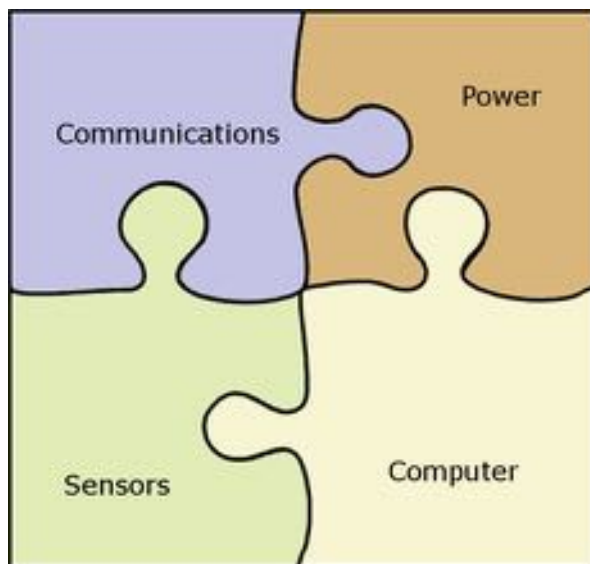
Researchers and developers in academia as well as in industry have worked on designing UAV payloads. Pastor et al. designed a low-cost embedded hardware/software architecture for micro UAVs (Pastor et al. 2007), and Semke et al. suggested an approach to use remote sensing payload design for digital imaging on UAVs for educational development (Semke et al. 2007). Stuckel et al. proposed a payload design for a platform stabilization system to remotely deliver more stable imagery during flights (Stuckel et al. 2011). And Everaerts et al. provided various remote sensing payload designs for visual, infrared, laser, or atmospheric sensors (Everaerts et al. 2004). The focus of this chapter is to design payloads for small UAVs, which can autonomously perform specific intelligence, surveillance, and reconnaissance missions only with an onboard computer and the supporting subsystem payloads.

The constraints of volume and weight play critical roles in the following major trade-offs for a small UAV payload:

- Autonomy versus human operator
- Autonomy (onboard processing) versus available communication bandwidth
- Autonomy (onboard capability) versus flight time
- Minimalist design versus flexibility and adaptability
- Optimal design versus simplicity

Designing a small UAV payload involves trade-offs among the basic subsystems shown in Fig. 9.1. Decisions made for one subsystem will affect the design of another subsystem. While managing this balance, the overall small UAV system constraints of weight and volume must be maintained. For example, if an extensive image processing algorithm is executed on images at a rate of five images per second, the single board computer (SBC) needs to handle the computational load. The larger the computer needs are the more power and cooling are necessary to complete the task. Since the payload battery capacity cannot be increased due to the overall weight restriction, the mission time has to be reduced. If mission time is not negotiable, can another choice be made that still accomplishes the goal of

Fig. 9.1 Major payload subsystems



the mission? Can the image processing rate be reduced from five to one image per second, or can a less computational image processing algorithm be used? These are typical questions that must be answered before a design is chosen.

This chapter provides a method to design an integrated payload that maximizes performance in a small UAV package. The chapter is organized to:

- (a) Define the mission requirements of the payload (Sect. 9.2).
- (b) Create preliminary weight, power, and volume budgets for the payload (Sect. 9.3).
- (c) Start an iterative design of the payload subsystems, constantly interacting among all subsystem designs to ensure that performance is maximized within the system constraints (Sect. 9.4).
- (d) Address other areas that can affect the payload design (Sect. 9.5).

Throughout this chapter, examples are provided to help explain the related concepts. There is also an underlying emphasis to ensure EMI does not contribute to the loss of the UAV's global positioning system (GPS) signal. Most small UAVs rely on GPS signals for navigation, so reliable performance of the GPS system is necessary for the success of a small UAV mission.

9.2 Payload Mission Requirements

Before a detailed design can be developed, the requirements of the payload must be defined. Some questions that should be asked that affect a small UAV payload design are:

- What is the flight duration required for the mission?
- What is the specific task of the mission? For example, searching for specific targets, setting up surveillance posts, providing a communications relay, etc.
- What are the detailed mission subtasks? How large is the search area?
- How long will the payload systems need to be operated on the ground before the flight?
- What is the maximum distance from the ground station that the UAVs will be operating?
- How many UAVs will be in flight?
 - If multiple UAVs, how is collision avoidance incorporated? For a small UAV, altitude separation is typically used so that avoidance hardware isn't necessary.
- What are the target characteristics?
 - Required accuracy of localization
 - Moving or stationary targets
 - Number of targets
 - Target separation
 - Size, shape, and color of targets
- What are the allowed frequencies that can be used for communication between UAVs and the ground station?
- What are the situational awareness requirements of the ground station operator?
 - Imagery refresh rate
 - Image quality and size
 - UAV position, mission status, and health
- Will other devices be operated in the same area that can cause communication interference?
- What are the environmental requirements (vibration, shock, humidity, and temperature)?

9.3 Preliminary Payload Design Budgets

The preliminary payload design is governed by two limiting factors: payload weight and volume. Power, in the form of batteries, is the main contributor to the payload's weight. Any reduction in power requirements of the payload subsystems will result in either an overall weight reduction or an increase in the possible mission time. Thus, in the flow diagrams used in each of the subsystem design sections, a reference to WPV is used to represent weight, power, and volume. Note that cost is another important factor in developing a design. This is specific to each small UAV application and is not addressed in this chapter.

When developing a preliminary design budget, an investigation of likely components needs to be completed. These components can then be used to determine the preliminary allocations of weight and volume. Figure 9.2 is an example payload

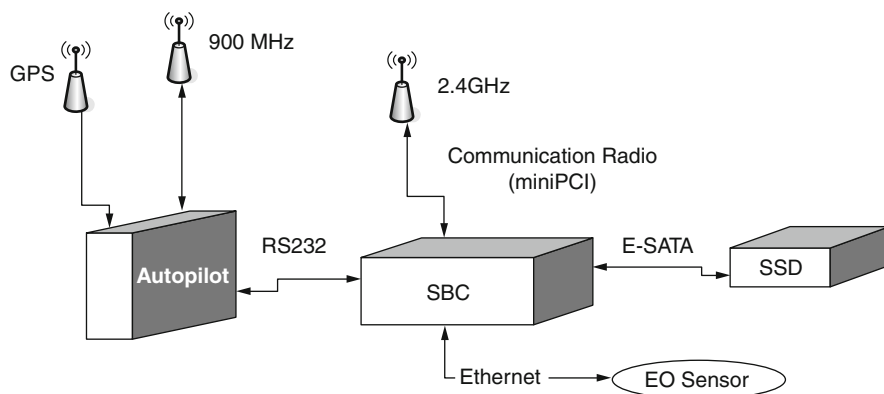


Fig. 9.2 An example of a payload hardware design: *SBC* (single board computer), *EO* (electro-optic) sensor, *SSD* (solid-state disk), and *E-SATA* (external serial advanced technology attachment)

hardware design including major components and I/O between the components. To start the payload sizing, a preliminary power budget is useful since battery weight is a major weight contributor.

9.3.1 The Power Budget

For a small UAV gasoline propulsion system, having an onboard generator is unlikely due to the weight constraint, so a battery is necessary to support the payload system. If the propulsion system is electric, the propulsion batteries are typically used just for propulsion to maximize flight time, again requiring the payload to have its own battery. Note that if the payload has small power requirements, the small UAV design may opt to use the propulsion batteries to support the payload with the understanding that the flight duration will be impacted. This section will focus on an example payload system with its own battery.

An estimated payload is used to compute an overall payload power requirement. Table 9.1 presents an example payload using an atom-based processor, a 2.4-GHz radio, and a security camera as the key payload components. Note that the camera in this example has electronic panning capability, so it addresses all requirements of an electro-optical (EO) sensor system. Since this table is developed early in the design, a 30 % management reserve is added to the power for use if needed as the design progresses. Based on this analysis, the SBC system (processor and disk) has a 14-W budget, the communications system has a 3-W budget, and the EO sensor system has a 3.6-W budget. Note that the current requirement for the wireless card assumes it is transmitting a little more than 50 % of the time and that the solid-state disk (SSD) has 50 % utilization. Based on the mission's requirements, these average currents can be adjusted. The available battery life is 0.9 h. If the combination of ground time and flight time usage is close to 0.9 h, a higher capacity battery should be employed.

Table 9.1 Example power budget

| Component | Voltage | Current | Watts |
|---|---------|-------------------|------------------------|
| Aurora/Corona SBC (http://www.diamondsystems.com/products/aurora/) | 5 | 2.0 | 10.0 |
| 32 GB SSD | 5 | 0.8 | 4.0 |
| Ubiquiti XR2 wireless card (http://www.ubnt.com/xr2) | 3.3 | 0.9 | 3.0 |
| Axis 212 PTZ camera (http://www.axis.com/products/cam_212/) | 5 | 0.7 | 3.6 |
| Misc. parts (servo switch, RC receiver, etc.) | 5 | 0.2 | 1.0 |
| Total watts | | | 21.6 |
| | | <i>Efficiency</i> | <i>Watts</i> |
| Adjusted for efficiency of DC-DC converter | | 0.9 | 24.0 |
| Adjusted for 30 % management reserve | | 0.7 | 34.3 |
| | | <i>Batt volt</i> | <i>Needed current</i> |
| Battery voltage and current for required power | | 14.8 | 2.3 |
| | | <i>Batt AH</i> | <i>Available hours</i> |
| Lithium polymer battery amp hours (AH) and available operational hours | | 2.1 | 0.9 |

Table 9.2 Example weight budget

| Component | Weight (g) | Percent weight |
|---|------------|----------------|
| Aurora/Corona SBC w/enclosure | 400 | 18.0 |
| 32 GB SSD, enclosure, and cable | 200 | 9.0 |
| Ubiquiti XR2 wireless card, cables, antenna | 100 | 4.5 |
| Axis 212 PTZ camera | 500 | 22.5 |
| Misc. parts | 100 | 4.5 |
| Power system (battery, DC-DC, wires) | 480 | 21.5 |
| Total weight | 1,780 | |
| Adjusted for 20 % management factor | 2,225 | 20.0 |
| Total weight estimate | 2,225 | 100 |
| UAV payload weight allowance | 2,270 | |
| Additional weight buffer | 45 | |

9.3.2 The Weight Budget

The maximum available payload weight, an important design constraint, is predetermined by the aircraft's lift and gross weight limits. Using the same components used for the power budget, Table 9.2 shows a preliminary weight budget. This example assumes that the small UAV allocation for maximum payload weight is 5 lb. In this case, the management reserve is 20 %. The resulting weight budgets are 27 % for the SBC system (includes the SSD), 4.5 % for the communications system, 22.5 % for the communications system, and 26 % for the power system which includes miscellaneous components.

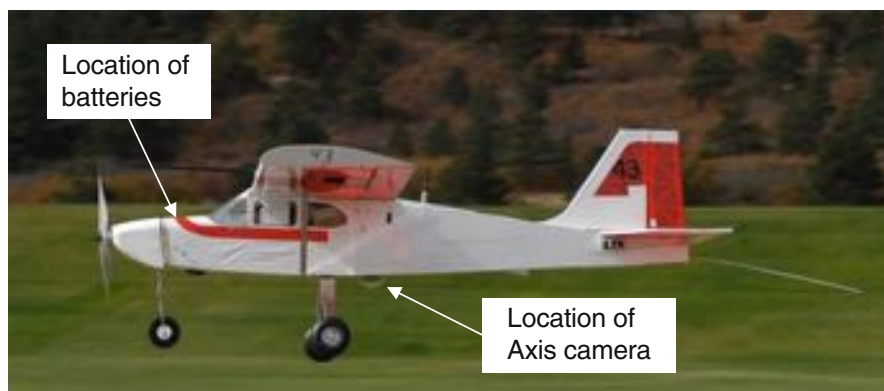


Fig. 9.3 Kadet senior UAV

9.3.3 Volume Allocation

When considering the volume available for payload components, it is not sufficient to simply allocate volume for each component. Instead, components must be configured within the payload compartment(s) so that they are placed for proper payload assembly and do not cause any electronic interference to each other. Furthermore, components must be arranged so that the aircraft center of gravity is maintained for stable flight. For example, Fig. 9.3 shows a small UAV (Kadet Senior aircraft, <http://www.sigmfg.com/IndexText/SIGRC58.html>). To maintain the proper location of the center of gravity, the heaviest objects, the propulsion and payload batteries, were placed in the front of the fuselage, and the electro-optic camera was placed behind the landing gear.

9.4 Subsystems Design

The following sections are introduced with a flow chart that presents the design flow for each subsystem. The number in each flow element corresponds to the items in the accompanying lists that further explain the flow chart elements.

9.4.1 Communications Subsystem

The design flow of the communications subsystem is described in Fig. 9.4.

1. Define mission constraints.
 - (a) Allowed frequencies. Unless licensed radios are available, the UAV will be using the unlicensed ISM (industrial, scientific, and medical) bands. For UAVs, the 902–928-MHz, the 2,400–2,483.5-MHz, and the 5,725–5,850-MHz bands are commonly used. These bands can provide one to multiple 20-MHz channel bands, and compliant radios are readily available.

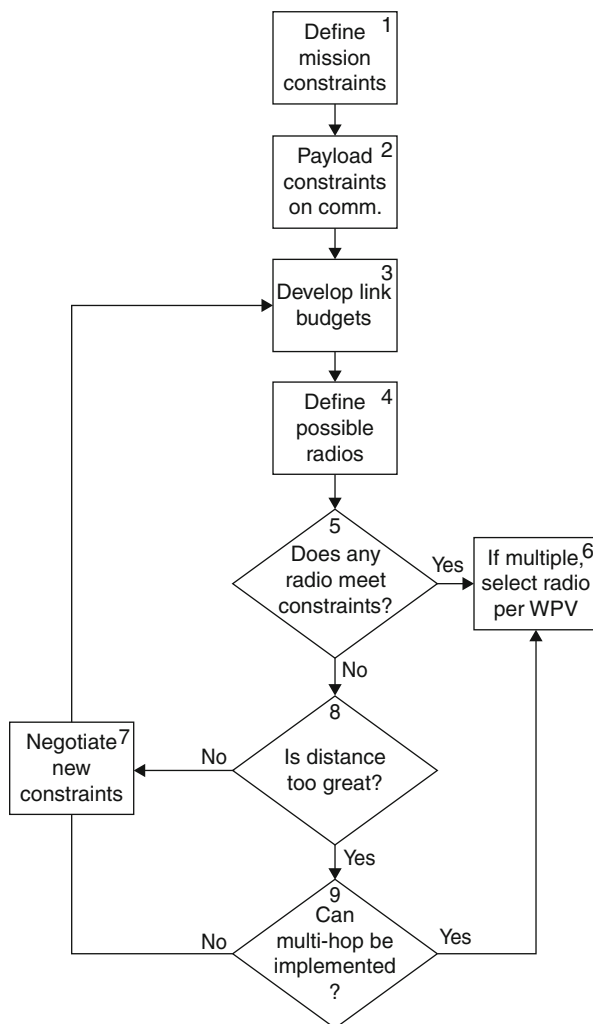


Fig. 9.4 Communications subsystem design flow

Transmission power is restricted per FCC regulations; radios and antenna gain need to be selected that comply with these regulations. Note that the 902–928-MHz band is frequently used for the autopilot wireless link.

- (b) Maximum distance. This distance is used to determine the link budget.
- (c) Number of UAVs in flight will determine if a multi-hop network is possible and the total bandwidth available at the ground station.
- (d) Other devices operating in the same area create noise and limit the communication system range if broadcasting over the same frequencies.

- (e) Situational awareness requirements of the ground station operator drive the bandwidth requirements of the system. A number of key questions to be asked are:
- (i) What command and control messages are necessary and what is the update rate of the messages?
 - (ii) What aircraft status messages need to be reported to the ground station? i.e., aircraft health, stage in mission, etc.
 - (iii) What size and compression factor of images are acceptable? Imagery is the largest consumer of the communication bandwidth. If a communication link is broken during a mission, imagery collected during the lost link may not be useful for the operator. If the imagery is sent using Transmission Control Protocol (TCP), the images have guaranteed delivery. During a lost link, these images are placed in a transmission queue and sent when the link is reestablished. Critical command and control messages need to wait until the queue is emptied of images. If only the current imagery is needed for situational awareness, imagery should be sent using User Datagram Protocol (UDP) which does not guarantee delivery. Thus, the imagery will not sit in a queue, and command and control messages are not delayed when the link is reestablished.
 - (iv) If multiple UAVs are used, how does the imagery need to be presented? To minimize the overall bandwidth, a thumbnail image can be displayed to the operator. The operator then has an option to select a thumbnail to start receiving larger imagery, reducing the overall system bandwidth.

Table 9.3 is an example calculation used to estimate the bandwidth requirements. In this UAS configuration, four UAVs converse with the ground station. Only one operator-selected UAV sends high-resolution imagery (320×240 pixels) to the ground station, and all four UAVs send thumbnail images. The “Number/s” column defines how many messages per UAV are allowed, and the “Total

Table 9.3 Estimated communication bandwidth requirements

| Information | Size (Kb) | Number/s | Total streams | Total Kb/s |
|------------------------------------|-----------|-----------------|---------------|--------------|
| Image 320×240 @ 10 % comp | 20 | 10 | 1.25 | 250 |
| Image 80×60 @ 30 % comp | 1 | 4 | 5 | 20 |
| Control messages | 0.1 | 5 | 5 | 2.5 |
| Health messages | 0.1 | 1 | 5 | 0.5 |
| Ground station messages | 0.1 | 2 | 5 | 1.0 |
| Other | 0.1 | 1 | 5 | 0.5 |
| Total | | | | 274.5 |
| Bandwidth efficiency | 25 % | | | |
| Bandwidth available (Kb/s) | Kb avail | <i>Kb avail</i> | | <i>Usage</i> |
| 6,000 | 1,500 | 187.5 | | 146 % |
| 12,000 | 3,000 | 375 | | 73 % |
| 24,000 | 6,000 | 750 | | 37 % |

streams” column is based on the number of UAVs in the system plus an increase of 25 % to assume 25 % of the messages are being relayed through another UAV (multi-hop). An assumed 25 % efficiency for the system accounts for message overhead and resends of lost messages. The total message bandwidth required is then compared to available data bandwidths for the 802.11-g protocol. The 6-Mb/s system does not have the capacity and a 12-Mb/s system is required. Since imagery is the largest bandwidth consumer, a reduction in frames per second could get the bandwidth requirement to a more manageable size:

2. Define payload constraints on the communication system. These are constraints generated by the other components of the payload. Some of the questions that must be answered are:
 - (a) If multiple UAVs are used, does one UAV need to communicate with another, or do all UAVs communicate directly with the ground station? If UAVs communicate with each other, is the link between UAVs direct or can the messages be routed through an access point at the ground station?
 - (b) What is the weight allocation for the communication system?
 - (c) What is the power allocation for the communication system?
 - (d) What are the input/output (I/O) interfaces available from the onboard computer?
3. Develop a communications link budget. A communications link budget is a method to estimate how well a communication channel will work for a specific system. It is a good tool for comparing multiple radios, antennas, and frequency bands. This section will not provide a detailed description but gives an example calculation.

The link budget is based on the Friis equation (1946): $P_R = P_T G_T G_R \left(\frac{\lambda}{4\pi d}\right)^2$. The power received by a radio, P_R , is equal to the power transmitted by a second radio, P_T , that is focused by the gains of the transmit and receive antennas, G_T and G_R , but is attenuated by the distance the signal has to travel. The last term, $\left(\frac{\lambda}{4\pi d}\right)^2$, is usually referred to as the *free space loss*: the λ represents the wavelength of the carrier wave, and d is the distance between the antennas.

The equation can be expanded to include additional losses and represented in decibels to simplify the calculations. $P_{R_{dB}} = P_{T_{dB}} + G_{T_{dB}} + G_{R_{dB}} - L_{FS_{dB}} - L_{Cbl_{dB}} - L_{Pol_{dB}} - L_{Pnt_{dB}}$, where transmit and receive gains are the maximum gains of the antennas, free space loss (L_{FS}) is defined above, cabling losses (L_{Cbl}) are due to the interconnections between the radio and the antenna, polarization loss (L_{Pol}) is due to misalignment of the electric fields, and pointing losses (L_{Pnt}) are due to misalignment of the physical antennas. The actual gain of an antenna varies as measured around the antenna. The typical antennas used on a UAV are dipoles and monopoles which have a gain pattern similar to a doughnut, with the antenna being the axle of the doughnut. These antennas are referred to as omnidirectional due to equal gain in all directions away from the antenna in a plane perpendicular to the antenna.

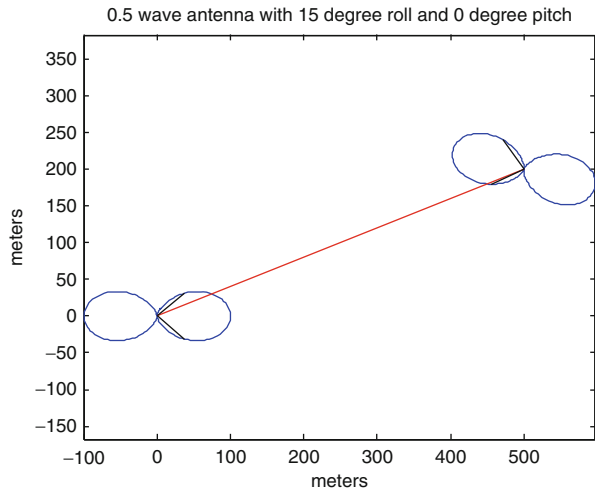


Fig. 9.5 Antenna pointing loss

The design of an antenna generates an alignment of its electric field, referred to as polarization. The most common type used in UAVs is linear polarization, where the electric field aligns with one plane. In the case of a dipole antenna, the field is aligned with the length of the antenna. Thus, for a UAV with a half-wave dipole antenna mounted vertically to the fuselage, the ground station antenna should also be mounted vertically. During flight, the antennas will not always be aligned due to the pitch or roll of the aircraft. This misalignment is referred to as polarization loss and is defined as $20 \log_{10}(\cos(\theta))$, where θ is the maximum misalignment angle. For missions where altitude is held constant, θ can be approximated by the maximum roll angle allowed by the autopilot.

Even if the antennas have no polarization loss, there will still be losses because the aircraft is at a different altitude from the ground station and when the aircraft is rolling away from the ground station. Figure 9.5 is an example of these two conditions. The blue outline is the gain pattern for a $\frac{1}{2}$ wavelength dipole antenna, the red line is the line of sight, and the black lines mark the point in the antenna gain pattern where the gain is $\frac{1}{2}$ of the maximum pattern gain. Antennas should be selected to ensure that the geometry of the UAV system stays within these $\frac{1}{2}$ power or 3 dB points. When maintained, the maximum pointing losses can be assumed to be 3 dB.

Let’s demonstrate the use of the link budget through an example. Two radio systems are being compared, a 2.4-GHz radio and a 900-MHz radio. A data bandwidth of 11 Mb/s needs to be maintained, resulting in transmit and receive powers presented in Table 9.4. Other system parameters are also defined in Table 9.4.

Table 9.4 Communications link budget problem definition

| Parameter | 2.4 GHz radio | 900 MHz radio |
|--|----------------------|----------------------|
| Required receive power for 11 Mb/s (dBm) | −92 | −90 |
| Transmit power for 11 Mb/s (dBm) | 28 | 28 |
| Maximum distance (mile) | 1 | 1 |
| Ground station antenna gain (dBi) | 4 | 4 |
| Aircraft antenna gain (dBi) | 2 | 0 |
| Maximum pointing error | Half power beamwidth | Half power beamwidth |
| Maximum bank angle (°) | 15 | 15 |
| Radio to antenna cable losses at each end (dB) | 0.5 | 0.5 |

Table 9.5 Communications link budget example solution

| Parameter | 2.4 GHz radio | | 900 MHz radio | |
|--|---------------|-----|---------------|-----|
| Power transmit for 11 Mb/s | 28.0 | dBm | 28.0 | dBm |
| Ground cabling loss | −0.5 | dB | −0.5 | dB |
| Ground station antenna gain | 4.0 | dBi | 4.0 | dBi |
| Ground pointing loss (half power pointing error) | −3.0 | dB | −3.0 | dB |
| Free space loss | −104.4 | dB | −95.8 | dB |
| Polarization loss | −0.3 | dB | −0.3 | dB |
| Aircraft pointing loss (half power pointing error) | −3.0 | dB | −3.0 | dB |
| Aircraft antenna gain | 2.0 | dBi | 0.0 | dBi |
| Aircraft cabling loss | −0.5 | dB | −0.5 | dB |
| Power received (total of above) | −77.7 | dBm | −71.1 | dBm |
| Required receive power for 11 Mb/s | −92.0 | dBm | −90.0 | dBm |
| Link margin | 14.3 | dB | 18.9 | dB |

Table 9.5 presents the solution of the link budget. The free space losses are calculated as

$$2.4 \text{ GHz radio} : 2.45 \text{ GHz} : 20 * \log_{10}((4\pi * 1,609)/0.122) = 104.4 \text{ dB}$$

$$900 \text{ MHz radio} : 915 \text{ MHz} : 20 * \log_{10}((4\pi * 1,609)/0.328) = 95.8 \text{ dB}$$

And the polarization loss is calculated as

$$\text{PolarizationLoss} : 20 * \log_{10}(\cos(15^\circ)) = 0.3 \text{ dB}$$

A good link margin should be above 10 dB. In this case, both radios exceed the link margin. Since both systems have sufficient link margin, other factors should

be considered such as in-band interferers. If the autopilot wireless link uses the 900-MHz band, there is potential interference for that radio. It would then be better to choose one of the 2.4-GHz channels.

4. Define possible radios. The link budget in Step 3 guides the choices of possible radio solutions. After possible radios are chosen, additional link budget comparisons can be made for final selections. Other considerations when choosing a radio are:
 - (a) What I/O interfaces are available from devices connected to the radio? Typical radio interfaces available are mini-PCI and PCI-Express. There are also radios available for connecting to the USB or Ethernet busses.
 - (b) What antenna connectors are available on the radios? A smaller connector, such as an SMA, is much lighter than the bulky N connector.
 - (c) Select antennas based on required coverage of the mission area. Directional antennas are not feasible on a small UAV because there isn't a sufficient weight budget to carry a pointing device. Also, the more directional an antenna is, the larger it is. Thus, small UAVs use omnidirectional antennas. On the other hand, the ground station can use a directional antenna if the UAV mission area is away from the ground station such that the UAVs will remain within the 3 dB beamwidth of the antenna.
5. Does any radio meet the communication system constraints?
6. Introduce WPV factors to make the final decision. Reduction in weight and power can either leave more allowance for other payload subsystems or increase flight time.
7. If a solution cannot be found, identify what can be supported and work with the customer to develop new constraints. Multiple design loops are typically required to develop a robust system.
8. Is the only constraint that wasn't met due to having an insufficient link margin at the required distance?
9. A multi-hop solution may be possible. In a multi-hop solution, messages from a distant UAV are relayed by another UAV. Note that this requires a doubling of bandwidth for any multi-hopped messages since they are sent first by the originating UAV and then again by the relaying UAV. Also, standard 802.11 a/b/g access points do not support multi-hop so an ad hoc network would need to be created.

9.4.2 Single Board Onboard Computer Subsystem

Note that references to the SBC refer to the entire computing system, including storage devices. Figure 9.6 shows the design flow.

1. Define mission constraints.
 - (a) Mission time. Define how long the SBC will be running, both prior to takeoff and during the mission.
 - (b) Environmental requirements affect the selection of the SBC system.

- (i) Will the interconnection of the SBC with other systems handle shock and vibration?
 - (ii) Will a hard disk be a viable solution or must a solid-state disk (SSD) be employed?
 - (iii) Do expected humidity and temperature conditions affect the SBC choice?
2. Define payload constraints on the SBC system.
 - (a) What are the computing requirements for processing images?
 - (b) What are the computing requirements for control algorithms?
 - (c) What are the computing requirements for processing sensor outputs?
 - (d) What I/O is needed to interface to autopilot, sensors, and communications systems?
 - (e) What operating system is necessary to support software requirements?
 - (f) What is the weight allocation for the SBC system?
 - (g) What is the power allocation for the SBC system?
3. Develop performance budgets. Table 9.6 is an example that assigns the SBC resources to different software tasks.
4. Look for possible SBC solutions that meet the performance budgets. The solution must also consider storage requirements.
5. Were any SBCs found that meet the requirements?
6. If there were, does the SBC create any GPS noise?
7. If multiple choices exist, select the best choice using WPV factors. Reduction in weight and power can either leave more allowance for other payload subsystems or increase flight time.
8. If no viable computing solution can be found, new constraints must be negotiated with the customer.
9. If an onboard SBC solution can't be found, is a ground solution viable?
10. Can a solution to the GPS EMI issue be found?
 - (a) In attempts to minimize weight, an SBC without an enclosure may be selected. Without the enclosure, the Faraday cage that contains the EMI is not present and noise may interfere with GPS.
 - (b) Directly connecting an SSD to the SBC via the serial advanced technology attachment (SATA) bus will generate severe GPS noise. SATA generates broadband noise to over 2 GHz. If requiring an external drive, either choose a parallel bus or incorporate SBC and SSD enclosures that support E-SATA and use the shielded E-SATA cable between them.
11. Areas to consider for a ground-based computing solution:
 - (a) A faster computer that doesn't need to meet weight requirements could be available for ground use. If performing image processing, high-resolution images from the aircraft may be required. Larger images require greater communication bandwidth, which increases with the number of UAVs in use. Can the communications system handle this bandwidth?
 - (b) With all processing on the ground, what happens to the success of the mission if messages are lost?

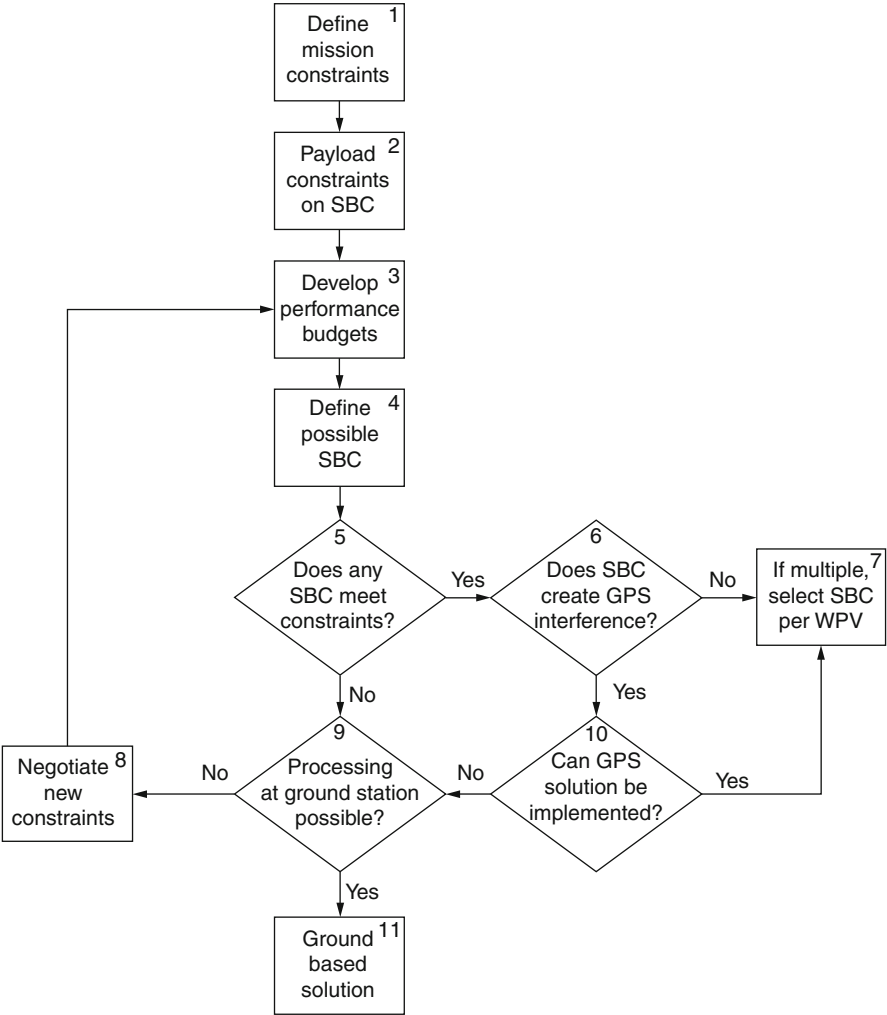


Fig. 9.6 SBC subsystem design flow

Table 9.6 Example of aircraft software budget table

| Priority | Component | CPU allocation (%) |
|----------|----------------------|--------------------|
| 1 | Image capture | 10 |
| 2 | Planning and control | 10 |
| 3 | Image processing | 45 |
| 4 | Data fusion | 15 |
| 5 | Housekeeping | 10 |
| Total | | 100 |

9.4.3 EO Sensor Subsystem

Though other sensors may be used, the electro-optical sensor is predominantly employed in small UAVs. This section presents design considerations for the EO sensor, but the methods can be readily applied to other types of sensors. The EO sensor subsystem includes the sensor hardware, image capture method, and any required gimbal mechanism for panning the sensor. Note that an inertial measurement unit (IMU) is not listed. For a small UAV, weight considerations limit the use of an IMU dedicated to the sensor. The aircraft attitude can be provided by the autopilot to the image processing system to geo-register the image. Figure 9.7 shows the design flow.

1. Define mission constraints.
 - (a) If multiple UAVs are used, is collision avoidance achieved through altitude separation? Would a camera with variable zoom level be needed so that a common ground footprint is maintained for all altitudes that will be used?
 - (b) Based on the target characteristics and ground coverage,
 - (i) What image processing techniques need to be employed?
 - (ii) Will sensor panning be required?
 - (iii) What image quality and size are needed?
 - (iv) How many images per second are required to process?
 - (c) Environmental requirements.
 - (i) Will aircraft vibration impact image quality and registration?
 - (ii) Will the sensor meet humidity requirements? Will it meet shock requirements?
2. Determine constraints from rest of payload. The EO sensor needs to comply with its allocation of weight, power, and volume, but it also drives the SBC computational requirements. The image processing algorithm selection needs to consider a reasonable SBC capability for a small UAV. Sensor fusion techniques may be employed to account for inaccuracies in the image processing results.
3. Develop a decision matrix for the EO sensor. Table 9.7 presents an example decision matrix with three possible candidates.
4. Define possible EO sensors. Two general types of EO sensors are typically employed: analog and digital. The analog sensor either complies to National Television System Committee (NTSC) or Phase Alternation Line (PAL) formats, with NTSC common in the U.S. and PAL common in Europe. An analog sensor requires an image capture board to convert the image stream to individual images in a common digital format such as Joint Photographic Experts Group (jpeg). Digital sensors provide an image that can be directly processed, usually in a raw format.

Each manufacturer of an EO sensor has specifications for their product, but the specifications don't provide sufficient information to determine if the color quality will meet the project's image quality requirements. After defining the best candidates, samples should be purchased so that image quality can be evaluated for the project's application.

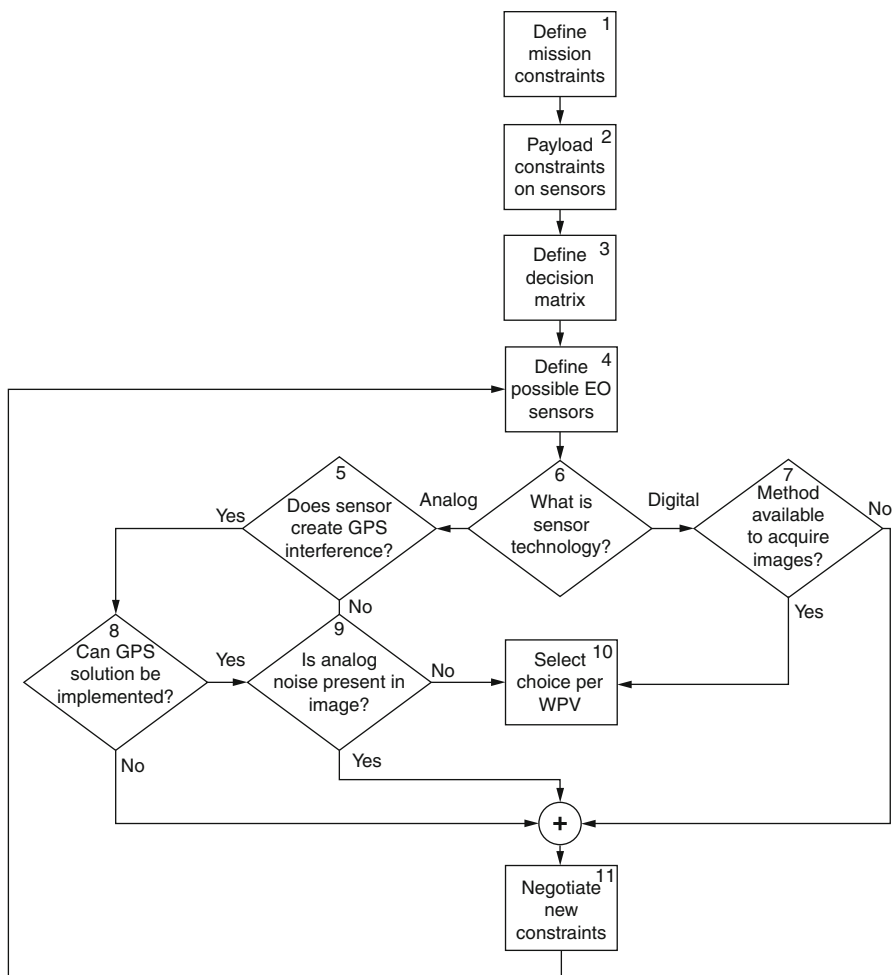


Fig. 9.7 EO sensor subsystem design flow

5. Will the analog sensor generate GPS noise? Though a digital EO sensor should also be checked, NTSC analog sensors use a clock speed that has a harmonic signal at the GPS frequencies. This noise is further increased during the image capture process. PAL analog EO sensors use a different frequency and do not generate the same noise.
6. Determine path between analog and digital.
7. Many digital EO sensors do not directly connect to a common SBC bus such as USB or Ethernet; thus, a field programmable gate array interface must be used. Test interface units are typically available from the manufacturer of the EO sensor though they tend to be bulky and don't have robust connections. Some complete sensors are available with either Ethernet or USB interfaces.

Table 9.7 Comparison of pan-tilt-zoom camera systems

| | AXIS-212 PTZ (http://www.axis.com/products/cam_212/) | WV-SC385 (http://panasonic.net/pss/security/products/hd/wv-sc385/index.html) | EVI-D100 (http://pro.sony.com/bbsec/ssr/cat-industrialcameras/cat-robotic/product-EVID100/) |
|------------------------------|--|--|--|
| Manufacturer | Axis communications | Panasonic | Sony |
| Image sensor type | 1/2" CMOS | 1/3" CMOS | 1/4" Super HAD CCD |
| Pixel resolution | 640 × 480 | 1,280 × 960 | 768 × 494 |
| Frame rate (fps) | 30 | 30 | 30 |
| Video signal output | HTTP/FTP (digital) MPEG-4 JPEG | HTTP/FTP (digital) H.264 JPEG | NTSC (analog) Y/C |
| Pan/tilt | ±70°/±52° (electronic) | 0–350°/–30 to 90° (mechanical) | ±100°/±25° (mechanical) |
| Zoom | 3× electronic zoom | 18× optic + 12× digital zoom (3.2–55.2° (H)/2.4–42.1° (V) AOV) | 10× optic + 40× digital zoom (6–65° AOV) |
| Weight (g) | 504 | 900 | 860 |
| Dimension (mm ³) | 89 × 94 × 77 | 115 × 115 × 155 | 113 × 130 × 122 |
| Power consumption (W) | 3.6 | 12.0 | 13.2 |

8. GPS EMI may be mitigated using a PAL versus NTSC sensor. If the UAV is large enough, separating the analog sensor and capture board from the GPS antenna may sufficiently mitigate the GPS interference.
9. Most analog sensors interlace its image which means that it creates all odd lines in an image and then the even lines of the image. This can create jagged edges since the UAV is moving while the odd lines are captured followed by the capture of the even lines. The performance of edge detection algorithms are impacted by the jagged edges.

Other considerations:

- If an analog sensor is used and images are transmitted to the ground station for processing, additional noise will be introduced from the analog wireless transmission.
- If a greater pan angle is required just for orbiting an object, mounting the sensor at an angle and then always orbiting in the direction that uses the desired tilt of the camera may eliminate the need for a more cumbersome gimbal system.

9.4.4 Power Subsystem

The power system includes batteries and DC-DC converters. Figure 9.8 shows the design flow.

1. Define mission constraints.
 - (a) Flight time will define the amount of battery capacity required for the mission. If the payload operates for a while on the ground before a UAV

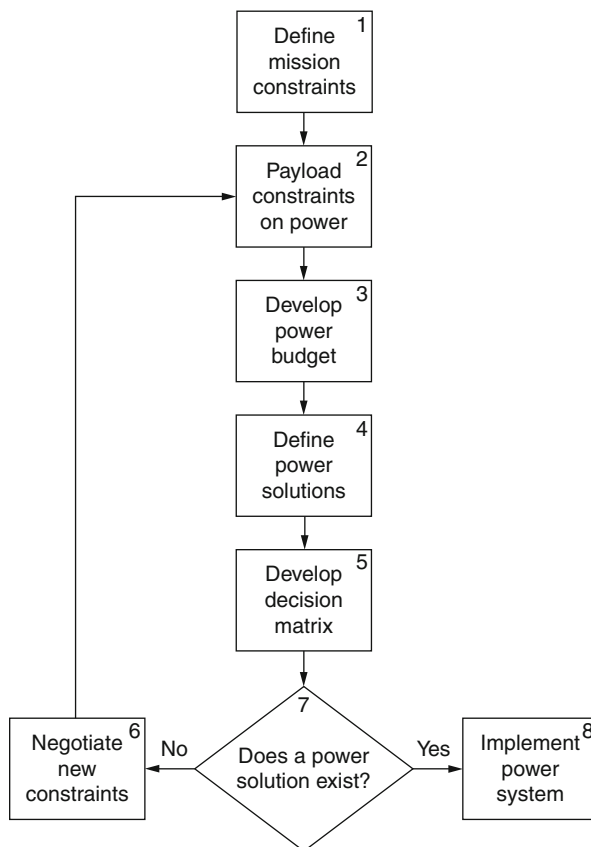


Fig. 9.8 Power subsystem design flow

takes off, additional capacity is needed. A battery backup switch may be employed which allows an external battery to be connected while on the ground and then removed before take-off. With a properly designed battery backup switch, the payload does not see a drop in voltage when the external battery is removed.

2. Payload constraints on the power system.
 - (a) What is the weight allocation for the power system?
 - (b) As subsystem designs evolve, are additional power requirements identified?
3. While completing the subsystem design, verify the power budget using the technique displayed in Table 9.1.
4. Nickel metal hydride (NiMH) and lithium polymer (li-po) batteries are the most common choices of battery technology in use for UAVs. Table 9.8 presents a comparison of the two types. Though a li-po is more expensive, it is a better choice for UAVs because of the higher power density for the same weight. Special care must be taken with li-po technology to ensure that no overcharging occurs and that the batteries are not discharged to less than 3 V per cell.

Table 9.8 Battery trade analysis

| Criteria | Nickel metal hydride | Lithium polymer |
|---------------------|-------------------------|-----------------|
| Cost | Half of lithium polymer | Expensive |
| Rechargeable | Yes | Yes |
| Volume (cu. in.)/AH | 13.6 | 6.6 |
| Weight (ounces)/AH | 18.1 | 6.9 |
| Charge rate (Amps) | 1 | 8 |

5. DC-DC conversion is done with a switching power supply. DC-DC ATX supplies that have been developed for use in automobiles are a good choice. They can use a wide input voltage range and output 3.3-, 5-, and 12-DC voltages that are commonly used by SBCs, sensors, and communication systems.
6. If a power solution is not feasible, a new design needs to be considered with new constraints.
7. If no viable power solution can be found, new constraints must be negotiated with the customer.
8. Decision box regarding existence of viable power solution.
9. Implement the selected power system, always recognizing opportunities to reduce weight and size wherever possible.

9.5 Other Payload Design Considerations

- Flight Time Restrictions: Most small UAV missions will be flown during good visibility and daylight hours so that the aircraft can always be observed in case of erratic behavior. Select sensors to optimize this time of day. Thermal sensors have difficulty identifying targets against warm backgrounds.
- Gas engines create greater vibration than electric motors. If a gas engine is required to increase the flight duration, place a vibration reduction mechanism at the engine instead of each of the payload subsystems.
- Autopilots and thermal cameras often fall under U.S. International Traffic in Arms Regulations (ITAR). Even though you may not be shipping these items to foreign countries, they also cannot be used by foreign nationals located at the facility. Also, if any software that directly communicates with the drivers of these products is also ITAR controlled, it must comply with the same restrictions.
- An autopilot with hardware in the loop (HIL) capability will greatly help in development of control algorithms. The SBC can be directly connected to the autopilot, and the control software can be executed in flight simulators.

Conclusion

In this chapter, payload design principles for small UAVs were discussed and proposed. An example based on a system design used by the USAF Academy Center for Unmanned Aircraft Systems Research described various issues and the corresponding solutions for a system with an autonomous ground target detection

and tracking intelligence, surveillance, and reconnaissance mission. In more detail, mission requirements (budget allocations for power, weight, and volume; and individual subsystem design) were explained with commercial hardware examples.

References

- J. Everaerts, N. Lewyckyj, D. Fransaer, PEGASUS: design of a stratospheric long endurance UAV system for remote sensing. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **XXXV** (Part B), 29–33 (2004)
- H.T. Friis, *Proc. Inst. Radio Eng.* **34**, 254 (1946)
- http://www.axis.com/products/cam_212/
- <http://www.diamondsystems.com/products/aurora/>
- <http://panasonic.net/pss/security/products/hd/wv-sc385/index.html>
- <http://pro.sony.com/bbsc/ssr/cat-industrialcameras/cat-robotic/product-EVID100/>
- <http://www.sigmfg.com/IndexText/SIGRC58.html>
- <http://www.ubnt.com/xr2>
- E. Pastor, J. Lopez, P. Royo, UAV payload and mission control hardware/software architecture. *IEEE Aerosp. Electron. Syst. Mag.* **22**(6), 3–8 (2007)
- W.H. Semke, R.R. Shultz, D. Dvorak, Utilizing UAV payload design by undergraduate researchers for educational and research development, in *ASME 2007 International Mechanical Engineering Congress and Exposition* (ASME, New York, 2007)
- K.J. Stuckel, W.H. Semke, N. Baer, R.R. Shultz, A high frequency stabilization system for UAS imaging payloads. *Struct. Dyn.* **3**, 1411–1419 (2011)

Steven A. Brandt

Contents

| | | |
|--------|---|-----|
| 10.1 | Introduction | 166 |
| 10.2 | Configuration Selection | 166 |
| 10.3 | Fueled Aircraft Sizing and Optimization | 167 |
| 10.3.1 | Sizing Equation | 168 |
| 10.3.2 | Aerodynamic Analysis | 168 |
| 10.3.3 | Constraint Analysis | 169 |
| 10.4 | Mission Analysis | 171 |
| 10.5 | Weight Analysis | 173 |
| 10.6 | Electric Aircraft Sizing and Optimization | 173 |
| 10.6.1 | Sizing Equation | 174 |
| 10.6.2 | Airframe Wing Loading Portion | 174 |
| 10.6.3 | Battery Wing Loading Portion | 175 |
| 10.6.4 | Motor and Prop Wing Loading Portion | 176 |
| 10.6.5 | Solving the Sizing Equation | 177 |
| 10.6.6 | Optimization | 177 |
| 10.6.7 | Example: The Red Falcon | 178 |
| 10.7 | Conclusion | 180 |
| | References | 180 |

Abstract

This chapter describes a rationale for selecting a low-risk UAV (Unmanned Aerial Vehicle) configuration and a methodology for sizing and optimizing the shape of both fueled and electric-powered aircraft. Three examples are given of low-risk UAV configurations. These are shown to be essentially similar from the point of view of optimization and sizing. The sizing methodology for fueled aircraft is described in detail, and a similar one for electric-powered aircraft is

described to the extent that it differs from that for fueled aircraft. The chapter also gives typical values of technology-related parameters used in sizing calculations. It then shows an example of a small UAV designed using this methodology. The UAV design project showed that the methodology is useful in obtaining aircraft of minimum size for the performance and payload required.

10.1 Introduction

The two important questions which a small UAV designer must answer about the air vehicle itself are “What will this aircraft look like?” and “How big will it be?” This chapter describes a rationale for selecting a low-risk UAV configuration and a methodology for sizing and optimizing the shape of both fueled and electric-powered aircraft. It then shows an example of a small UAV designed using this methodology.

10.2 Configuration Selection

To answer the question “What will it look like?,” the savvy UAV designer might easily just ask “What do most other small UAVs look like?” Indeed, there is a very Darwinian process in the small UAV industry that tends to eliminate designs that are less useful, have less performance, or are more expensive and more difficult to use. Figure 10.1 illustrates three aircraft configurations commonly used in small UAVs. Note their similarities.

Although they do not look exactly alike, the three example UAV designs in Fig. 10.1 all employ a single main lifting wing placed high on the fuselage and a stabilizing tail aft. Since the days of the Wright brothers, this general configuration, as opposed to canard or tailless flying wing configurations, has proven most successful. For reasons why, see Brandt (2004).

The three configurations above differ in the details of their aft stabilizing surfaces or empennages. The left-hand one has an inverted T-tail arrangement with the vertical stabilizer and rudder mounted on top of a horizontal stabilizer and elevator. The middle one employs a V-tail which combines the horizontal stabilizing surfaces into a V-shape and combines the rudder and elevators into “ruddervators.” The right-hand design adds to the V a downward-projecting vertical surface. Historically, none of these tail arrangements has proven significantly better than the others, so the choice is left as a matter of preference for the designer.

The other main differences in the three configurations in Fig. 10.1 are the placement of the motor/propeller and the consequent shape of the rear fuselage. The two left-hand designs place their motor/propellers in the front, while the third places it at the aft end of the fuselage. The aft-facing motor/propeller then requires the extra downward-projecting vertical tail to protect it from hitting the ground on landing.



Fig. 10.1 Typical UAV designs

In general, aft-facing motor/propellers have more cooling difficulties and result in heavier designs. The decision to use such a configuration would probably result from a design requirement to keep the nose area of the UAV clear for sensors or other mission payloads. If such a requirement does not exist for a particular design, then a forward placement of the motor/propeller is preferred.

Neither of the main differences in the typical UAV designs just described has a large effect on the sizing and optimization which answers the second question “How big will it be?” Because aft placement of the motor/propeller usually results in a slightly heavier aircraft structure, it usually has a slight effect on the aircraft size. For initial UAV sizing and optimization, this slight effect can be ignored.

Sizing and optimization methodologies differ somewhat between fueled aircraft and those that use electric power. This chapter will describe methods for both types of aircraft.

10.3 Fueled Aircraft Sizing and Optimization

The main difference between fueled and battery-powered aircraft is in the way the aircraft weight changes during the mission. For fueled aircraft, fuel is burned during the mission, so the aircraft’s weight decreases as the mission unfolds. In general,

the weight of battery-powered aircraft does not change unless payload is dropped or otherwise released during the mission. Also, the amount of energy which can be extracted from a given mass of fuel is at present much higher than can be extracted from a similar mass of batteries. For these two reasons, fueled aircraft are often the least expensive way to achieve extremely long flights.

10.3.1 Sizing Equation

The central equation in this analysis is a simple statement that the sum of the weights of the various components of the aircraft must equal the aircraft total weight:

$$W_{TO} = W_{airframe} + W_{payload} + W_{engine} + W_{f_{mission}}$$

When divided by the wing area, the equation expresses the equality between the wing loading of the aircraft and the sum of the portions of that wing loading due to each of the components:

$$\frac{W_{TO}}{S} = \frac{W_{airframe}}{S} + \frac{W_{payload}}{S} + \frac{W_{engine}}{S} + \frac{W_{f_{mission}}}{S}$$

The total wing loading, the left side of the equality, is usually chosen in order to allow the aircraft to meet performance requirements specified by the customer. That analysis is called constraint analysis. The payload is specified by the customer, the airframe and engine wing loading portions are obtained from weight analysis, and the wing loading portion for mission fuel burn is obtained from the mission fuel fraction calculated in mission analysis. The strategy for using this equation to size and optimize the aircraft involves determining values for all but the payload wing loading portion term and then using the weight of the required payload to solve for the required wing reference planform area, S . The resulting equation is called the sizing equation:

$$S = \frac{W_{payload}}{\frac{W_{TO}}{S} - \frac{W_{airframe}}{S} - \frac{W_{engine}}{S} - \frac{W_{f_{mission}}}{S}} \quad (10.1)$$

10.3.2 Aerodynamic Analysis

All of the analysis that follows depends on having good models for the aerodynamics of the UAV being sized. Methods for predicting aerodynamics of small subsonic aircraft are abundant and have acceptable accuracy. See Brandt (2004) for a simple, accurate aerodynamic prediction method. In many cases, a model of the UAV configuration in question may be tested in a wind tunnel for even better accuracy.

10.3.3 Constraint Analysis

Constraint analysis is a method for defining a solution space such that all design points within that space allow the aircraft to satisfy all of the customer's performance requirements. The design points are defined by combinations of wing loading, W_{TO}/S and thrust-to-weight ratio, or thrust loading, T_{SL}/W_{TO} , where W_{TO} is maximum takeoff gross weight and T_{SL} is sea-level static maximum thrust, that is, the maximum thrust the aircraft propulsion system can produce at sea level when the aircraft is sitting still on the ground. The equation for constraint analysis is based on a modification of the equation for specific excess power. Power is the rate of doing work, so for an aircraft thrust power is the thrust produced by the aircraft's propulsion system, T , multiplied by the aircraft's velocity, V . Excess thrust is the difference between the aircraft's thrust and its drag, D . Drag is the resistance created by the air to the aircraft's motion through it. Excess power is the excess thrust multiplied by the aircraft's velocity and specific excess power is the excess power divided by the aircraft's weight. Specific excess power, P_s , is a measure of the aircraft's ability to increase its energy either by climbing or accelerating. All of these ideas are captured in the following equation for specific excess power:

$$P_s = \frac{(T - D)V}{W} = \frac{dh}{dt} + \frac{V}{g} \frac{dV}{dt}$$

where T is thrust, D is drag, W is weight, V is true airspeed, h is altitude, and g is the acceleration of gravity. Dividing by velocity yields

$$\frac{T}{W} - \frac{D}{W} = \frac{1}{V} \frac{dh}{dt} + \frac{1}{g} \frac{dV}{dt} \quad (10.2)$$

The master equation for constraint analysis is obtained by substituting the following relations into (10.2): $T = \alpha T_{SL}$, where α is the ratio of available thrust to maximum sea level static thrust. Available thrust is the maximum thrust an aircraft is able to produce at full throttle for a given flight condition.

$W = \beta W_{TO}$, where β is the specified weight fraction for that design requirement. This is usually specified as a value of 1 for takeoff and a value based on full payload and 50 % internal fuel for most mission performance constraints.

$q = \frac{1}{2} \rho V^2$ is a quantity called dynamic pressure. In the equation for q the symbol ρ represents the air density.

The drag on the aircraft for a given flight condition is represented by a drag coefficient C_D multiplied by dynamic pressure and the reference planform area of the wing, S . Planform area is the area of the shadow of the wing when the sun is directly overhead, including the shadow of the part of the wing that is inside the aircraft's fuselage. The expression for C_D in turn is a quadratic equation called the aircraft's drag polar. The drag polar has three constants and a variable called lift coefficient, C_L . Lift coefficient is related to the actual lift produced by the aircraft in the same way that drag coefficient is related to the aircraft's drag. The constants

in the drag polar are zero-lift drag coefficient C_{D_o} , the induced drag factor k_1 , and a camber effect term, k_2 .

When an aircraft is producing zero lift, $C_L = 0$ and $C_D = C_{D_o}$. When an aircraft produces lift, normally when it is flying, additional drag results from swirling tornado-like vortices which form at the wingtips. This additional drag-due-to-lift is modeled by the k_1 term in the drag polar. Wings with special shaping called camber have less drag for a given amount of lift, an effect modeled by the k_2 term in the drag polar. The value of k_2 is usually negative while k_1 is positive in a typical aircraft drag polar:

$$D = C_D q S = (C_{D_o} + k_1 C_L^2 + k_2 C_L) q S$$

$$C_L = \frac{L}{q S} = \frac{n W}{q S}$$

These substitutions produce the “master equation” for constraint analysis:

$$\begin{aligned} \frac{T_{SL}}{W_{TO}} = \frac{q}{\alpha} \left[\frac{C_{D_o}}{\left(\frac{W_{TO}}{S} \right)} + k_1 \left(\frac{n\beta}{q} \right)^2 \left(\frac{W_{TO}}{S} \right) + k_2 \left(\frac{n\beta}{q} \right) \right] \\ + \frac{\beta}{\alpha} \left[\frac{1}{V} \frac{dh}{dt} + \frac{1}{g} \frac{dV}{dt} \right] \end{aligned} \quad (10.3)$$

The master equation is written in a form that expresses the minimum T_{SL}/W_{TO} which can achieve a given performance requirement at a particular value of W_{TO}/S . A separate form of the master equation is written using values of velocity, load factor, acceleration, rate of climb, etc., for each design requirement specified by the customer. Lines representing these equations are plotted as boundaries on a thrust-to-weight ratio vs. wing loading diagram. Combinations of design thrust-to-weight ratio and wing loading which fall outside the boundary specified by each constraint result in an aircraft design which will not meet that design requirement. When all the constraint boundaries are plotted, they define a solution space which, for all combinations of thrust-to-weight ratio and wing loading within the space, allows the aircraft to meet all the design requirements. Figure 10.2 illustrates a typical constraint diagram. Ideal and actual design points are shown by the circle and plus sign on the plot.

The design point chosen on the constraint diagram specifies the value of W_{TO}/S to be used in the sizing equation. It also specifies the value of T_{SL}/W_{TO} which, with a little manipulation, produces the value of W_{engine}/S required to solve the sizing equation. This manipulation requires only a knowledge of the technology level of the engine, as expressed by the thrust-to-weight ratio of the engine itself, T_{SL}/W_{engine} . With this value in hand, W_{engine}/S is calculated as follows:

$$\frac{W_{engine}}{S} = \frac{\frac{T_{SL}}{W_{TO}} \frac{W_{TO}}{S}}{\frac{T_{SL}}{W_{engine}}}$$

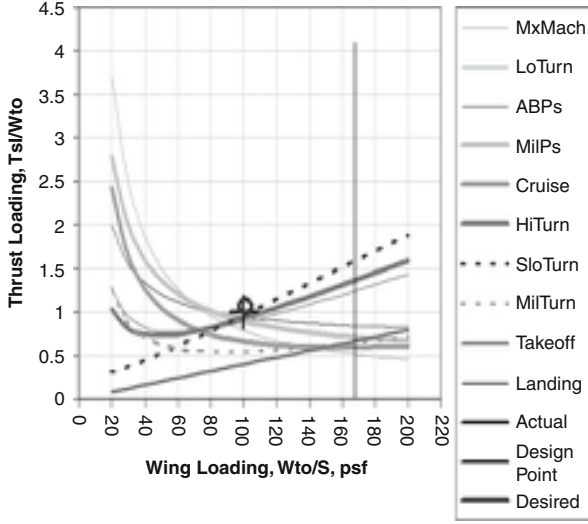


Fig. 10.2 Typical constraint diagram

10.4 Mission Analysis

Mission analysis uses the aircraft's drag polar and the design point determined from constraint analysis, plus an appropriate model for engine thrust-specific fuel consumption, to predict the amount of fuel the aircraft will burn on its design mission(s). The master equation for mission analysis is essentially the time integral form of the constraint analysis master equation substituting actual thrust for thrust available and multiplying by thrust-specific fuel consumption, c , to yield fuel flow rate:

$$cT = -\dot{W}_f$$

The minus sign in front of fuel flow rate indicates that aircraft weight decreases as fuel is burned. The integral of fuel flow rate yields fuel burn which, divided by weight, is the fuel fraction:

$$\begin{aligned} \int_{t_0}^{t_1} \frac{cT}{W_{TO}} dt &= \int_{t_0}^{t_1} \frac{-\dot{W}_f}{W_{TO}} dt \\ &= \int_{t_0}^{t_1} c q \left[\frac{C_{D_o}}{(W_{TO}/S)} + k_1 \left(\frac{n\beta}{q} \right)^2 \left(\frac{W_{TO}}{S} \right) + k_2 \left(\frac{n\beta}{q} \right) \right] dt \\ &\quad + \int_{t_0}^{t_1} c\beta \left[\frac{1}{V} \frac{dh}{dt} + \frac{1}{g} \frac{dV}{dt} \right] dt \end{aligned}$$

which is easily integrated for the most common cases of constant velocity climbs and cruise, constant altitude accelerations, and for relatively short mission legs where it is acceptable to assume the aircraft's drag polar and weight fraction are approximately constant:

$$\begin{aligned} \frac{W_0 - W_1}{W_{TO}} &= \frac{\Delta W_f}{W_{TO}} = \\ &= c_q \left[\frac{C_{D_0}}{(W_{TO}/S)} + k_1 \left(\frac{n\beta}{q} \right)^2 \left(\frac{W_{TO}}{S} \right) + k_2 \left(\frac{n\beta}{q} \right) \right] (t_1 - t_0) \\ &\quad + c\beta \left[\frac{1}{V} (h_1 - h_0) + \frac{1}{g} (V - V_0) \right] \end{aligned} \quad (10.4)$$

Equation (10.4) is known as the master equation for mission analysis. Note that W_{TO}/S in (10.4) is, for this analysis, a constant, since it was chosen as a result of constraint analysis. The assumption of constant β is not acceptable for long cruise or loiter legs, and the assumption of constant drag polar is not acceptable for transonic accelerations, but these are easily accommodated in computer implementations of the method by breaking long legs into many similar short legs.

When the mission analysis master equation is applied to each leg of a typical design mission, the fuel fractions from all preceding legs of the mission are used to determine the weight fraction for each subsequent leg. When the fuel fractions for each leg are summed, the total fuel fraction for the mission is obtained:

$$\begin{aligned} \beta_{takeoff} &= 1 \quad (\text{since } W = W_{TO}) \\ \beta_{accel} &= \beta_{takeoff} - \frac{W_{f_{takeoff}}}{W_{TO}} \\ \beta_{c\ lim\ b} &= \beta_{accel} - \frac{W_{f_{accel}}}{W_{TO}} \\ \beta_{cruise} &= \beta_{c\ lim\ b} - \frac{W_{f_{climb}}}{W_{TO}} \\ \beta_{loiter} &= \beta_{cruise} - \frac{W_{f_{cruise}}}{W_{TO}} \\ \frac{W_{f_{mission}}}{W_{TO}} &= \frac{W_{f_{takeoff}}}{W_{TO}} + \frac{W_{f_{accel}}}{W_{TO}} + \frac{W_{f_{climb}}}{W_{TO}} + \frac{W_{f_{cruise}}}{W_{TO}} + \frac{W_{f_{loiter}}}{W_{TO}} \end{aligned}$$

$W_{f_{mission}}/W_{TO}$ is known as the design mission fuel fraction. It is multiplied by the design point wing loading to yield the value of $W_{f_{mission}}/S$ needed to solve the sizing equation:

$$\frac{W_{f_{mission}}}{S} = \frac{W_{f_{mission}}}{W_{TO}} \frac{W_{TO}}{S}$$

10.5 Weight Analysis

Once mission analysis has predicted the mission fuel burn and fuel fraction, weight-estimating methods are used to predict the structural weights per unit area of the airframe components. For a small UAV, it is often adequate to weigh a similar UAV airframe to obtain approximate weights per area for the various airframe components. Whether predicted or measured, the component weights per unit area are then scaled by the ratios of their component reference areas to the UAV reference planform area and summed to obtain the airframe wing loading portion needed to solve the sizing equation. Once a value of S is determined, it may be multiplied by the design point wing loading to obtain the sized takeoff gross weight:

$$W_{TO} = S \left(\frac{W_{TO}}{S} \right) \quad (10.5)$$

The results obtained in (10.1) and (10.5) are final, provided some other size or volume constraints, such as the dimensions of the payload, do not force redesign of the aircraft configuration. The entire aircraft is just scaled up or down as necessary to achieve the specified wing area.

It is more common, however, that the fuselage of a small UAV is sized by payload dimensions, so that when the aircraft is sized up or down, the fuselage size remains unchanged. In this common situation, after resizing, the entire analysis must be accomplished again. This is because resizing the wing and tail without resizing the fuselage changes the aircraft's drag polar. Since drag polar is one of the required inputs to constraint analysis, constraint analysis must be reaccomplished, yielding new values for wing loading and thrust loading. A change in the configuration also changes the value of the airframe wing loading portion, and a new solution to the sizing equation results. The process must be repeated iteratively until a converged solution is obtained. Convergence may require human intervention, since the numerical scheme is not necessarily stable. In addition, the savvy UAV designer will explore variations of the configuration to find the optimum design, that is, that design which can meet all the customer's requirements for the lowest cost.

10.6 Electric Aircraft Sizing and Optimization

The use of electric power in small UAVs has increased significantly as cost and weight of batteries have decreased. The advantages of batteries over chemical fuels for UAV storage, transportation, and deployment are obvious. As battery performance continues to improve, a larger proportion of UAVs will likely use electric power.

10.6.1 Sizing Equation

The sizing methodology for electric-powered aircraft is based on a similar analysis for solar-powered aircraft described in Brandt and Gilliam (1994). As in the method for fueled aircraft, the central equation in this analysis is a simple statement that the sum of the weights of the various components of the aircraft must equal the aircraft total weight.

When divided by the wing area, the equation expresses the equality between the wing loading of the aircraft and the sum of the portions of that wing loading due to each of the components:

$$\frac{W_{Takeoff}}{S} = \frac{W_{Airframe}}{S} + \frac{W_{Battery}}{S} + \frac{W_{Motor \& Prop}}{S} + \frac{W_{Payload}}{S}$$

The strategy in using this equation to size and optimize an electric aircraft is identical to that used to obtain Eq.(10.1) for fueled aircraft and the resulting equation is similar:

$$S = \frac{W_{payload}}{\frac{W_{TO}}{S} - \frac{W_{aircraft}}{S} - \frac{W_{battery}}{S} - \frac{W_{motor}}{S}} \quad (10.6)$$

As for fueled aircraft, total wing loading is usually chosen in order to allow the aircraft to meet performance requirements specified by the customer, commonly represented on a constraint diagram such as Fig. 10.3. The details of constraint analysis for electric aircraft are identical to those for fueled aircraft, except that the thrust lapse, α , for electric aircraft behaves somewhat differently. From a constraint diagram the designer selects values for wing loading and thrust or power loading which will allow the aircraft to meet the customer's needs. In the case of Fig. 10.2, the selected values are $W_{TO}/S = 1.1$ and $T/W_{TO} = 0.21$.

The thrust values graphed in the constraint diagram are specified as those at static sea level conditions. Wind tunnel tests of the motor and propeller may be used to correct the thrust required for each of the constraints to sea level static values for plotting.

10.6.2 Airframe Wing Loading Portion

As with fueled aircraft, determining a value for the airframe wing loading portion involves determining representative values for similar aircraft and estimating what is achievable with current and future structures and materials technology. Since weights of aircraft components typically vary with their areas, this airframe wing loading portion, once calculated for a given aircraft configuration, will not vary significantly as the aircraft is sized up or down. A nominal value for airframe wing loading portion for a conventional configuration small UAV constructed of carbon fiber and epoxy composite skin over a Styrofoam core is 0.2 lb/ft².

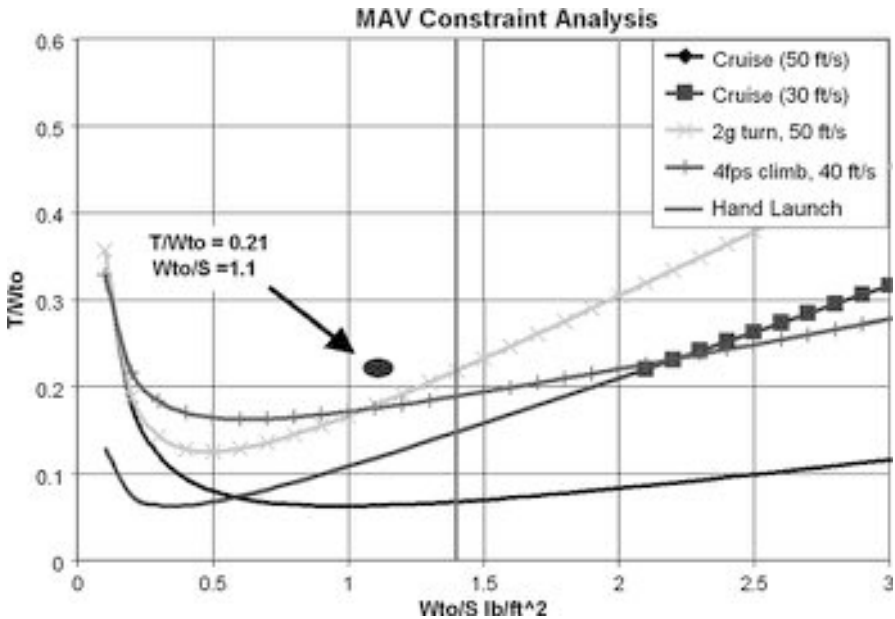


Fig. 10.3 Example small UAV constraint diagram

10.6.3 Battery Wing Loading Portion

The weight of the battery depends on the level of battery technology, represented by its energy density, e , the efficiency of the motor and propeller, η_{motor} and η_{prop} , and the total thrust energy which the motor and propeller must provide to overcome the aircraft’s drag and to increase the aircraft’s mechanical energy by climbing or accelerating. The total thrust energy required will therefore depend on the details of the design mission profile. Figure 10.4 illustrates the mission profile for a small backpack UAV.

For this mission, the only change in energy occurs immediately after launch, when the aircraft climbs 500 ft. above its launch altitude. Throughout the mission, the aircraft will fly at an approximately constant speed, and since its weight will not change, neither will its drag. As a result, the total thrust energy required for the mission is given by

$$E = W_{TO}\Delta h + DVt \tag{10.7}$$

where Δh is the change in altitude during the initial climb, D is the drag throughout the mission, V is the velocity, and t is the total mission duration. The required weight of the battery is the total energy which must be provided by the battery divided by the battery’s energy density. This energy provided by the battery must be enough so that after losses in the motor and propeller, modeled by efficiency factors for these devices, there is useable thrust energy equal to that required by (10.7).

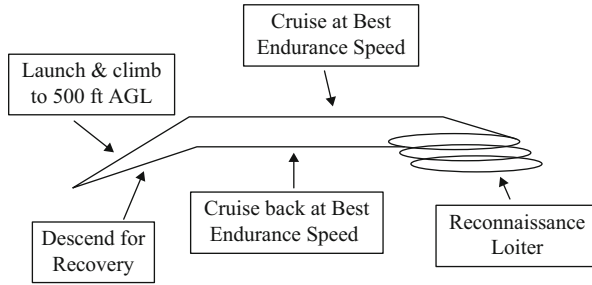


Fig. 10.4 Backpack UAV general mission profile

If, as is the case in the mission profile of Fig. 10.3, the energy to climb is small compared to the total energy required for the mission, then the weight of the battery required is approximated by

$$W_{\text{Battery}} = \frac{DVt}{e\eta_{\text{Motor}}\eta_{\text{prop}}} \quad (10.8)$$

Dividing (10.8) by the aircraft weight, W_{TO} , which is equal to the required lift, L , yields

$$\frac{W_{\text{Battery}}}{W_{TO}} = \frac{Vt}{\frac{L}{D} e\eta_{\text{Motor}}\eta_{\text{prop}}} \quad (10.9)$$

Note that the ratio L/D in (10.9) is a common means of describing the aerodynamic efficiency of an aircraft and is a function of the aircraft configuration and flight conditions. For the simple mission profile being considered, L/D will be essentially constant throughout the mission. Multiplying (10.9) by the total wing loading chosen from constraint analysis yields

$$\frac{W_{\text{Battery}}}{S} = \frac{W_{\text{Battery}}}{W_{\text{Takeoff}}} * \frac{W_{\text{Takeoff}}}{S} = \frac{W_{\text{Takeoff}}}{S} \frac{Vt}{\frac{L}{D} e\eta_{\text{Motor}}\eta_{\text{prop}}} \quad (10.10)$$

10.6.4 Motor and Prop Wing Loading Portion

Finally, the motor and propeller wing loading portion is estimated using the thrust loading and wing loading obtained from constraint analysis and an estimate of available motor and propeller technology expressed as a motor and prop thrust-to-weight ratio:

$$\frac{W_{\text{Motor}}}{S} = \frac{T}{W_{\text{Takeoff}}} * \frac{W_{\text{Motor}}}{T} * \frac{W_{\text{Takeoff}}}{S} \quad (10.11)$$

The motor and prop thrust-to-weight ratio, along with values for η_{motor} and η_{prop} , may be obtained by testing several typical motor/prop combinations in a low-speed wind tunnel. Figure 10.5 illustrates the results of a typical test. It plots the variation of motor and propeller efficiency for the best motor/propeller combination tested, as a function of free-stream velocity.

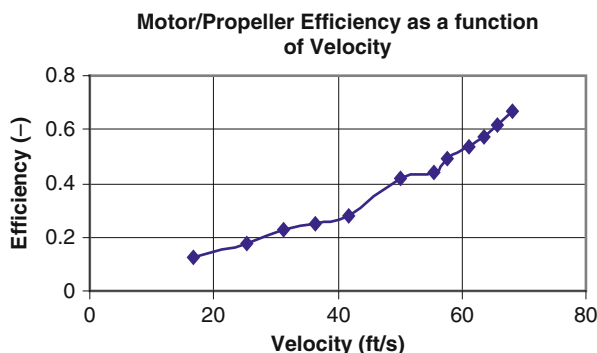


Fig. 10.5 Variation of efficiency with forward velocity for typical small UAV electric motor and propeller

10.6.5 Solving the Sizing Equation

With all terms evaluated, Eq. (10.6) is solved for planform area. As with fueled aircraft, this result is then used in Eq. (10.5) to solve for sized takeoff gross weight. The results are final provided payload volume or some other size constraint does not force a change to the configuration. In the common case where the fuselage must remain a constant size while the wing is sized up or down, the resulting new configuration will require an updated drag polar and weight model. This in turn will require a complete new iteration through the sizing method. Frequently, several iterations are required to reach a final solution. This numerical scheme may be unstable, so human intervention may be required to enforce convergence. In general, adjusting the wing area to an average between that modeled in the previous iteration and that recommended by the new solution to the sizing equation will lead to faster convergence.

In the extreme case, the analysis may not converge. This typically happens when specified mission duration is too long. When this occurs, radical changes to the configuration or radical new technologies like increased battery energy density may be required to meet the requirements.

10.6.6 Optimization

The configuration modeling, aerodynamic analysis, constraint analysis, and sizing cycle just described can be repeated for a variety of aircraft configurations. If configuration parameters such as wing sweep, aspect ratio, airfoil thickness, and camber are varied systematically, an optimization carpet plot such as Fig. 10.6 can be obtained. From such a plot, an optimum wing planform or other configuration characteristic can be selected, which minimizes the aircraft size while still meeting all customer requirements.

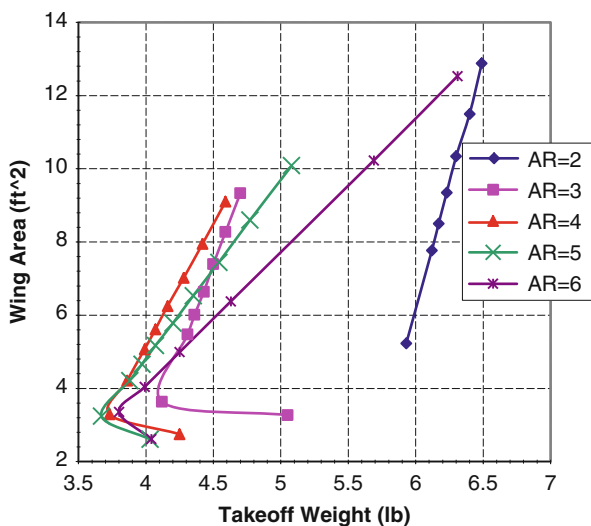


Fig. 10.6 Typical wing optimization carpet plot

10.6.7 Example: The Red Falcon

Results of a recent UAV design effort at the United States Air Force Academy (USFA) will illustrate how the method may be applied to a typical UAV design problem. The example design problem required a UAV small enough to fit in a backpack and when deployed able to carry a small spy camera, as depicted in Fig. 10.7. This camera, with its built-in downlink transmitter and its power supply weighed approximately 100 g or slightly more than 3 ounces. The design team selected rechargeable lithium-ion batteries, with an energy density of 68 W-h/lb, as their power source. The team measured motor and propeller efficiencies during wind tunnel testing. They also performed wind tunnel tests of several candidate aircraft configurations including pusher, canard, and flying wing variations but in the end settled on a conventional tractor propeller, tail aft configuration with an unswept wing with an aspect ratio (AR) of 4. Figures 10.8 and 10.9 illustrate the wing sweep and aspect ratio optimization plots for this configuration. In the end, an aspect ratio of 4 was selected for ease of packaging in the backpack and because the decrease in size for higher aspect ratios was minimal.

The above design decisions led to an aircraft configuration which sized to a wing area of slightly over 1 ft² and a maximum design flying weight of slightly over 1 lb. The aircraft, named the “Red Falcon,” is shown in Fig. 10.10. This aircraft flew a number of flight test missions to verify performance predictions and flying qualities. Although it never went into production, it provided a sizing example for a number of subsequent UAV developments. It served to verify the validity of the methodology described in this chapter and the practicality of using a small UAV for this design mission.

Fig. 10.7 Daylight spy camera and downlink

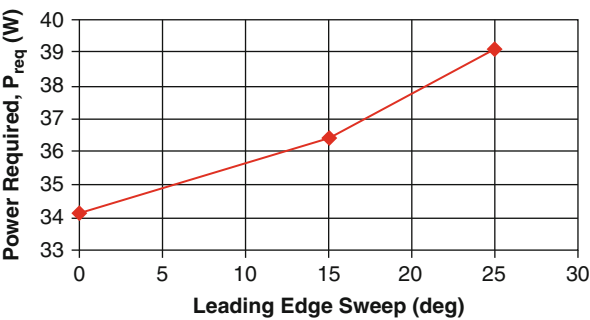


Fig. 10.8 Wing sweep optimization plot

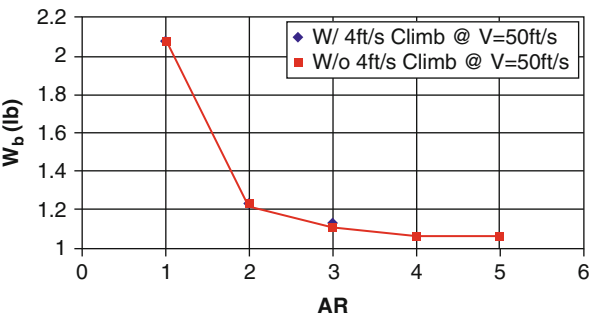
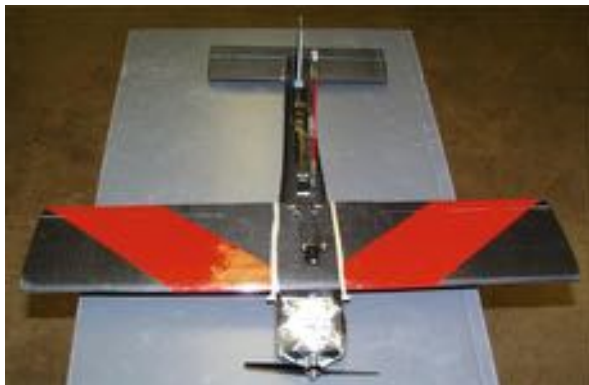


Fig. 10.9 Wing aspect ratio optimization plot

Fig. 10.10 The Red Falcon



10.7 Conclusion

This chapter has presented a rationale for the UAV designer to choose an aircraft configuration and a methodology for sizing it. By iteratively changing such aircraft configuration parameters as wing aspect ratio and camber, the designer can determine the smallest/cheapest UAV that can meet the design mission. This process is one way to optimize UAV designs for maximum performance and minimum size/cost.

References

- S.A. Brandt, F.P. Gilliam, A design analysis methodology for solar-powered aircraft. *J. Aircr.* 766–780 (1994)
- S.A. Brandt et al., *Introduction to Aeronautics: A Design Perspective*. AIAA Education Series, 2nd edn. (AIAA, Reston, 2004)
- J.R. Smith, P.G. Batish, S.A. Brandt, S.R. Morton, A student-developed sizing methodology for electric-powered aircraft applied to small UAVs, in *WAC Paper 2000-01-5536, Proceedings of the 5th AIAA/SAE World Aviation Congress*, San Diego, 10–12 Oct 2000

Systematic Design Methodology and Construction of Micro Aerial Quadrotor Vehicles

11

Swee King Phang, Kun Li, Ben M. Chen, and Tong H. Lee

Contents

| | | |
|--------|------------------------------------|-----|
| 11.1 | Introduction | 182 |
| 11.2 | Virtual Design Environment | 183 |
| 11.2.1 | MSC Patran and Nastran | 184 |
| 11.2.2 | SolidWorks | 188 |
| 11.3 | Hardware Design | 190 |
| 11.3.1 | Motor and Propeller | 190 |
| 11.3.2 | Electronics | 191 |
| 11.3.3 | Power Supply | 193 |
| 11.3.4 | Avionics Design | 193 |
| 11.4 | Dynamic Modeling and Control | 194 |
| 11.4.1 | Nonlinear Mathematics Model | 194 |
| 11.4.2 | Parameters Identification | 198 |
| 11.4.3 | Controller Design | 200 |
| 11.5 | Flight Test Results | 200 |
| 11.6 | Conclusions | 202 |
| | References | 205 |

Abstract

This chapter presents a guideline to systematically design and construct an ultralight micro aerial quadrotor vehicle, capable of autonomous flight. The guideline details the steps to design a stable quadrotor with not more than 50 g takeoff weight while having a flight duration of 8 min. Optimization was done to all mechanical parts of the vehicle with the constraints of its weight and

S.K. Phang • K. Li

Control and Simulation Lab, National University of Singapore, Singapore

e-mail: king@nus.edu.sg; kunli89@nus.edu.sg

B.M. Chen • T.H. Lee

Control and Simulation Lab, Department of Electrical and Computer Engineering, National University of Singapore, Singapore

e-mail: bmchen@nus.edu.sg; eleleeth@nus.edu.sg

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_116,

© Springer Science+Business Media Dordrecht 2015

181

its performance. The guideline first covers the structural analysis of the micro quadrotor air frame, followed by its design in a 3D virtual simulator. Then, design of avionic system such as processor and sensors will be discussed, followed by controller implementation with the aid of software simulation. Components of the micro quadrotor are then fabricated and implemented based on the optimum results obtained from various simulations. Finally, the feasibility of the constructed aircraft is tested and confirmed with real flight missions.

11.1 Introduction

With more requirements imposed on unmanned aerial vehicle (UAV) for military tasks of surveillance, reconnaissance, and detecting in obstacle-rich areas, clandestine military bases, radiant areas, and other dangerous regions, the desire and interest in research of micro aerial vehicle (MAV), or even nano aerial vehicle (NAV), are aroused (Wang et al. 2011). These requirements involve higher intelligence for simultaneous localization and mapping (SLAM), path planning, and obstacle avoidance, along with battery technology to maintain long endurance flight and communication with the ground control station (Achtelik et al. 2009; Eresen et al. 2012; Meier et al. 2011; Phang et al. 2010). In the recent few decades, the development of integrated circuits and micro-electromechanical systems (MEMS) triggered the emergence of smaller and lighter electronics and mechanical systems. It provides the possibility to shrink the size and weight of the UAV to a new milestone.

In 1997, the Defense Advanced Research Projects Agency (DARPA) sets an official definition for MAV, which requires maximum dimension of the aircraft in any direction to be no greater than 15 cm and that the gross takeoff weight should not exceed 100 g, with up to 20 g devoted to payload. Based on these requirements, many UAV research groups initiated their MAV study with different design approaches, including fixed-wing, rotary-wing, flapping-wing, and other unconventional platforms (Al-Qadi et al. 2006; Michelson 2010; Petricca et al. 2011). One good example would be the Black Widow, an 80 g fixed-wing MAV developed by AeroVironment, which has a 15 cm wingspan and is claimed to be fully autonomous (Grasmeyer and Keennon 2011). It is, however, difficult for the fixed-wing platform to show its capability in indoor environments or cluttered outdoor environments such as forest and urban environment full of building and other obstacles, due to its limitation of hovering and vertically taking off and landing (VTOL) ability. Flapping-wing is a promising platform with its VTOL and hovering capabilities like that of an insect or a bird. In 2007, a 3 cm size flapping-wing MAV was developed by Harvard University, biologically mimicking the dipteran insects (Wood 2007). Unfortunately, there is no mature development of autonomous flapping-wing platforms due to the sophisticated flapping mechanism and wing structure aerodynamic analysis. Ducted-fan platform is also a possible approach for miniature aerial vehicle. A 150 g ducted-fan platform was developed by University of Bologna. This platform requires extra fins to control its attitudes, resulting in a large takeoff weight (Naldi et al. 2010).

Recently, due to its extreme stability and mechanical simplicity, quadrotor platforms have aroused the research interests of many researchers. There are, however, not many quadrotor MAV platforms lower than 50 g capable of autonomous flight. A research group from Stanford University has developed a micro quadrotor “Mesicopter”, with an impressive weight of 3 g. It is, however, unable to take off due to its rotor efficiency problem (Kroo and Kuns 2001). Mellinger and his collaborators from GRASP lab have been working on the formation flight of quadrotor MAVs weighing 60 g each and have achieved impressive results (Mellinger et al. 2010, 2012). From the perspective of the platform approach, rotary-wing platforms, especially in the form of quadrotor, have the advantages in maneuverability and mechanical feasibility.

In this chapter, a guideline to design a micro aerial quadrotor vehicle, or in short quadrotor MAV, with the following specifications will be presented:

1. Target weight not higher than 50 g
2. Largest dimension smaller than 15 cm in any direction
3. Flight duration of 8 min

Systematic design procedures with scopes of the design blueprint in virtual 3D environment, functionalities analysis in simulations, hardware selection and avionic system design, electrical circuit and software debugging, mathematical modeling and parameters identification, model simulation and verification, as well as aircraft orientation control with real flight test will be discussed in detail. In regard to this work, great effort was made to investigate the feasibility of overall structure and to evaluate the platform from the aspects of weight, size, and power consumption with both experiments and simulations before manufacturing, which guarantees a flyable and stable platform.

Main contents of this chapter are divided into five sections. Section 11.2 is the virtual design of the overall platform, with respect to mechanical structure and natural frequency analysis. Section 11.3 describes the hardware selection and avionic design procedures. Section 11.4 is the nonlinear mathematics model derivation concerning the aspects of kinematics, rigid body dynamics, and motor dynamics, followed by parameters identification methods and controller design. In Sect. 11.5, real flight test data is used to verify the mathematical model and control laws.

11.2 Virtual Design Environment

Before constructing the quadrotor platform, the design was done in virtual design environments. Differing from the usual UAV in larger scale, the natural modes of the structure or platform play an important role in determining the stability of the quadrotor MAV. As the aircraft gets smaller and lighter, the effect of its vibration due to natural modes is disastrous. Therefore in this section, two design processes in developing small-scale quadrotor MAV are proposed to systematically determine the optimum shape for the platform, in terms of lightweight and natural mode avoidance.

The first process is to perform finite element analysis (FEA) on various platform shapes. FEA is a numerical approach in solving structural problems by discretizing

the problem where the domain to be analyzed is divided into a number of discrete elements. Using this approach, it offers great flexibility to model complex geometries which would be near impossible if analytical approach is taken. By undergoing FEA, the natural modes and frequencies of the platform can be estimated numerically. With this, a best possible candidate can be chosen from among the designs.

Upon obtaining an optimum structural shape of the platform according to the FEA results, the platform of the quadrotor MAV can then be designed with a 3D simulation software, where each individual part of the aircraft could be assembled in the software for an overall design. An advantage of utilizing simulation software is that it could estimate various important parameters of the designed aircraft, such as the weight, density, and the moment of inertia, in a way that the design could be customized freely in order to meet their requirements before they are fabricated to real physical objects.

11.2.1 MSC Patran and Nastran

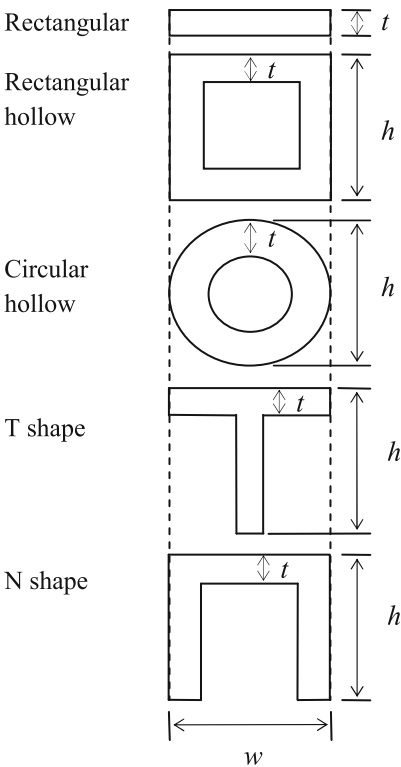
MSC Nastran, one of the most widely used FEA commercial software, is utilized in simulating the vibration properties of the quadrotor MAV. It is a useful tool which simulates static and dynamic cases for a wide variety of complicated structural problems. In addition to Nastran, a modeler software called Patran is employed for finite element modeling. Patran provides geometry modeling, meshing, boundary condition, and material properties set up for Nastran. It is also used for postprocessing purpose. Provided in Nastran, Lanczos (1950) method is utilized for eigenvalue and eigenvector extraction purpose.

To investigate and design the quadrotor platform, single arm and full quadrotor models are constructed in Patran with the properties of carbon fiber composite material assigned to the software. The model is then fetched to Nastran to investigate its structural resonance. In corresponding specifically to the design of micro aerial quadrotor, the study is concentrated on natural mode analysis to determine the natural frequencies and mode shapes of the model. The natural mode analysis predicts the resonance for the structure and the type of resonance. On the other hand, the displacement response of the model can be determined using frequency response analysis when external steady-state oscillatory excitation (simulating rotor rotation) is applied.

11.2.1.1 Single Quadrotor Arm

A quadrotor consists of four extended arms attached to a body which holds the onboard electronics. The aim of carrying out the single quadrotor arm analysis is to provide a useful summary on the performance of different quadrotor arm shapes, and then among the few possible candidates under the constrained of weight and size, a best solution in terms of shape, length, and dimension can be obtained. In this analysis, the quadrotor arm is approximated as a cantilever beam which has a fixed end and a free end.

Fig. 11.1 Cross sections of five common beams



Five different types of cantilever beams are designed and analyzed. The cross sections for each of the beams are shown in Fig. 11.1. A few different parameters on the dimension of the beams will be varied, and the corresponding natural mode (1–4) will be compared. Note that the composite material for all beams to be simulated has the properties of carbon fiber code name carbon/epoxy T300/976. All simulations are done in Nastran by employing the discrete element with six degrees of freedom (DOF) per node, and all nodes at one of the tips are assumed to be fixed by setting all six DOF to zero.

The first variable to be investigated is the length of the beam. It is well known that slender bodies are more easily exposed to vibration, or in other words, shorter beams are stiffer. To verify the relationship between the length of the beam and its natural modes, the rectangular beam (first cross section in Fig. 11.1) with different length is analyzed in the simulation. In Table 11.1, a summary of the Nastran natural mode analysis results obtained for this study is given. Based on the results, it is evident that natural frequencies for the first four modes increase as the structure becomes shorter. In general, beams of other shape show similar behavior, and thus the results are trivial and not to be included here. Despite the results favoring shorter beams, there are other restrictions on the minimum length of the quadrotor arms. One important factor would be the aerodynamic interferences between the rotors, which generally limits the minimum length of the quadrotor arms to be at least twice the rotor radius.

Table 11.1 Natural frequencies of thin plate with varying length

| Length (mm) | Natural frequency (Hz) | | | |
|-------------|------------------------|----------|----------|----------|
| | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| 50 | 614.6 | 3, 656.7 | 3, 803.4 | 10, 518 |
| 60 | 426.85 | 2, 546.1 | 2, 642.6 | 7, 313.4 |
| 70 | 313.62 | 1, 873.6 | 1, 942.1 | 5, 377.1 |
| 80 | 240.3 | 1, 437.1 | 1, 491.3 | 4, 137.1 |
| 90 | 189.73 | 1, 135.4 | 1, 175.3 | 3, 255.4 |
| 100 | 153.68 | 920.14 | 952.05 | 2, 637.5 |

Table 11.2 Natural frequencies of beam with different cross sections and thickness

| Type of cross section | 1 mm thickness | | 0.5 mm thickness | |
|-----------------------|----------------|-------------|------------------|-------------|
| | Weight (g) | Mode 1 (Hz) | Weight (g) | Mode 1 (Hz) |
| Rectangular | 0.5328 | 426.85 | 0.2664 | 213.45 |
| Rectangular hollow | 1.776 | 3,032.4 | 0.9768 | 3,270.7 |
| Circular hollow | 1.3949 | 2,647.9 | 0.767 | 2,851.6 |
| T shape | 0.9768 | 1,902.8 | 0.5106 | 1,845.3 |
| N shape | 1.4208 | 2,717.5 | 0.7548 | 2,810.4 |

The second variable to be investigated is the thickness of the material. As two common thickness of carbon fiber sheet or beam available commercially are of 0.5 and 1 mm thick, the natural mode of all five general shapes of beams are analyzed and compared in 0.5 and 1 mm thickness. For fair comparison, the model for each cross section was constructed with length $l = 60$ mm, width $w = 6$ mm, and height $h = 6$ mm (see Fig. 11.1 for illustration). Table 11.2 shows the comparison between beam with thickness $t = 1$ mm and $t = 0.5$ mm, together with their calculated weight. One can notice that in general, the beam having a closed cross-sectional shape, i.e., the rectangular hollow and circular hollow beams, has a much higher natural frequency compared to other shapes. Also, the thickness of the material has little effect on them, while the weight could be half as light.

The last variable to be investigated is the width and height of these closed cross-sectional beams. It can be seen from Table 11.3 that the height variation of the rectangular hollow beam affects the first mode frequency more severely than the second mode frequency. On the other hand, the width variation of the beam affects the second mode frequency much more than the first mode frequency. In conclusion, utilizing a square beam with equal height and width will give similar frequencies for the first and second modes. This is further proven in Table 11.4 where the circular hollow beams were analyzed.

In general, although the rectangular solid and T-shape configurations show better weight budget, their natural frequencies are comparatively much lower than that of rectangular hollow and circular hollow shapes, which are in the closed cross section forms. In the following section, a full quadrotor configuration with rectangular hollow beam of 3 mm height and 3 mm width will be investigated.

Table 11.3 Natural frequencies of the rectangular hollow beam with different width and height (0.5 mm thickness)

| Height (mm) | Width (mm) | Natural frequency (Hz) | | | |
|-------------|------------|------------------------|---------|---------|--------|
| | | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| 3 | 5 | 1,625.9 | 2,515.8 | 9,673.4 | 14,904 |
| 3 | 6 | 1,651.8 | 2,979.7 | 9,718.5 | 17,415 |
| 3 | 7 | 1,670.3 | 3,430.8 | 9,708.0 | 19,762 |
| 3 | 8 | 1,683.5 | 3,871.3 | 9,656.9 | 21,954 |
| 3 | 9 | 1,693.0 | 4,302.7 | 9,576.0 | 23,571 |
| 3 | 10 | 1,699.4 | 4,725.6 | 9,472.4 | 22,983 |
| 4 | 6 | 2,218.0 | 3,105.6 | 12,845 | 17,908 |
| 5 | 6 | 2,756.0 | 3,199.4 | 15,699 | 18,185 |
| 6 | 6 | 3,270.7 | 3,270.7 | 18,312 | 18,312 |

Table 11.4 Natural frequencies of the circular hollow beam with different radius (0.5 mm thickness)

| Radius (mm) | Natural frequency (Hz) | | | |
|-------------|------------------------|---------|--------|--------|
| | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| 2.0 | 1,839.3 | 1,839.3 | 11,070 | 11,070 |
| 2.5 | 2,347.5 | 2,347.5 | 13,875 | 13,875 |
| 3.0 | 2,851.6 | 2,851.6 | 16,495 | 16,495 |
| 3.5 | 3,349.6 | 3,349.6 | 18,911 | 18,911 |
| 4.0 | 3,840.1 | 3,840.1 | 21,117 | 21,117 |
| 4.5 | 4,322 | 4,322 | 23,116 | 23,116 |
| 5.0 | 4,794.3 | 4,794.3 | 24,918 | 24,918 |
| 5.5 | 5,256.1 | 5,256.1 | 26,538 | 26,538 |

11.2.1.2 Full Quadrotor Configuration

Due to structure stability shown by closed shape beams, the full configuration is then tested using rectangular hollow beam and circular hollow beam to form the arms attached to each corner of the main frame. The main frame is designed to be 28.28×28.28 mm with thickness 2 mm. Meanwhile, 60 mm long beam is employed as the quadrotor arm. Again, unidirectional carbon/epoxy T300/976 is used. In Patran, the main frame is modeled as 2-D shell element while the quadrotor arm using 1-D beam.

For this analysis, the main frame is assumed to be rigid and thus fixed in all three translation DOF. For a quadrotor model with 3 mm width, 3 mm height, and 1 mm thickness rectangular hollow beam, the resulted natural frequency for the first and second modes are 971.92 and 5,843.3 Hz. On the other hand, the natural frequency for quadrotor model using circular hollow beam with outer diameter of 3 mm and thickness of 1 mm is at 940.35 Hz for the first mode and 5,070.7 Hz for second mode.

Subsequently, dynamic analysis is performed to investigate the response of the quadrotor to oscillatory excitation produced by the propeller blades. In this simulation, a 0.25 N oscillatory force is applied to the tip of each quadrotor arms. The analysis is performed over frequency range of 0–4,000 Hz. Results obtained from the Nastran are displayed in Fig. 11.2a, b. From the response spectrum, the frequency at peak for both quadrotor models matches the first natural mode

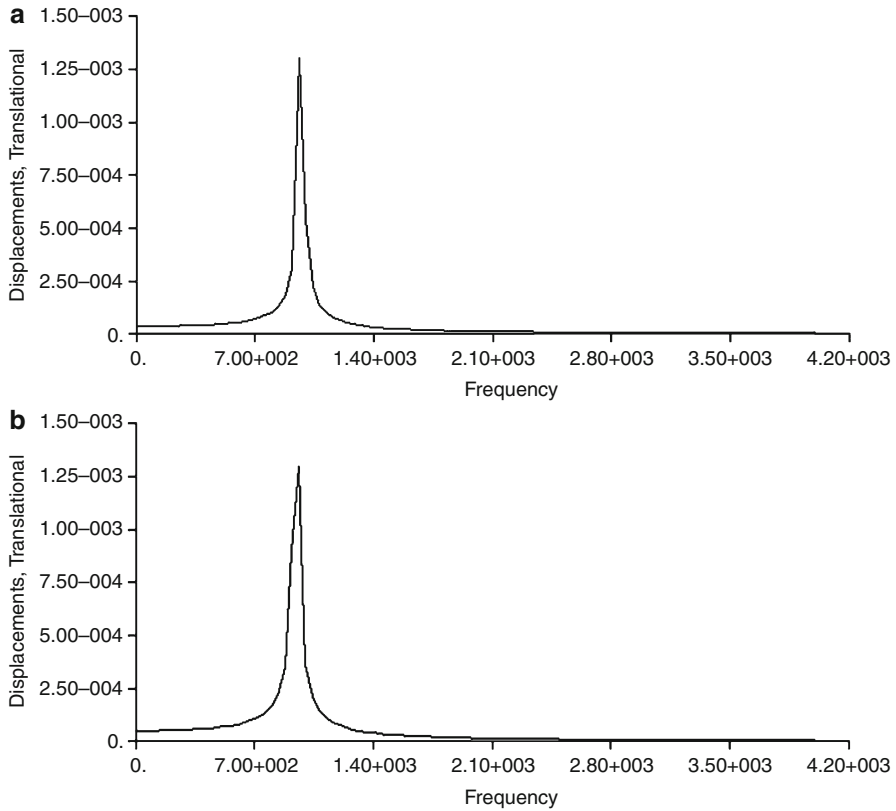


Fig. 11.2 Displacement at tip of the arm with rectangular (a) and circular (b) hollow beam

simulated earlier. Based on the simulation result, both models are suitable for miniature quadrotor design as they have the highest stiffness to weight ratio among the tested shapes. In conclusion, using a beam with a closed cross-sectional shape as the quadrotor arm can guarantee extreme stability for the whole quadrotor model.

11.2.2 SolidWorks

An overall design blueprint of this quadrotor platform is essential to provide a broad overview of the quadrotor functionalities, size, weight, and appearance design. A 3D mechanical design software named SolidWorks, developed for efficient and quicker design of mechanical products and components, facilitates the design tasks for the platform. This 3D software is chosen as the design and analyzing software over other mechanical design tools due to a few advantages, such as increase design efficiency, ease of fabricating, and easy access to simulated geometric data.

Table 11.5 Weight budget for quadrotor MAV

| Components | Amount | Estimate weight (g) | Current weight (g) |
|---------------------|--------|---------------------|--------------------|
| Battery | 1 | 10 | 9.8 |
| Motor and propeller | 4 | 3.5 | 3.59 |
| Quadrotor arm | 4 | 1 | 0.93 |
| Quadrotor frame | 1 | 4 | 2.13 |
| Onboard system | 1 | 10 | 7.32 |
| Miscellaneous | | 2 | 1.7 |
| Total | | 44 | 39.03 |

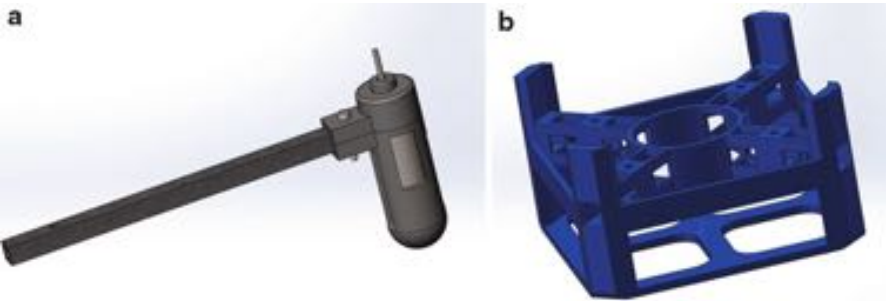


Fig. 11.3 Quadrotor arm and frame design in SolidWorks

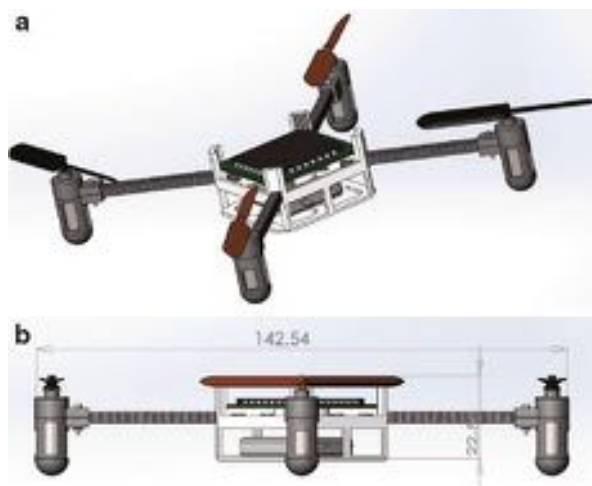
Based on the natural mode analysis in Nastran (see previous section), a cross-shaped frame structure is designed with four rectangular hollow carbon fiber beams fixed into a holder locating the avionic system along with the battery. The design takes the following considerations:

1. The CG should be located on the z -axis at the geometrical center of the cross shape.
2. Mechanical parts should be carefully designed with the weight and moment of inertia symmetrically distributed along the x - and y -axis.
3. Materials with low density and high intensity are to be used as the frame and protection to satisfy weight limit.

In order to accurately approximate the weight and dimension of the overall design, components are to be drawn on scale, with the correct weight and density assigned to it in SolidWorks. As weight is the main constraint to the design of the MAV, the design parts are minimized. Table 11.5 details the total parts in the design of the quadrotor MAV, with specific weight budget assigned to each of them. On the right of the table listed the real weight of the components or designed parts, which will be detailed here and also in the next sections.

- The structure design can be separated into two different parts:
1. *Quadrotor arms*: A protection scheme is developed as a shell to encapsulate the motor into a tight chamber to fix its position. Carbon fiber beam can be mounted and screwed to the side, as shown in Fig. 11.3a.

Fig. 11.4 Mechanical structure layout and size of the quadrotor



2. *Main body*: The main frame structure has four slots for the quadrotor arms and contains two layers, where the avionic system is placed on the top and the battery is located in the lower level, as shown in Fig. 11.3b. It is fabricated with acrylonitrile butadiene styrene (ABS).

The mechanical structure layout of the proposed quadrotor MAV is shown in Fig. 11.4a with all the parts assembled together as shown in Fig. 11.4b. Distance of two diagonal rotors is 142.54 mm, with a total height of 22.50 mm. Details of weight breakdown are reflected in Table 11.5. In this table, the estimated weight is approximated based on the design guideline shown in this chapter, while the current weight on the right of the table is the measured weight of the quadrotor MAV prototype, code name KayLion, made by National University of Singapore, following the guideline.

11.3 Hardware Design

Hardware system and the avionics are the core of any unmanned aircraft design. The hardware system includes the motors and propellers, avionics such as onboard processor (CPU), sensors like inertial measurement unit (IMU), and power supply. Suitable choices of the hardware and components will be proposed based on the requirement on their performance under the constraint on weight and size.

11.3.1 Motor and Propeller

Motor and propeller sets are the main actuators of the quadrotor MAV. As each quadrotor consists of four sets of motor and propeller, they need to be chosen carefully as their characteristics must satisfy the design requirements. A few

important design requirements of the quadrotor MAV are directly related to the characteristic of the motor and propeller. They are the operation voltage and current consumption of the motors, the weight of the motors, and the maximum thrust it can produce with different propellers.

Based on the requirements stated above, a 8,000kV single cell brushed direct current (DC) motor is utilized. Combined with two sets of propeller, each has a clockwise and an anticlockwise propellers; the total weight of a single motor and propeller is approximately 4 g. Test bench experiment has proved that the combination could produce a maximum thrust of 16 g each, which combined to be approximately 1.5 times larger than the proposed MAV at 40 g.

11.3.2 Electronics

The avionic system consists of electronic components for the electrical system of the quadrotor MAV. Different from the conventional design of the avionic system for larger UAV, where each component can be chosen from the commercially off-the-shelf (COTS) product, with firm packaging and plug-and-play connection ready, the choices of electronic components for the quadrotor MAV need to be made down to integrated chip (IC) level. A printed circuit board (PCB) incorporated all onboard electronics will then be designed.

11.3.2.1 Control CPU

The processor is the core of the avionic system. In the compact MAV design, the main tasks of the processor are (1) collecting data from IMU sensor from serial port, (2) receiving command from the receiver in the form of pulse position modulation (PPM) signal, (3) decoding and analyzing data, (4) realizing flight control laws, (5) sending control signals in the form of pulse width modulation (PWM) signal to the electronic speed controllers (ESC), and (6) sending logging data to the logger via serial port.

For the selection of the microprocessor for the abovementioned tasks, ATmega328P, a high-performance Atmel 8-bit AVR RISC-based microcontroller, is adopted because of the following features: (1) availability of various ports, such as UART, SPI, and PWM ports; (2) low power consumption; and (3) ease of coding.

It is proved that the ATmega328P microprocessor is good enough to run the software for the tasks mentioned above within 5 ms. As a result, the control loop running in the software is user configurable up to 200 Hz, depending on the update frequency of the IMU sensor.

11.3.2.2 Inertial Measurement Unit

An IMU is an essential sensor to any autonomous aircraft. It provides important motion measurements of the body it attached to, such as accelerations, angular rates, and magnetic values. It is noted that for most of the aircraft orientation control, 3-axis Euler angle measurements must be provided. However, it is not necessary that all IMU provides the angular measurements as they can be estimated by using an extended Kalman filter (EKF) (Jang and Liccario 2007) or complimentary

filtering (Yun et al. 2007). These filters are, however, computationally intensive; thus, they might be a burden for the onboard AVR microprocessor. Here, a solution is to find a powerful IMU with in-built EKF algorithm in a small and light package.

The VN-100 SMD from VectorNAV is chosen in the design. It is a lightweight (3 g) and miniature ($24 \times 22 \times 3$ mm) high-performance IMU and Attitude Heading Reference System (AHRS). In this tiny package, it includes 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetic sensor. An extra advantage of VN-100 is the in-built 32-bit microprocessor to compute and output a real-time and drift-free 3D orientation solution with EKF algorithm. The IMU output rate is user-configurable from 40 to 200 Hz, depending on the output methods.

11.3.2.3 Motor Speed Controller

ESC is essential for each brushed motor used in the MAV design. The main contribution of ESC is to convert the commonly used PWM signal to analog signal to be fed to the motor. The current of the analog signal will be further boosted to drive the motor. For the ESC design, a single 8-pin processor ATtiny13A is utilized for speed controlling. The processor is responsible for decoding PWM signals fed from the control CPU (ATmega328P) and outputting a stable analog voltage. The current output of the analog port is, however, too low to drive the motor. Thus a single MOSFET chip is also included to feed the current from the aircraft's power supply (battery) directly to the motor. Four ESCs and MOSFETs are needed for the quadrotor MAV design.

11.3.2.4 Radio-Frequency Receiver

Conventionally, a radio-frequency (RF) receiver is used to receive and decode RF signals sent from a transmitter controlled by a ground pilot in radio-controlled (RC) flights. In general, a receiver is not needed in a fully autonomous flight control system as no remote pilot is needed. Most of the UAV designs, however, retain the receiver component for fail-safe purposes, where a ground pilot has higher authority to remotely control the UAV during emergencies such as controller failures. In the quadrotor MAV design, the receiver has a different function. Besides receiving the control signals from the remote pilot, the receiver is utilized to receive control signals or measurement values from the ground control station in autonomous mode. It is particularly useful when the MAV system is navigating in indoor environment with the aid of VICON motion technology, where the VICON system measures the position and velocity of the MAV, then sends the control signals or the measurement values to the aircraft's onboard CPU via the transmitter to receiver link. This communication system could be realized with a PCTx cable, a product by Endurance R/C. The PCTx cable connects the ground station (desktop or laptop) to the transmitter, making use of the transmitter to send RF signals to the onboard receiver. Upon receiving signals from the ground, the receiver will then output the signals in the form of PPM signal to the onboard CPU for processing. In order to fulfill the mentioned requirement, a good candidate would be Rx31d from DelTang. It is chosen to be integrated in the avionic system due to its ultra tiny package of 10×10 mm with 0.21 g. It is able to provide up to seven channels of PPM signals.

11.3.2.5 Data Logger

In most UAV designs, important flight data such as state variables of the UAV model will be recorded for post-flight observation and investigation. In the design of the MAV, a logger is required to be small, light, and simple enough to work. An open source data logger, called OpenLog from SparkFun, is utilized in the design. OpenLog is presented in a tiny package with 1.70 g. It will start logging any serial data up to 115,200 baud rate to the micro SD card upon powered up. Optionally, as SparkFun provides OpenLog firmware and design files, it can be redesigned to the main PCB of the MAV.

11.3.3 Power Supply

The main consideration in designing the power supply is to meet the overall system and flight duration requirements. The choice of power supply is important as it constitutes most (approx. 30 %) of the overall weight of the MAV, and the power needed to lift the MAV will be increased due to its own weight. As all onboard components can be powered up with 3.3 V, a single cell Lithium-Polymer (Li-Po) battery with current capacity of 360 mAh is utilized to power the avionics and to drive the motors. A 3.3 V regulator is included to provide a clean voltage to the components, as single cell Li-Po battery's output varied from 4.2 V during full charged down to lower than 3.4 V when it is used up. The battery is as light as 10 g and is able to provide enough energy for an 8-min flight duration.

11.3.4 Avionics Design

In this subsection, the design process of the PCB for the avionic system of the quadrotor MAV will be described in detail. Among the five components to be included to the avionic system, the IMU, flight control CPU, and four ESCs will be incorporated into the design, while the receiver and the logger will be attached to the designed PCB. A general guideline to design avionics PCB for quadrotor MAV with Altium Designer is as follows:

1. *Schematic design* – A schematic diagram of the design must be drawn in Altium Designer with all the components needed, including four status indication LEDs. A 3.3 V voltage regulator is also included.
2. *Layout assignment* – The layout of the components of the PCB is important so as to reduce the electromagnetic interference between the components. To satisfy the dimension and weight constraints, a maximum of 4×4 cm PCB layout is imposed. The IMU must be placed in the middle of the design to be as closed to the CG as possible with the correct orientation. Then, components such as flight control CPU and motor speed controllers are placed at the opposite side of the board. Lastly, the four LEDs are located in a way that they are clearly visible to the user during flight.
3. *Routing* – The final step of designing PCB is the routing to connect each component according to the connection assigned in the schematic phase. The routing could be easily done (see Fig. 11.5) in a 2-layer-PCB setup.

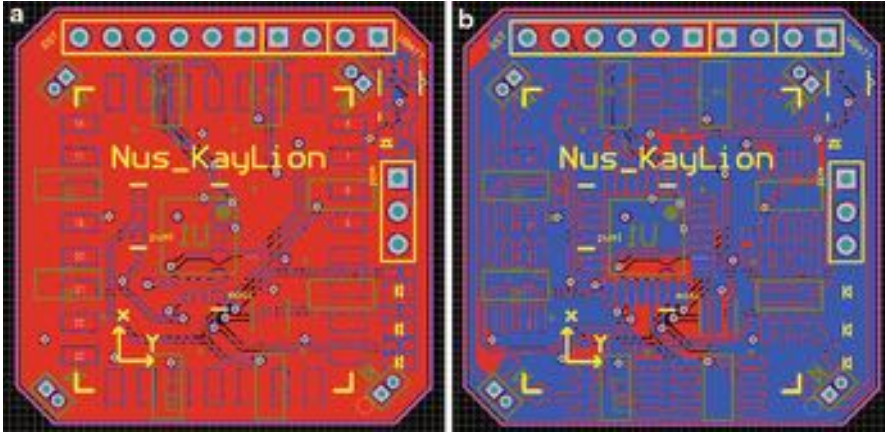


Fig. 11.5 PCB layout

Once the design is done, it can be sent to PCB manufacturer to fabricate. With PCB thickness of 1 mm, the fabricated product is approximately 7 g including all components, within the dimension of $40 \times 40 \times 1$ mm.

11.4 Dynamic Modeling and Control

In this section, a nonlinear mathematic model of the quadrotor MAV is derived. The methods of parameters identification will also be discussed, which include direct measurement, test bench experiments, as well as evaluation in virtual environment. This mathematic model is further verified with the real flight test data obtained by the VICON motion capture system. PID control law is first implemented in the virtual model in Simulink, then further fine-tuned on the real platform based on the flight tests.

11.4.1 Nonlinear Mathematics Model

The mathematical models of quadrotor aircraft are quite well developed in the literature (Bouabdallah et al. 2004; Erginer and Altug 2007; Guerrero-Castellanos et al. 2011; Kim et al. 2007; Phang et al. 2012; Pounds et al. 2006). The overall structure view of this quadrotor platform is pictured in Fig. 11.6, where δ_{ail} , δ_{ele} , δ_{thr} , and δ_{rud} represent the normalized input signals from aileron, elevator, throttle, and rudder channels, respectively. δ_1 , δ_2 , δ_3 , and δ_4 are the normalized input values to each motors. Ω_1 , Ω_2 , Ω_3 , and Ω_4 are motors' individual rotational speeds. Linear velocity u , v , w and angular velocity p , q , r can be obtained by 6 degree-of-freedom (DOF) rigid body dynamics, and position in ground frame x , y , z and Euler angles ϕ , θ , ψ can be calculated through kinematics equation.

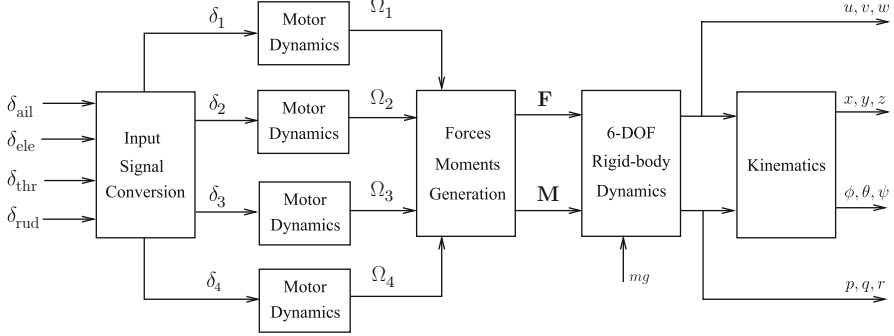


Fig. 11.6 Overall structure of the quadrotor

Two coordinate frames, north-east-down (NED) frame $[x_i \ y_i \ z_i]$ and the body frame $[x_b \ y_b \ z_b]$, will be considered (Wang et al. 2012). The NED frame is stationary with respect to a static observer on the ground, and the body frame is the coordinate frame with the origin located at the CG and orientation moving together with the aircraft fuselage. The origin of the body frame is located at the CG of the platform with x -axis parallel with rotor 2 and rotor 3 along with y -axis parallel with rotor 1 and rotor 2. Euler angles $\Theta = [\phi \ \theta \ \psi]^T$ are the angles rotating about x -, y -, and z -axis, describing the roll, pitch, and yaw motions. T_1 , T_2 , T_3 , and T_4 are the lift force created by rotor 1, 2, 3, and 4. Rotors 1 and 3 rotate clockwise, while rotors 2 and 4 rotate counterclockwise.

11.4.1.1 Kinematics and 6 Degree-of-Freedom Rigid Body Dynamics

At any time instant, the aircraft can be pictured as having a translational and rotational motion with respect to the stationary NED (or ground) frame. The following kinematics equations represent the navigation equations applicable to transformation between the two frames:

$$\dot{\mathbf{P}}_n = \mathbf{R}_{n/b} \mathbf{V}_b, \quad (11.1)$$

$$\dot{\Theta} = \mathbf{S}^{-1} \Omega, \quad (11.2)$$

where $\mathbf{R}_{n/b}$ represents the transformation matrix and \mathbf{S}^{-1} represents a lumped transformation matrix. They are given by:

$$\mathbf{R}_{n/b} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \quad \mathbf{S}^{-1} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix}, \quad (11.3)$$

with $s_* = \sin(*)$, $c_* = \cos(*)$, $t_* = \tan(*)$.

Based on Newton-Euler formalism describing the translational and rotational dynamics of a rigid body, the dynamic equations can be written into the following input-output form:

$$m\dot{\mathbf{V}}_{\mathbf{b}} + \Omega \times (m\mathbf{V}_{\mathbf{b}}) = \mathbf{F}, \quad (11.4)$$

$$\mathbf{J}\dot{\Omega} + \Omega \times (\mathbf{J}\Omega) = \mathbf{M}, \quad (11.5)$$

where \mathbf{F} and \mathbf{M} are the force and moment vectors, m is the mass of aircraft, and \mathbf{J} is the tensor of inertia matrix.

11.4.1.2 Forces and Moments Generation

Based on the working principle of the quadrotor (Cai et al. 2011), the forces and torques are mainly generated by the four rotors (Goel et al. 2009). The following equation represents the component part of the overall force and moment vector:

$$\Lambda = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_g + \mathbf{F}_m \\ \mathbf{M}_m + \mathbf{M}_{gy} + \mathbf{M}_{rt} \end{bmatrix}, \quad (11.6)$$

where subscripts g , m , gy , and rt correspond to forces and moments generated by gravity, rotors, gyroscopic effect, and reactional torque, respectively.

As the platform is four-way symmetric, the CG is located on the z -axis, thereby the gravitational force only contributes to the force vector. Considering the coordinate frames, the gravity only exists on the z -axis in NED frame and needs to be transformed to the body frame by the transformation matrix:

$$\mathbf{F}_g = \mathbf{R}_{\mathbf{n}/\mathbf{b}}^{-1} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \begin{bmatrix} -mg s_{\theta} \\ mg c_{\theta} s_{\phi} \\ mg c_{\theta} c_{\phi} \end{bmatrix}. \quad (11.7)$$

It is the motor and propeller pairs which produce the main movements to generate the forces and moments. As in Pounds et al. (2004), let T_i and Q_i be the thrust and torque created by i -th rotors ($i = 1, 2, 3, 4$); they can be expressed as below:

$$T_i = C_T \rho A (\Omega_i R)^2, \quad (11.8)$$

$$Q_i = C_Q \rho A (\Omega_i R)^2 R, \quad (11.9)$$

where C_T and C_Q are the propeller aerodynamic coefficient, ρ is the air density, A is the area of the propeller swept by the rotating rotor, and R is the radius of the rotor A . Assuming that the distortion of the propellers during high frequency rotation can be ignored, Eqs. (11.8) and (11.9) can be simplified as:

$$T_i = k_T \Omega_i^2, \quad (11.10)$$

$$Q_i = k_Q \Omega_i^2, \quad (11.11)$$

where these two coefficients k_T and k_Q can be obtained by a series of experiments. Thereby, the sum of these four thrusts will result in a total thrust in negative z -axis in the body frame, as below:

$$\mathbf{F}_m = \begin{bmatrix} 0 \\ 0 \\ -(T_1 + T_2 + T_3 + T_4) \end{bmatrix}. \quad (11.12)$$

The moments are generated when the four thrusts have different magnitudes, results in pitch, roll, and yaw movements, as shown below:

$$\mathbf{M}_m = \begin{bmatrix} \frac{\sqrt{2}}{2}l(T_2 + T_3 - T_1 - T_4) \\ \frac{\sqrt{2}}{2}l(T_1 + T_2 - T_3 - T_4) \\ Q_1 - Q_2 + Q_3 - Q_4 \end{bmatrix}, \quad (11.13)$$

where l is the distance from the center of the motor to the platform CG.

Gyroscopic effect is caused by the combinations of rotations of four propellers and can be modeled as

$$M_{gy} = \sum_{i=1}^4 J_r \left(\Omega \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) (-1)^i \Omega_i \quad (11.14)$$

$$= \begin{bmatrix} -J_r q(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) \\ J_r p(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) \\ 0 \end{bmatrix} \quad (11.15)$$

where J_r is the moment inertia of the rotor, including the propeller and motor shaft. The gyroscopic effects produced by the propellers are only related to the moments of the aircraft.

The first-order derivative of rotational movement can cause the inertia counter torque, and this torque only affects the movement in yaw direction, which is

$$\mathbf{M}_{rt} = \begin{bmatrix} 0 \\ 0 \\ -J_r(\dot{\Omega}_1 - \dot{\Omega}_2 + \dot{\Omega}_3 - \dot{\Omega}_4) \end{bmatrix}. \quad (11.16)$$

11.4.1.3 Motor Dynamics

Each brushed motor is controlled by a single cell brushed ESC, and the steady-state value of rotor angular speed to brushed ESC input can be approximated as a linear

process near the equilibrium (hovering) value. The transient property of this brushed motor can be approximated as a first-order process, as shown below:

$$\dot{\Omega}_i = \frac{1}{T_m} [K_m(\delta_i - \delta_i^*) - \Omega_i], \quad (11.17)$$

where δ_i is the normalized input to the i -th motor, T_m and K_m are the time constant and process gain of the first-order model, and δ_i^* is the offset to the normalized input values. Note that the offset values of each motor are determined in the hovering condition.

11.4.2 Parameters Identification

Several parameters need to be identified to obtain the complete model. Here, several methods of identifying each parameters will be presented. They are all to be done by direct measurement, experiments, or computer simulation.

11.4.2.1 Direct Measurement

Parameters that can be directly measured by a weighing balance and ruler are mass of each components and the length of quadrotor arms, as follows:

$$m = 0.032 \text{ kg}, \quad (11.18)$$

$$L = 0.058 \text{ m}. \quad (11.19)$$

Gravitational acceleration, depends on the location, can be easily calculated given the latitude of the location. In Singapore in particular,

$$g \approx 9.781 \text{ m/s}^2. \quad (11.20)$$

11.4.2.2 Computer Simulation

Several parameters can be estimated numerically with computer software. As mentioned in the previous section, mechanical parts of the quadrotor MAV are first designed in SolidWorks, with exact density and scale to the real physical parts. A few parameters, such as the tensor moment of inertia of the quadrotor MAV and the rotating moment of inertia of the propeller, are calculated with the mass property function of SolidWorks:

$$J = \begin{bmatrix} 3.0738 & 0 & 0 \\ 0 & 3.0849 & 0 \\ 0 & 0 & 5.9680 \end{bmatrix} \times 10^{-5} \text{ kg m}^2, \quad (11.21)$$

$$J_r = 5.897 \times 10^{-8} \text{ kg m}^2. \quad (11.22)$$

Note that J is diagonal as the designed quadrotor structure is highly symmetric.

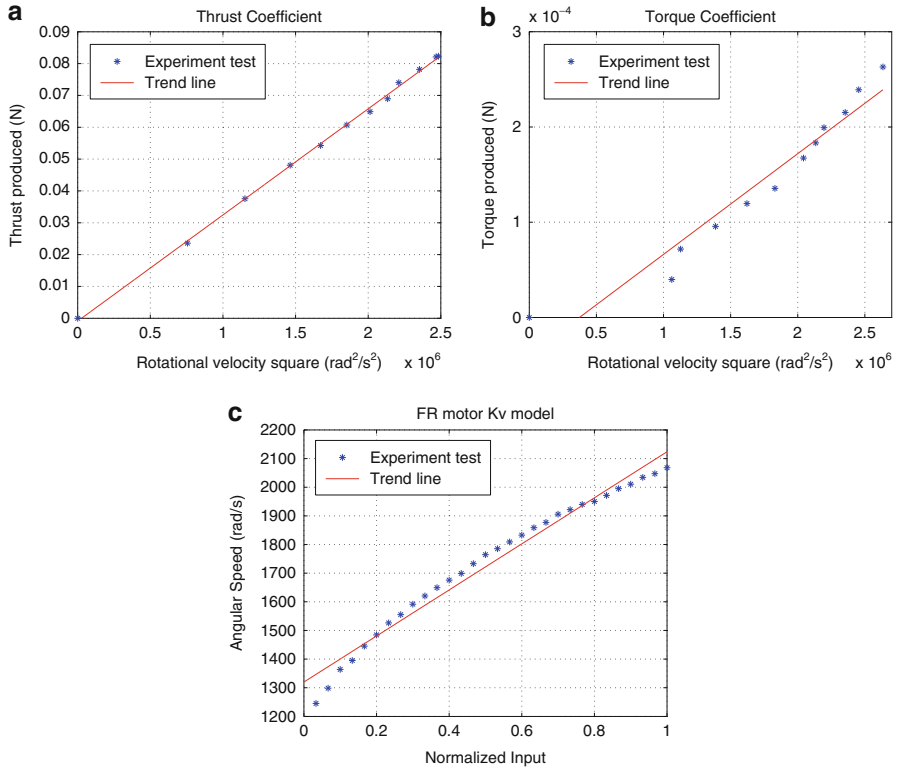


Fig. 11.7 Measurement of thrust coefficient, torque coefficient, and motor's steady-state gain

11.4.2.3 Test Bench Experiment

The thrust and torque coefficient of the rotor can be measured by lever-balance setup. An infrared transceiver is used to measure the time interval between two adjacent cutting of the propeller, where the real-time angular velocities can be calculated. Thrust and torque produced by the rotor can be measured from the weighing scale. The results are plotted in Fig. 11.7a, b. Thrust and torque coefficients k_T and k_Q can then be obtained as the gradient of the approximate line:

$$k_T = 3.334 \times 10^{-8} \text{ N}/(\text{rad}^2/\text{s}^2), \quad (11.23)$$

$$k_Q = 1.058 \times 10^{-10} \text{ Nm}/(\text{rad}^2/\text{s}^2). \quad (11.24)$$

The next experiment is conducted to estimate the brushed motor dynamics. Several different values of constant input are fed to the motor, and the resultant angular speeds of the motors are recorded with the similar setup as the experiment above. The results are plotted in Fig. 11.7c. Here the motor steady-state velocity is

assumed to be proportional to the input value at equilibrium (during hover). Thus, the steady-state gain, K_m , can be extracted as

$$K_m = 803.9. \quad (11.25)$$

Also, as presented in Eq. 11.17, the motor dynamics can be approximated as a first-order process:

$$\frac{\Omega_i(s)}{R_i(s)} = \frac{K_m}{T_m s + 1}, \quad (11.26)$$

where average time constant T_m is estimated from the transient response of a step input to the motor. The value obtained is

$$T_m = 0.0821 \text{ s}. \quad (11.27)$$

11.4.3 Controller Design

Upon obtaining the mathematical model of the aircraft, a simple yet reliable PID controller is designed and simulated with the aid of Simulink in MATLAB.

Generally, the dynamics of a quadrotor without orientation stabilizer is too fast even for a skilled human pilot. The fast dynamics is contributed by the roll and pitch movements, while the yaw movement exhibits much slower dynamics.

In the initial stage of the PID gains tuning, the controller is first designed in simulation by utilizing the Ziegler-Nichols method. Simulated result in Fig. 11.8 has shown a stable quadrotor system, with all the Euler angles, and angular rate signals are attenuated to zero within 5 s. This set of PID gains is further fine-tuned with real flight tests. Finally, a set of PID gains for each axis in which the aircraft is able to stabilize horizontally on this setup was obtained, as shown in Table 11.6.

11.5 Flight Test Results

The PID controller designed with software simulator shown above is implemented to the manufactured quadrotor MAV code-named KayLion (Fig. 11.9). Flight tests are then carried out to test the endurance of the vehicle and to verify the mathematical model derived in the previous section.

In particular, chirp-like oscillating inputs are sent to the MAV system, while its Euler angles and angular rates responses are recorded in the onboard logger. The recorded responses are then compared to the simulated responses by using

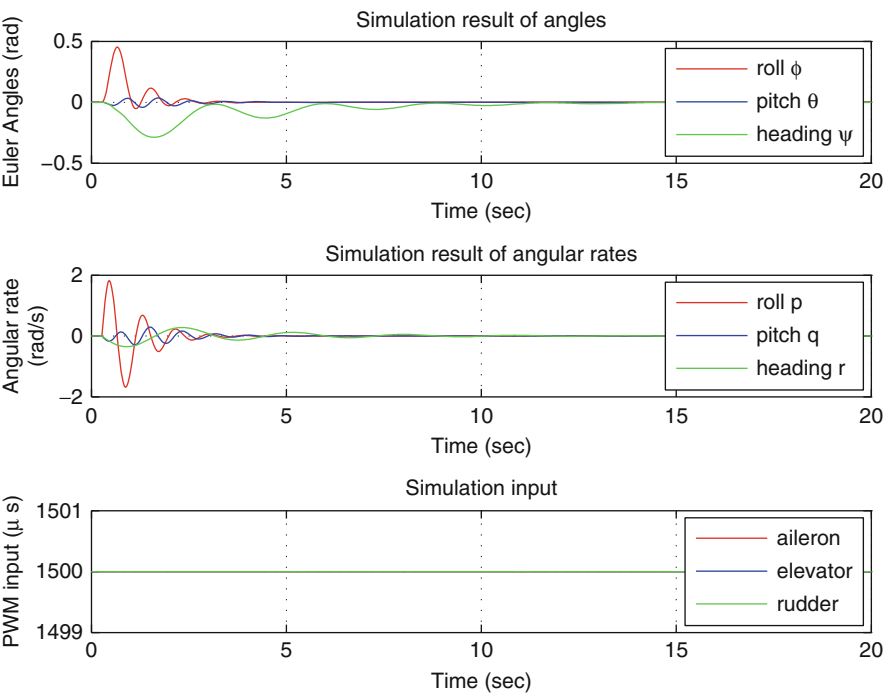


Fig. 11.8 Simulation results with PID gains

Table 11.6 PID gains implemented to the quadrotor MAV

| Channel | K_p | K_i | K_d |
|---------|-------|-------|-------|
| Pitch | 1.60 | 2.45 | 0.51 |
| Roll | 1.60 | 2.45 | 0.51 |
| Yaw | 4.00 | 0.75 | 0.10 |

the exact inputs to the simulator. Figure 11.10a shows the perturbation signal on elevator channel to the MAV, while Fig. 11.10b shows the responses of the system in pitch direction plotted together with the simulated results. Besides some random low-amplitude oscillations caused by the disturbances from the air movement, the responses match fairly well. Beside the pitch direction, the roll direction is assumed to be similar to the pitch, as the quadrotor MAV is of four-way symmetry.

In the other flight test, a chirp-like signal was injected to the throttle channel of the quadrotor MAV, resulting in agitated heave movement. In this experiment, a VICON Motion Tracking System is used to measure and compute the position of the quadrotor with reference to the start-up origin. Both the input signals and position measurement are logged and plotted in Fig. 11.11a, b. In the latter figure, it can be seen that the derived mathematical model on heave movement matches well with

Fig. 11.9 A full working prototype quadrotor MAV, KayLion, designed and built by the National University of Singapore (NUS)



the experimental data to a certain perturbation frequency, which is approximately 1 Hz. The quadrotor MAV was unable to respond to the perturbation signal above this frequency, as it was observed with naked eyes during the flight tests.

11.6 Conclusions

This chapter summarizes the steps to design and implement a micro scale quadrotor MAV, with a target take-off weight of less than 50 g. It has covered the full design areas, including the structural design, platform design, avionics design, and controller design, in which all areas are supported and verified by simulation or experimental results.

Structural analysis was first done in MSC Patran and Nastran to obtain suitable candidates for the platform design, where the shape of the beam with the highest stiffness against weight ratio was selected. The platform including the arms was then designed in SolidWorks to ensure proper placement of components. Next, the avionic system design was discussed, first with the selection of suitable electronics and constraints to the trade-off between weight and performance, followed by a detailed guideline to integrate these components to a single PCB. Once the platform was fabricated, a nonlinear mathematical model of the quadrotor MAV was derived. PID controller was implemented to the quadrotor, and flight tests were done to fine-tune the controller.

The flight test results have verified the design of the quadrotor MAV, from structural development to the avionic system implementation, and it has further proven the accuracy of the derived mathematical model. With the availability of the mathematical model for this quadrotor MAV, it serves as a good platform to test and verify control law design and implementation.

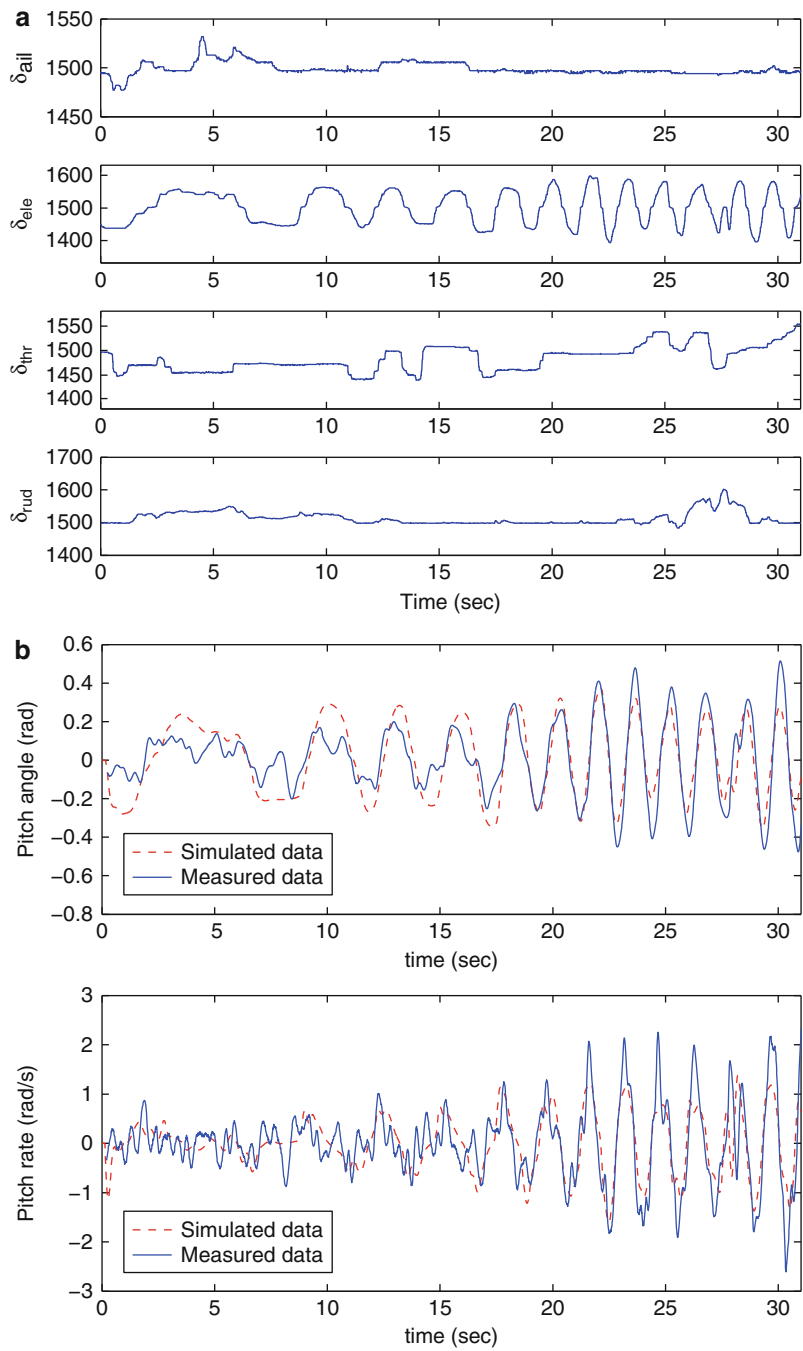


Fig. 11.10 Pitch angle and angular rate of the system response together with simulated response

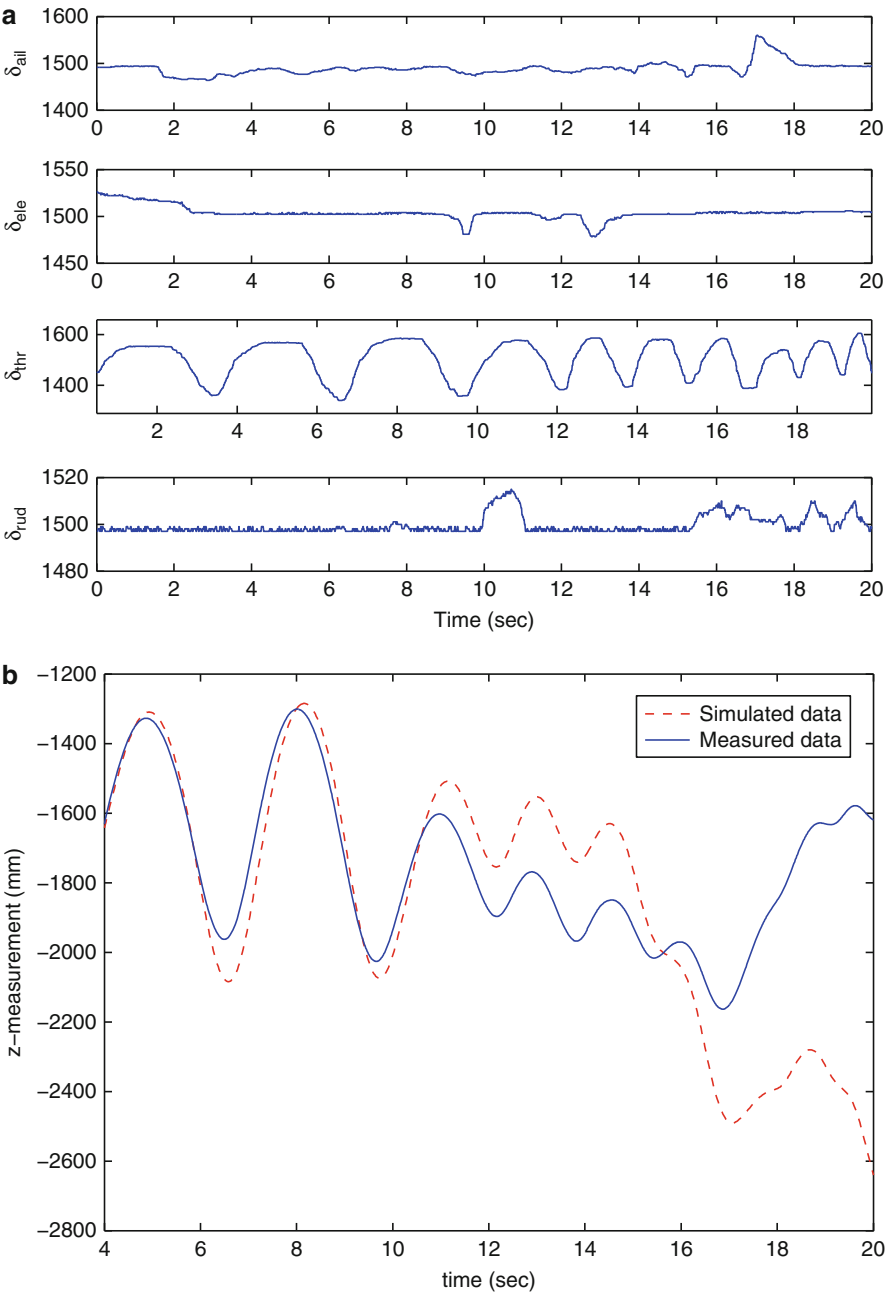


Fig. 11.11 z-axis position of the system response together with simulated response

References

- M. Achteik, A. Bachrach, R. He, S. Prentice, N. Roy, Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments, in *IEEE ICRA*, Kobe, 2009
- M.I. Al-Qadi, A.M. Al-Bahi, Micro aerial vehicles design challenges: state of the art review, in *SSAS UAV Scientific Meeting & Exhibition*, Jeddah, 2006
- S. Bouabdallah, A. Noth, R. Siegwart, PID vs LQR control techniques applied to an indoor micro quadrotor, in *IEEE International Conference on Intelligent Robots and Systems*, Sendai, 2004, pp. 2451–2456
- G. Cai, B.M. Chen, T.H. Lee, *Unmanned Rotorcraft Systems* (Springer, London/New York, 2011)
- A. Eresen, N. Imamoglu, M.O. Efe, Autonomous quadrotor flight with vision-based obstacle avoidance in virtual environment. *Expert Syst. Appl.* **39**(1), 894–905 (2012)
- B. Erginer, E. Altug, Modeling and PD control of a quadrotor VTOL vehicle, in *IEEE Intelligent Vehicles Symposium*, Istanbul, June 2007
- R. Goel, S.M. Shah, N.K. Gupta, N. Ananthkrishnan, Modeling, simulation and flight testing of an autonomous quadrotor, in *Proceedings of International Conference on Environmental and Agriculture Engineering*, Bangalore, 2009
- J.M. Grasmeyer, M.T. Keennon, Development of the black widow micro air vehicle. *Prog. Astronaut. Aeronaut.* **195**, 519–35 (2011)
- J.F. Guerrero-Castellanos, N. Marchand, A. Hably, S. Leseq, J. Delamare, Bounded attitude control of rigid bodies: real-time experimentation to a quadrotor mini-helicopter. *Control Eng. Pract.* **19**(8), 790–797 (2011)
- J.S. Jang, D. Liccardo, Small UAV automation using MEMS. *IEEE Aerosp. Electron. Syst. Mag.* **22**, 30–34 (2007)
- T.S. Kim, K. Stol, V. Kecman, Control of 3 DOF quadrotor model, in *Robot Motion and Control 2007*, ed. by K. Kozłowski (Springer, London, 2007), pp. 29–38
- I. Kroo, P. Kuns, Mesoscale flight and miniature rotorcraft development, in fixed and flapping wing aerodynamics for micro air vehicle applications, of progress in astronautics and aeronautics, **195**, pp. 503–517 (2001)
- C. Lanczos, An iteration method for the solution of the eigenvalue problem linear differential and integral operators. *J. Res. Natl. Bur. Stand.* **45**, 255–282 (1950)
- L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, PIXHAWK: a system for autonomous flight using onboard computer vision, in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011
- D. Mellinger, M. Shomin, V. Kumar, Control of quadrotors for robust perching and landing, in *International Powered Lift Conference*, Philadelphia, Oct 2010
- D. Mellinger, M. Nathan, V. Kumar, Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Int. J. Robot. Res.* **31**(5), 664–674 (2012)
- R.C. Michelson, Overview of micro air vehicle system design and integration issues, in *Encyclopedia of Aerospace Engineering*, ed. by R. Blockley, W. Shyy (Wiley, Chichester/Hoboken, 2010)
- R. Naldi, L. Gentili, L. Marconi, A. Sala, Design and experimental validation of a nonlinear control law for a ducted-fan miniature aerial vehicle. *Control engineering practice*, Special issue on aerial robotics, **18**(7), pp. 747–760 (2010)
- L. Petricca, P. Ohlckers, C. Grinde, Micro- and nano-air vehicles: state of the art. *Int. J. Aerosp. Eng.* **2011**, 1–17 (2011)
- S.K. Phang, J.J. Ong, R. Yeo, B.M. Chen, T.H. Lee, Autonomous mini-UAV for indoor flight with embedded on-board vision processing as navigation system, in *IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering (SIBIRCON)*, Irkutsk Listvyanka, 2010
- S.K. Phang, C. Cai, B.M. Chen, T.H. Lee, Design and mathematical modeling of a 4-standard-propeller (4SP) quadrotor, in *World Congress on Intelligent Control and Automation*, Beijing, 2012

- P. Pounds, R. Mahony, J. Gresham, P. Corke, J. Roberts, Towards dynamically-favourable quadrotor aerial robots, in *Proceedings of the Australian Conference on Robotics and Automation*, Canberra, 2004
- P. Pounds, R. Mahony, P. Corke, Modelling and control of a quad-rotor robot, in *Proceedings Australasian Conference on Robotics and Automation 2006*, Auckland, 2006
- F. Wang, S.K. Phang, J. Cui, B.M. Chen, T.H. Lee, Search and rescue: a UAV aiding approach, in *Canadian Congress of Applied Mechanics*, Vancouver, June 2011
- F. Wang, S.K. Phang, J. Cui, G. Cai, B.M. Chen, T.H. Lee, Nonlinear modeling of a miniature fixed-pitch coaxial UAV, in *American Control Conference*, Montreal, 27–29 June 2012
- R.J. Wood, Design, fabrication and analysis of a 3DOF, 3 cm flapping-wing MAV, in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, 29 Oct–2 Nov 2007
- B. Yun, K. Peng, B.M. Chen, Enhancement of GPS signals for automatic control of a UAV helicopter system, in *Proceedings of IEEE International Conference on Control and Automation*, Guangzhou, 2007, pp. 1185–1189

Guangying Jiang and Richard M. Voyles

Contents

| | | |
|------------|---|-----|
| 12.1 | Introduction | 208 |
| 12.1.1 | Force Closure | 209 |
| 12.1.2 | Aerial Mobile Manipulation | 210 |
| 12.1.3 | 3-D Translational Flight | 212 |
| 12.2 | Dexterous Hexrotor Theory | 213 |
| 12.2.1 | Dexterous Hexrotor Structure | 215 |
| 12.2.2 | Force Decomposition | 216 |
| 12.2.3 | Mapping from Actuator Space to Force/Torque Space | 219 |
| 12.2.4 | Dexterous Hexrotor Performance Optimization | 221 |
| 12.3 | Dexterous Hexrotor Control | 225 |
| 12.3.1 | Dexterous Hexrotor Control System | 225 |
| 12.3.2 | Hexrotor Attitude Controller | 227 |
| 12.4 | Dexterous Hexrotor Prototype | 228 |
| 12.5 | Experiments and Results | 231 |
| 12.5.1 | 6-DOF Force/Torque Test | 231 |
| 12.5.2 | Exert Forces Test | 232 |
| 12.6 | Conclusion | 236 |
| References | | 236 |

Abstract

Low-altitude flight usually introduces ground-effect disturbances and other backwash issues. In the new field of aerial mobile manipulation, it often includes close operations to structures for either inspection or manipulation of

G. Jiang (✉) • R.M. Voyles

Department of Electrical and Computer Engineering, University of Denver, Denver, CO, USA

e-mail: john.jiang@du.edu; richard.voyles@du.edu

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_130,

© Springer Science+Business Media Dordrecht 2015

207

the structures. Although there has been a fair amount of research work of free-flying satellites with graspers, the more recent trend has been to outfit UAVs with graspers to assist various manipulation tasks. While this recent work has yielded impressive results, it is hampered by a lack of appropriate test beds for aerial mobile manipulation, similar to the state of ground-based mobile manipulation a decade ago. Typical helicopters or quadrotors cannot instantaneously resist or apply an arbitrary force in the plane perpendicular to the rotor axis, which makes them inadequate for complex mobile manipulation tasks. Based on the concept of force closure (a term from the dexterous manipulation community), this chapter introduces a new type of dexterous, 6-DOF UAV which provides the unique capability of being able to resist any applied wrench, or generalized force/torque, providing more precise control during low-altitude flight.

12.1 Introduction

A multicopter is a UAV (unmanned aerial vehicle) that is lifted and manipulated by a number of rotors. Rotors of contemporary multicopters are mostly installed on the frame such that all their thrust vectors are parallel and vertical. Control of vehicle motion is achieved by alternating the pitch or roll rotation rates.

Numerous multicopter systems have been developed as custom research platforms, custom teaching platforms, toys, and commercially available systems. There even exist a number of open-source projects. Most of these multicopters have a number of fixed-pitch propellers with parallel and vertical thrust vectors.

Depending on how these N thrust actuators are physically arranged, various combinations of thrust magnitudes result in net force/torque vectors that span a subset of the 6-D Cartesian space of generalized forces. Coriolis forces induced by the spinning rotors can even be used to augment the net force vector on the multirotor body. Mathematically, a matrix can be constructed that provides a mapping from N -D actuator space to 6-D Cartesian force space, but this matrix can have rank no greater than four when the thrusters are all parallel. In fact, the standard quadrotor configuration results in rank of exactly four as the four thrusters provide linear force along Z axis, and torques around X and Y , while Coriolis effects provide torque around the Z axis. (This is what makes tri-rotors infeasible, and conventional single-rotor helicopters also have four actuators: main rotor, tail rotor, and two actuators on the swash plate.) Instantaneous exertion of linear forces along X and Y is impossible with these configurations; they can only lift, pitch, roll, and yaw. They cannot move laterally without first rolling or pitching an angle, so these systems are nonholonomic.

The multicopter system is dynamically unstable but allows for high maneuverability. Some attempts have been made to change this system like quadrotors, to increase controllability, stability, or maneuverability, but they still only have four motors. This instability lends itself to surveillance where quick movement is key and inability to maintain a stable pose is not an issue, but not much else can be done with these platforms.

Several hexrotor platforms have been developed including a miniature version from Airbotix and robust six- and eight-rotor versions from Aerobot and Dragonfly. The U.S. Air Force announced a solicitation for hexrotor platforms for military procurement. All these known hexrotor configurations employ parallel thrust vectors (identical to the quadrotors) which result in rank-4 actuator-to-Cartesian mappings incapable of exerting arbitrary forces in 6-D space. These six-rotor helicopters have been created simply for the increased lift of six rotors, redundancy, and smoother operation.

Manipulation is making a comeback in robotics. Through most of the 1990s and early 2000s, mobile robots dominated research and application, such as the DARPA grand challenge and autonomous helicopters. Now in kind of the mid-2000s and later up to now, there has been research in manipulation again. Manipulation is about agility, dexterity, and immobilizing a part when grasping.

In harkens back in manufacturing, there is an emerging field of mobile manipulation, combining the growth of mobile robots with robots people can manipulate (Holmberg and Khatib 2000). The field of mobile manipulation combines two broad classes of robots, locomotors and manipulators, yet the potential impact is greater than the sum of the parts. Aerial mobile manipulation is an emerging field within mobile manipulation for which the locomotor is a UAV (Stramigioli et al. 2012). The popular quadrotor has become the main UAV of choice in robotics research, due to its ease of control and low cost. The added mobility and access that quadrotors provide brings a new dimension to the study of mobile manipulation and new challenges as well.

One of the greatest challenges that UAVs, in general, introduce, and quadrotors in particular, is the inability to instantaneously exert forces in the plane. Quadrotors are nonholonomic; in order for them to move forward or sideways, they first have to pitch the entire body of the quadrotor to direct the thrust vector in the desired direction. What this means for aerial mobile manipulation is that the quadrotor cannot resist an arbitrary generalized force/torque. In the parlance of the dexterous manipulation community, it lacks “force closure.” In fact, in one of the first attempts to use a UAV to interact physically with its environment, Albers et al. (2010) had to add an auxiliary actuator to maintain contact, so Newton’s third law of equal and opposite reaction would not immediately push the UAV away.

12.1.1 Force Closure

Force closure and *form closure* (Mason and Salisbury 1985) are concepts from dexterous manipulation that long predate the field of robotics. Reuleaux (1963) and his contemporaries analyzed mobility under constraint in the late 1800s. Since those early days, force closure and form closure have received significant attention, yet definitions have not always been consistent within the robotics literature. Historically, force closure has been the more mature research area with a well-defined theory and set of definitions revolving around screw theory. Form closure, on the other hand, historically, has been more imprecise. Rimon and Burdick



Fig. 12.1 A ski resort being assembled in Canada (Reprinted with permission from Judy Shellabarger Gosnell)

published a seminal work in robotics that rigorously defined first- and second-order force and form closure and showed their equivalence.

Force closure, as defined by Rimon and Burdick (1996), is the ability of a mechanism to directly resist any arbitrary wrench (combination of force and torque) applied to it. A “force closure grasp” in the dexterous grasping literature (Nguyen 1988; Bicchi 1995; Pham et al. 2006) is a grasp of an object that can resist an arbitrary wrench applied to the object. This class of grasps is important to the dexterous manipulation community but is often ignored by the mobile manipulation community because of the large mass of the mobile base and other issues have greater priority.

12.1.2 Aerial Mobile Manipulation

Aerial mobile manipulation is a newly emerging field even though it has existed for decades. Figure 12.1 shows a ski resort being assembled in Canada. There is a group of workers doing the adjustment as this big helicopter is doing the heavy lifting.

However, the helicopter can’t actually do the assembly. Due to Newton’s third law of equal and opposite reaction, as soon as the manipulated part comes into contact with the environment, it would balance away. This is because the helicopter

cannot exert forces instantaneously in the plane. It can only exert forces up and down, in addition to some pitch and yaw. Through most of the 1990s and early 2000s, mobile robots dominated research and application. And one of those things was quadrotors, helicopters with four blades. So people have started to apply helicopters and quadrotors to mobile manipulation.

Aaron Dollar has studied unstable dynamics of the vehicle and coupled object-aircraft motion while grasping objects during flight (Dollar et al. 2011). They also demonstrate grasping and retrieval of variety of objects while hovering, using Yale Aerial Manipulator. Paul Oh equipped the quadrotor with a gripper and studied about contact forces and external disturbances acted on the gripper and the entire manipulation system (Korpela and Oh 2011). Bernard et al. (2010) and the GRASP lab at the University of Pennsylvania (Mellinger et al. 2010) both have worked on using multiple collaborative UAVs in order to perform transportation tasks. They did research on the interactions between UAVs, physical couplings in the joint transportation of the payload, and stabilizing the payload along three-dimensional trajectories.

The helicopter or quadrotor approach is limited though, only the top of objects can be grasped by the underhung grasper and oddly balanced objects must be lifted by multiple UAVs. The UAV teams rigidly clamp to the manipulated object, but, due to their design, they can only apply limited forces and torques. These piloted helicopters and quadrotors manipulate objects by hanging them from a line and/or employ fewer than six vehicles, severely limiting the range of wrenches that can be applied to the object. In generalized force space, they are effectively degenerate.

The floating nature of aerial platforms and the highly compliant nature of their positioning bring the issue of force closure to the fore. With fixed-base manipulators, determining the degree of force closure of a manipulation system is simplified to determining the degree of force closure of the gripper or end effector. In mobile manipulation, the manipulator base is not fixed to the ground, so determining the set of wrenches that can be resisted is not strictly limited to the capabilities of the end effector. But due to the large differences in mass of the mobile base and end effector, it is generally safe to assume the degree of force closure is limited by the end effector and not by the ability of the mobile base to remain motionless. Therefore, for aerial mobile manipulation, the concept of force closure of the entire manipulation system needs to be considered.

Conventional aerial platforms are not able to resist an arbitrary wrench, so an end effector carried by such a vehicle will not be able to exhibit force closure. Force closure for arbitrary rigid objects in 3-D space requires six controllable degrees of freedom in force/torque space to truly accomplish. Current quadrotors lack both the number of degrees of freedom and independence of the degrees of freedom due to the fact that the force vectors are all parallel.

Furthermore, it is not sufficient to simply attach a 6-DOF manipulator to the bottom of a quadrotor or other degenerate aerial platform, as this does not guarantee force closure. While a 6-DOF manipulator can exert arbitrary wrenches when grounded, if the base upon which it is mounted cannot resist an arbitrary wrench, the combination remains degenerate.

Finally, with the multi-quadrotor approach, force closure has not been obtained; lateral forces must be compensated for by torques, not direct opposition. A single UAV can have more manipulation abilities, chiefly force closure; all forces put on the grasped object can be opposed directly. Also a single UAV has numerous advantages over a system that requires multiple UAVs including simplicity, communications, cost, and available poses.

12.1.3 3-D Translational Flight

Besides force closure, an interesting side effect of a UAV with control over six degrees of freedom is the ability to accomplish 3-D translational flight (hover at any orientations and translate in any directions). It would be much easier for a 6-degrees of freedom UAV to achieve unusual body attitudes as Mellinger et al. did (2010). In their work, they developed elaborate dynamic control methodologies to achieve unusual body attitudes during aggressive flight maneuvers to allow their quadrotor UAVs to pass through narrow windows and other hazards. Using high precision external sensing of the pose of the vehicle and an accurate dynamic model, their quadrotor is capable of passing through the diagonal of a rectangular orifice, presumably to enter a damaged building through a narrow window in response to a disaster. The window presents an orifice of which the horizontal width is not sufficient to allow passage of the UAV, but the diagonal distance is. Since a quadrotor cannot hover at an arbitrary orientation (the result of a mapping from actuator space to Cartesian generalized force space with rank less than six, indicative of incomplete force closure), an aggressive dynamic maneuver is required to achieve entry.

Therefore, as a way to gain full controllability over the 6-DOF robot pose and track an arbitrary trajectory over time (e.g., it can hover on one spot with an angled pose), some UAV platforms with tilting propellers have been designed. M. Ryll proposed a quadrotor design with tilting propellers (Ryll et al. 2012). In their work, to solve the problem of limited mobility of standard quadrotor, rather than four fixed rotors, four variable-pitch rotors are used to provide an additional set of control inputs. Because of standard quadrotor's inherent underactuation for 6 degrees of freedom parameterizing the quadrotor position/orientation in space, Ryll claims that they can gain full controllability over the quadrotor position/orientation by means of these four additional actuated degrees of freedom.

Developed about the same time as this Dexterous Hexrotor design, a similar nonplanar design was introduced by the University of Manchester (Langkamp et al. 2011). Using six fixed-pitch/variable-speed or variable-pitch/fix-speed rotors, the "Tumbleweed" is designed to achieve full flight envelope (their saying of 3-D translational flight).

A prototype is proposed with fixed-pitch/variable-speed rotors, and all rotors are pitched at 45° . But as they claimed, it cannot achieve full flight envelope because of limited forces it can generate in frame plane. So they abandoned this design and shifted to a variable-pitch/fix-speed rotor design. And by the use of high-thrust/weight electric variable-pitch units and a low airframe mass fraction,

the Tumbleweed can achieve full flight envelope if the robot can be lifted by only one pair of rotors.

In these two works, the actuation concept of tilting propellers during flight actually makes it possible to access all 6 degrees of freedom of the robot. But for *aerial manipulation* tasks, the aerial platform needs to exert forces as fast as possible to resist any arbitrary wrench. Tilting propellers during flight using servos may not be fast enough for our purposes. So each rotor on Dexterous Hexrotor is pitched at an optimized angle based on specific task.

This chapter proposes a new approach in aerial vehicles as Dexterous Hexrotor UAV that can instantaneously resist arbitrary forces – in other words, it provides force closure. To perform precise and effective mobile manipulation, this is a property that any locomotor must have, be it ground based, water based, or air based. To achieve this, the thrusters of Dexterous Hexrotor are canted so that the combined thrust vectors span the space of Cartesian forces and torques. This adds little cost or complexity to a conventional hexrotor.

With decomposed forces and torques, a thrust mapping from actuator space to force/torque for Dexterous Hexrotor is derived.

Since each rotor is rotated a cant angle around its radius, a metric for the optimization of Dexterous UAV performance for manipulation-based tasks based on the condition number of the conversion matrix in the thrust mapping is developed.

With an attitude controller created to stabilize the Dexterous Hexrotor for human-controlled flight, a flight-capable prototype with translational force control has been built and tested.

12.2 Dexterous Hexrotor Theory

The basic idea behind hexrotor is 6 actuators will be provided. So the input is 6 degrees of freedom. If it needs to control 6 degrees of freedom, it has to have a minimum of 6 actuators. Or else the matrix is going to be deficient. If only 4 actuators are provided as a quadrotor, it cannot possibly have independent control over 6 degrees of freedom.

There are many hexrotors in the commercial world today. They all have six parallel thrust propellers spaced evenly around the circumference of a circle. All its thrusters are vertical, just like typical quadrotors as in Fig. 12.2. Therefore, they still result in rank-deficient matrices; in other words, there are no components from parallel thrusts that act in the plane perpendicular to the rotor axis. In this kind of configuration, these hexrotors work like quadrotors.

Because all these thrusters are parallel and vertical, they can only provide linear force along Z axis, and torques around X, Y axes. Torque around Z is achieved indirectly through Coriolis forces resulting from differential angular velocities of the counterrotating propellers.

As shown in Fig. 12.3, if all the rotors spin at a particular same speed, the robot hovers. If the speed of these rotors varies, the robot can roll, pitch, and yaw.

Fig. 12.2 Typical quadrotor

This configuration only has controllability over 4 degrees of freedom parameterizing the robot position and orientation in space. But it still can move arbitrarily in 6 degrees of freedom space, because translational acceleration in the plane perpendicular to the rotor axis can be achieved by maintaining a nonzero pitch or roll angle. So with these four degrees of freedom, quadrotors can do hover, roll, pitch, and yaw. If the robot wants to move in the plane perpendicular to the rotor axis, it has to maintain a nonzero pitch or roll angle to get the translational acceleration.

To achieve 6 degrees of freedom, 6 actuators are provided as a typical hexrotor design. Then these parallel thrusters are made nonparallel to explore full entirely 6-D space of forces and torques.

As Fig. 12.4 shows, Dexterous Hexrotor has six independently controlled rotors arranged in pairs on three planes. Each rotor is rotated a cant angle around its radius. Therefore in-plane components result while still maintaining a symmetric basis of vectors, and forces and torques around each axis can be produced. Since Dexterous Hexrotor can span the force/torque space, by detaching and combining forces and torques produced by each rotor, forces and torques acting on the UAV around each axis can be accomplished, and controllability over full 6 degrees of freedom can be achieved. By varying the speed and choosing the direction of the rotation of the rotors, Dexterous Hexrotor can get not only forces along Z axis and torques around X, Y, Z axes but also forces along X, Y axes.

As shown in Fig. 12.5, with the force along Z axis and torques around X, Y, Z axes, Dexterous Hexrotor can hover, roll, pitch, and yaw just like typical quadrotor does. But instead of pitching or rolling an angle like typical quadrotors, Dexterous Hexrotor can get translational acceleration by simply varying speed of these rotors. It can truly control its 6 degrees of freedom mobility.

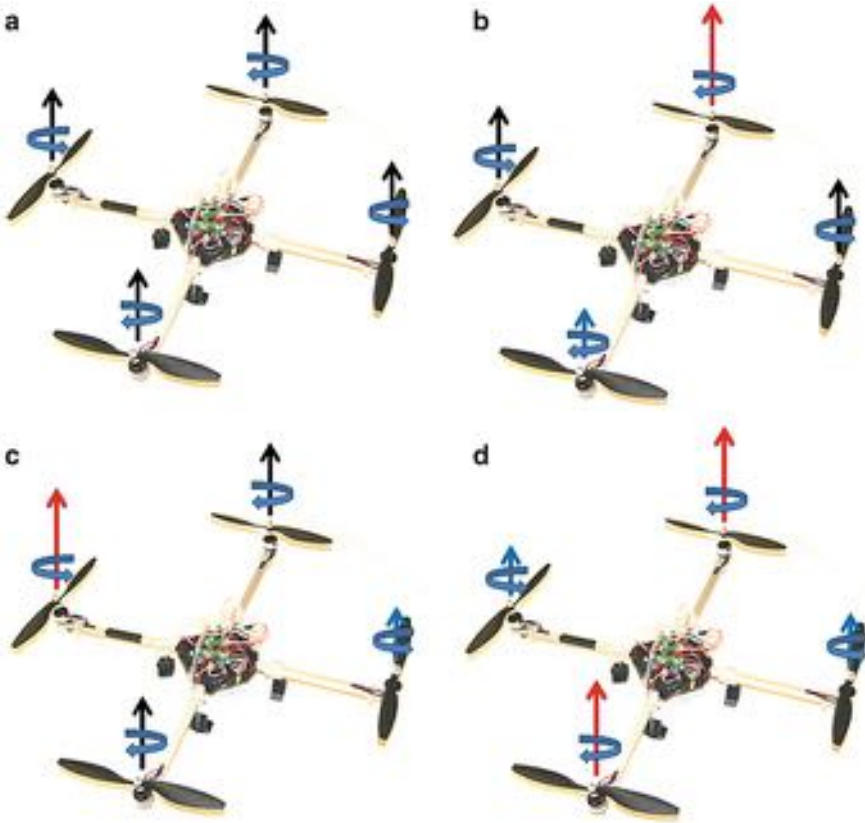


Fig. 12.3 How quadrotor moves: (a) hover, (b) roll, (c) pitch, and (d) yaw

12.2.1 Dexterous Hexrotor Structure

There are many ways to use six motors to create six independent degrees of freedom, but to create a design that was easy to fabricate, a disk design was used. To get linear independence for all six degrees of freedom, the thrust vectors could not be on the same axis nor pointed along one axis. This fact leads to an arrangement which these six rotors are put in pairs of three planes and each thruster is rotated a cant angle ϕ around its radius. These rotors are laid out along the edge of the disk canted tangentially to the edge of the disk alternating clockwise, counterclockwise, clockwise, and so on.

Position of six motors and their rotation are defined in Fig. 12.6. X configuration of multicopter configuration is chosen, so X axis aligns with Motor3 and Motor6.

To keep the direction of each rotor's torque same with its force's in-plane components, Motor1, Motor3, and Motor5 are rotating clockwise, while Motor2, Motor4, and Motor6 are rotating counterclockwise. And to minimize the effects



Fig. 12.4 Dexterous Hexrotor: note the rotors are nonparallel

of the torques from the motors themselves, pusher and puller propellers are used; Motor1, Motor3, and Motor5 use one type, while Motor2, Motor4, and Motor6 use the other.

12.2.2 Force Decomposition

Each rotor produces a force and a torque. The force is a thrust pointed out along the rotor axis. The torque is generated by the drag of the propellers and acts on the body of the robot. The direction of the torque will be in the opposite direction of the motion of the propeller. This torque serves as yawing torque in typical quadrotors. But for Dexterous Hexrotor, it will contribute to torque around all three axes.

The lift and drag produced by the propellers is proportional to the square of angular velocity. And the square of angular velocity is proportional to the pulse width modulation command sent to the motors. Therefore, force and torque produced by each motor can be expressed as Eq. (12.1):

$$\begin{aligned} F_{\text{motor}} &= K_1 * \text{PWM}_{\text{motor}} \\ \tau_{\text{motor}} &= K_2 * \text{PWM}_{\text{motor}} \end{aligned} \quad (12.1)$$

where F_{motor} and τ_{motor} are force and torque produced by the motors. K_1 and K_2 are motor-dependent parameters and can be determined experimentally. $\text{PWM}_{\text{motor}}$ is the pulse width modulation command sent to the motor.

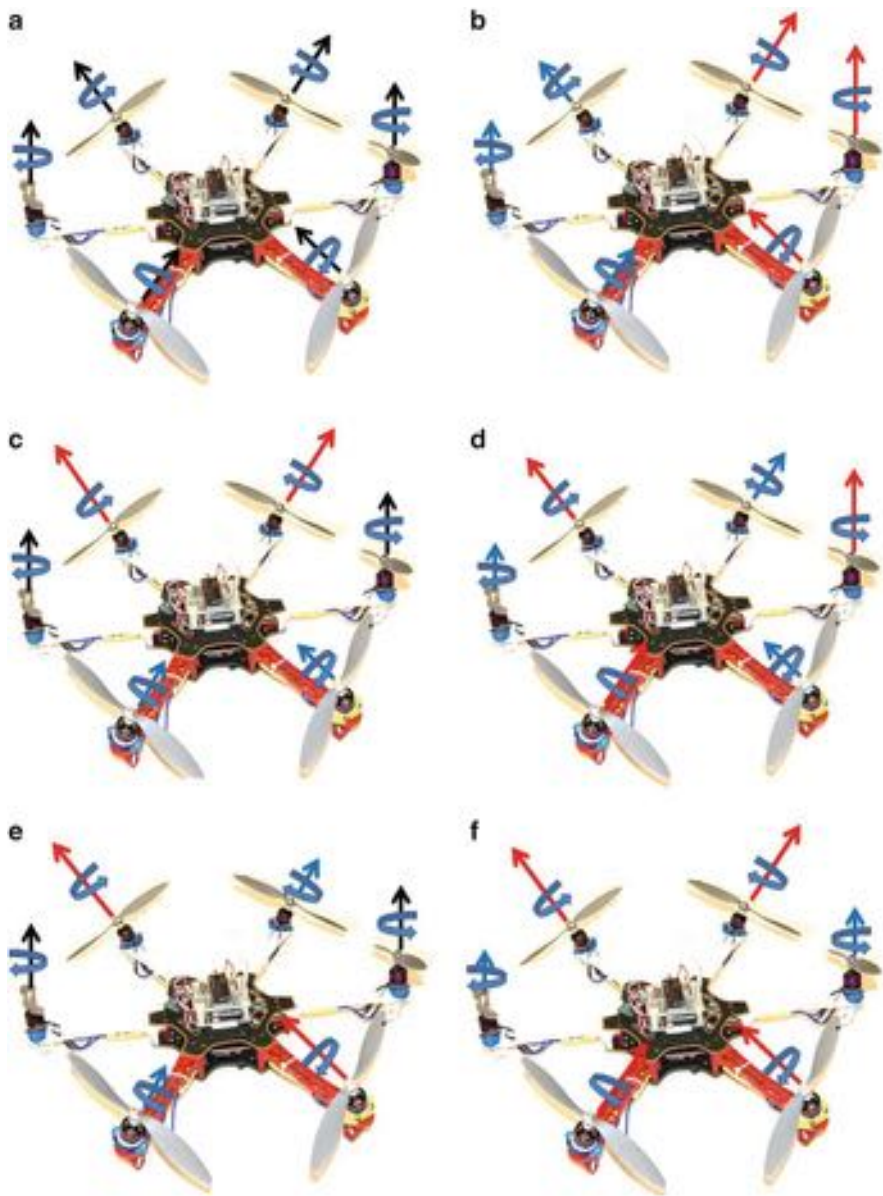
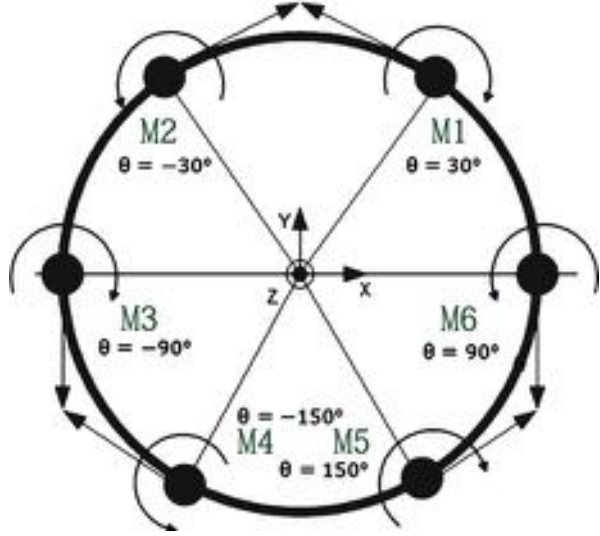


Fig. 12.5 How Dexterous Hexrotor moves: (a) hover, (b) roll, (c) pitch, (d) yaw, (e) translational acceleration along X axis, and (f) translational acceleration along Y axis

Fig. 12.6 Motor definition of Dexterous Hexrotor, where n is the motor Number and θ represents angular displacements of the motors



To compute the net force/torque acting on the UAV from all thrusters, first each motor's thrust and torque is decomposed into X, Y, and Z components on the body frame. The components of Cartesian generalized forces from Motor1 can be put into a matrix as in Eqs. (12.2) and (12.3):

$$\begin{bmatrix} F_{1fx} \\ F_{1fy} \\ F_{1fz} \\ F_{1\tau x} \\ F_{1\tau y} \\ F_{1\tau z} \end{bmatrix} = \begin{bmatrix} -K_1 * PWM_1 * \cos(\theta_1) * \sin(\phi) \\ K_1 * PWM_1 * \sin(\theta_1) * \sin(\phi) \\ K_1 * PWM_1 * \cos(\phi) \\ d * K_1 * PWM_1 * \cos(\theta_1) * \cos(\phi) \\ -d * K_1 * PWM_1 * \sin(\theta_1) * \cos(\phi) \\ d * K_1 * PWM_1 * \sin(\phi) \end{bmatrix} \quad (12.2)$$

$$\begin{bmatrix} T_{1x1} \\ T_{1y1} \\ T_{1z1} \end{bmatrix} = \begin{bmatrix} -K_2 * PWM_1 * \cos(\theta_1) \sin(\phi) \\ K_2 * PWM_1 * \sin(\theta_1) \sin(\phi) \\ K_2 * PWM_1 * \cos(\phi) \end{bmatrix} \quad (12.3)$$

where K_1 and K_2 are the constants between force and torque produced by Motor1 and PWM1 (the pulse width modulation command sent to Motor1), ϕ is the cant angle from vertical, θ represents the rotor's position, and d is the distance from rotor 1 to the central axis (the radius of Dexterous Hexrotor). $[F_{1fx} \ F_{1fy} \ F_{1fz} \ F_{1\tau x} \ F_{1\tau y} \ F_{1\tau z}]^T$ are forces and torques decomposed from F_1 (the force produced by Motor1) and $[\tau_{1\tau x} \ \tau_{1\tau y} \ \tau_{1\tau z}]^T$ are torques decomposed from τ_1 (the torque produced by Motor1).

Then the total force/torque $[F_{1x} \ F_{1y} \ F_{1z} \ \tau_{1x} \ \tau_{1y} \ \tau_{1z}]^T$ acting on Dexterous Hexrotor from Motor1 can be derived as

$$\begin{aligned}
\begin{bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ \tau_{1x} \\ \tau_{1y} \\ \tau_{1z} \end{bmatrix} &= \begin{bmatrix} F_{1fx} \\ F_{1fy} \\ F_{1fz} \\ F_{1\tau x} + \tau_{1\tau x} \\ F_{1\tau y} + \tau_{1\tau y} \\ F_{1\tau z} + \tau_{1\tau z} \end{bmatrix} \\
&= \begin{bmatrix} -K_1 * PWM_1 * \cos(\theta_1) * \sin(\phi) \\ K_1 * PWM_1 + \sin(\theta_1) * \sin(\phi) \\ K_1 * PWM_1 * \cos(\phi) \\ PWM_1 * \cos(\theta_1) * (d * K_1 * \cos(\phi) - K_2 * \sin(\phi)) \\ PWM_1 * \cos(\theta_1) * (-d * K_1 * \cos(\phi) + K_2 * \sin(\phi)) \\ PWM_1 * (-d * K_1 * \sin(\phi) + K_2 * \sin(\phi)) \end{bmatrix} \\
&= PWM_1 * \begin{bmatrix} -K_1 C \theta_1 S \phi \\ K_1 C \theta_1 S \phi \\ K_1 C \phi \\ C \theta_1 (d K_1 C \phi - K_2 S \phi) \\ C \theta_1 (-d K_1 C \phi + K_2 S \phi) \\ d K_1 S \phi + K_2 C \phi \end{bmatrix} \tag{12.4}
\end{aligned}$$

Once one rotor is decomposed, forces and torques produced by other motors can be decomposed in the same way.

12.2.3 Mapping from Actuator Space to Force/Torque Space

After decomposition of force/torque produced by each motor, now the net force/torque of Dexterous Hexrotor can be computed as Eq. (12.5):

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ \tau_{1x} \\ \tau_{1y} \\ \tau_{1z} \end{bmatrix} + \begin{bmatrix} F_{1x} \\ F_{1y} \\ F_{2z} \\ \tau_{2x} \\ \tau_{2y} \\ \tau_{2z} \end{bmatrix} + \begin{bmatrix} F_{3x} \\ F_{3y} \\ F_{3z} \\ \tau_{3x} \\ \tau_{3y} \\ \tau_{3z} \end{bmatrix} + \begin{bmatrix} F_{4x} \\ F_{4y} \\ F_{4z} \\ \tau_{4x} \\ \tau_{4y} \\ \tau_{4z} \end{bmatrix} + \begin{bmatrix} F_{5x} \\ F_{5y} \\ F_{5z} \\ \tau_{5x} \\ \tau_{5y} \\ \tau_{5z} \end{bmatrix} + \begin{bmatrix} F_{6x} \\ F_{6y} \\ F_{6z} \\ \tau_{6x} \\ \tau_{6y} \\ \tau_{6z} \end{bmatrix} \tag{12.5}$$

where $[F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T$ is the net force/torque acting on the body of Dexterous Hexrotor.

From force decompositions, the right part of equation equals this 6×6 conversion matrix multiplied by $[PWM_1 \ PWM_2 \ PWM_3 \ PWM_4 \ PWM_5 \ PWM_6]^T$.

$$\begin{bmatrix} -K_1 C \theta_1 S \phi & K_1 C \theta_2 S \phi & K_1 C \theta_3 S \phi & K_1 C \theta_4 S \phi & K_1 C \theta_5 S \phi & K_1 C \theta_6 S \phi \\ K_1 S \theta_1 S \phi & -K_1 S \theta_2 S \phi & K_1 S \theta_3 S \phi & -K_1 S \theta_4 S \phi & K_1 S \theta_5 S \phi & -K_1 S \theta_6 S \phi \\ K_1 C \phi & K_1 C \phi & K_1 C \phi & K_1 C \phi & K_1 C \phi & K_1 C \phi \\ C \theta_1 (dK_1 C \phi - K_2 S \phi) & C \theta_2 (dK_1 C \phi - K_2 S \phi) & C \theta_3 (dK_1 C \phi + K_2 S \phi) & C \theta_4 (dK_1 C \phi - K_2 S \phi) & C \theta_5 (dK_1 C \phi - K_2 S \phi) & C \theta_6 (dK_1 C \phi + K_2 S \phi) \\ S \theta_1 (-dK_1 C \phi + K_2 S \phi) & S \theta_2 (dK_1 C \phi - K_2 S \phi) & S \theta_3 (-dK_1 C \phi + K_2 S \phi) & S \theta_4 (-dK_1 C \phi + K_2 S \phi) & S \theta_5 (-dK_1 C \phi + K_2 S \phi) & S \theta_6 (-dK_1 C \phi + K_2 S \phi) \\ dK_1 S \phi + K_2 C \phi & -dK_1 S \phi - K_2 C \phi & dK_1 S \phi + K_2 C \phi & -dK_1 S \phi - K_2 C \phi & dK_1 S \phi + K_2 C \phi & -dK_1 S \phi - K_2 C \phi \end{bmatrix}$$

Express this matrix as M_φ . This equation can be concluded as

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = M_\varphi \cdot \begin{bmatrix} PWM_1 \\ PWM_2 \\ PWM_3 \\ PWM_4 \\ PWM_5 \\ PWM_6 \end{bmatrix} \quad (12.6)$$

In Dexterous Hexrotor, $[F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T$ is the desired force/torque vector. Values of this vector will decide the robot's orientation and position. $[PWM_1 \ PWM_2 \ PWM_3 \ PWM_4 \ PWM_5 \ PWM_6]^T$ are six independent controlled inputs for the robot. Based on this equation, a relationship between inputs (PWM commands) and outputs (force/torque) of Dexterous Hexrotor can be established, and a mapping from UAV's actuator space to Cartesian force/torque space can be built. Therefore, to control Dexterous Hexrotor and get desired force/torque vector, the inversed conversion matrix is multiplied by the desired force/torque vector, then PWM commands are calculated and sent to the motors. The force/torque control equation can be derived as

$$\begin{bmatrix} PWM_1 \\ PWM_2 \\ PWM_3 \\ PWM_4 \\ PWM_5 \\ PWM_6 \end{bmatrix} = [M_\varphi]^{-1} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (12.7)$$

The conversion matrix is the mapping from actuator space to force/torque space. With K_1 and K_2 determined for motors, if the cant angle is zero, the conversion matrix is M_{0° :

$$M_{0^\circ} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 5.7 & 5.7 & 5.7 & 5.7 & 5.7 & 5.7 \\ 1.33 & 1.3 & 0 & -1.33 & -1.33 & 0 \\ -0.769 & 0.769 & 1.54 & 0.769 & -0.769 & -1.54 \\ 0.13 & -0.13 & 0.13 & -0.13 & 0.13 & -0.13 \end{bmatrix}$$

which would conform to a typical hexrotor design. M_{0° only has a rank of 4. It has no ability to instantaneously control forces in X and Y axes through this matrix.

If the thrusters are canted at an angle, for example, at 20° , the thrust mapping becomes M_{20° :

$$M_{20^\circ} = \begin{bmatrix} -1.69 & 1.69 & 0 & -1.69 & 1.69 & 0 \\ 0.975 & 0.975 & -1.95 & 0.975 & 0.975 & -1.95 \\ 5.36 & 5.36 & 5.36 & 5.36 & 5.36 & 5.36 \\ 1.21 & 1.21 & 0 & -1.21 & -1.21 & 0 \\ -0.701 & 0.701 & 1.4 & 0.701 & -0.701 & -1.4 \\ 0.649 & -0.649 & 0.649 & -0.649 & 0.649 & -0.649 \end{bmatrix}$$

which would conform to a nonparallel design. This matrix provides a mapping from 6-D actuator space to 6-D force/torque space and has a rank of 6, indicating 6 independent controlled degrees of freedom in Cartesian force/torque space.

12.2.4 Dexterous Hexrotor Performance Optimization

The cant angle decides how much force it can get in $[F_z \ \tau_x \ \tau_y]^T$ or $[F_x \ F_y \ \tau_z]^T$. If $\varphi = 0^\circ$, the matrix becomes

$$M_{0^\circ} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 5.7 & 5.7 & 5.7 & 5.7 & 5.7 & 5.7 \\ 1.33 & 1.3 & 0 & -1.33 & -1.33 & 0 \\ -0.769 & 0.769 & 1.54 & 0.769 & -0.769 & -1.54 \\ 0.13 & -0.13 & 0.13 & -0.13 & 0.13 & -0.13 \end{bmatrix}$$

Clearly the matrix becomes degenerate as there are no forces in X or Y from rotor force/torque decomposition; this would conform to a fairly typical quadrotor design. There's no mapping from actuator space to forces in X and Y. It's rank deficient. The opposite happens when φ is 90° :

$$M_{90^\circ} = \begin{bmatrix} -4.94 & 4.94 & 0 & -4.94 & 4.94 & 0 \\ 2.85 & 2.85 & -5.7 & 2.85 & 2.85 & -5.7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.11 & -0.11 & 0 & 0.11 & 0.11 & 0 \\ 0.065 & -0.065 & -0.13 & -0.065 & 0.065 & 0.13 \\ 1.54 & -1.54 & 1.54 & -1.54 & 1.54 & -1.54 \end{bmatrix}$$

Force can be applied in X and Y but no lift and the matrix becomes deficient again. With these examples in mind, it is obvious that the closer to $\varphi = 0^\circ$, the more

Table 12.1 Parameters of Dexterous Hexrotor engineering prototype

| | |
|---------------------|--------|
| K1 | 5.7 |
| K2 | 1.3 |
| Diameter | 0.27 m |
| Mass of UAV | 2 kg |
| Mass of manipulator | 0.5 kg |
| Desired payload | 0.3 kg |

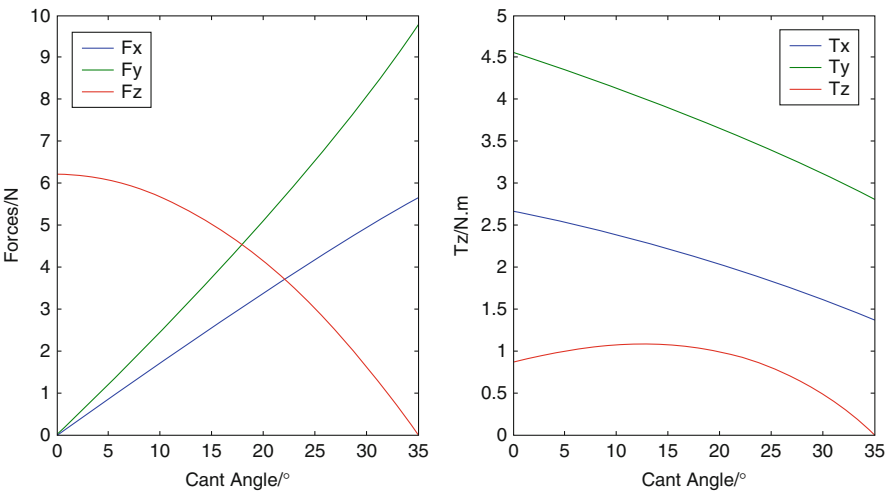


Fig. 12.7 Maximum forces Dexterous Hexrotor can achieve at different cant angles

Dexterous Hexrotor can lift, but the closer to $\varphi = 90^\circ$, the more lifting force can be obtained while tilted, or the more Dexterous Hexrotor can tilt.

A cant angle has to be chosen somewhere between 0° and 90° when a prototype of Dexterous Hexrotor is being built. So a metric for optimization of Dexterous Hexrotor UAV’s performance has been developed.

The performance of Dexterous Hexrotor at different cant angles is affected by several variables, including the thrust of the motors, the diameter of the UAV, and the load it needs to carry. For manipulation tasks, the engineering requirement for Dexterous Hexrotor is the desired payload.

So for Dexterous Hexrotor engineering prototype, the thrust of the motors and the diameter of the UAV are given based on the motors and frame. Given the system’s own mass and desired payload, the load it needs to carry is also set. The only thing undecided is the cant angle.

With parameters in Table 12.1, maximum forces and torques Dexterous Hexrotor that can achieve near hover condition at different cant angles can be plotted in Fig. 12.7.

Clearly at $\varphi = 0^\circ$, there are no forces in X or Y and not much torques around Z. The opposite happens when $\varphi = 35^\circ$, where all forces in Z are used to provide

lift, but more force can be applied in X and Y. This means Dexterous Hexrotor engineering prototype can operate from 0° to 35° . So a cant angle between 0° and 35° should be chosen.

To optimize the cant angle for the performance of Dexterous Hexrotor, Yoshikawa's concept of "manipulability" was adapted here. As defined by Yoshikawa, "manipulability measure" is a quantitative measure of manipulating ability of robot arms in positioning and orienting the end effectors, by looking at the isotropism of manipulator's motion in linear dimensions X, Y, Z and angles roll, pitch, and yaw (Yoshikawa 1985). So for Dexterous Hexrotor the combination of forces and torques is considered as a similar measure of mobility. The isotropism of the forces and torques is going to be checked, not just how strong is it. To visualize how isotropic are the forces and torques, these force/torque ellipsoids are plotted in Fig. 12.8.

The radius of these ellipsoids represents the magnitude of maximum force/torque Dexterous Hexrotor can achieve around X, Y, Z axes at near hover condition. When the cant angle is 0° , it gets no ability to control F_x , F_y , but a lot of ability to control F_z . It also has ability to control τ_x τ_y τ_z , but not too much control over yaw because Coriolis effect is weak. Then the motors are canted a little bit; it gets a little bit control over F_x , F_y , but still very strong control in F_z . And likewise it gets a little more control over yaw. At 20° , the force ellipsoid becomes almost equal, and the torque ellipsoid also gets more round. At 30° , both force ellipsoid and torque ellipsoid start to squash down again. Therefore, from 0° to 30° and beyond, the force/torque ellipsoid can get very isotropic at some point, which is good in this mobility measure.

The condition number of the conversion matrix is also a way to check the isotropism of the forces and torques. Condition number is a mathematic term from linear algebra, which is the ratio of maximum eigenvalue to the minimum eigenvalue of the matrix. It can be seen as the rate at which one side of the equation will change with respect to the other side. If the condition number is large, even a small error in one side may cause a large error in the other side.

As shown in Fig. 12.9, at 0° , when it has no control over F_x and F_y , the condition number would be infinite, because two of the eigenvalues are 0. Then it can get smaller and smaller when increasing the cant angle. It is hard getting to 1 since force and torque are measured under different scales. Eventually, it gets higher again and it will be infinite at 90° , because one of the eigenvalues is 0.

These two metrics can be combined on a same plot as shown in Fig. 12.10. The condition number of the conversion matrix with force/torque ellipsoids at different cant angles is plotted, giving a measure of the isotropism of Dexterous Hexrotor UAV.

Therefore, dependent on the motors, particular load of the manipulator, and diameter of the UAV frame, the cant angle is optimized at 20° for this Dexterous Hexrotor engineering prototype. If the load or any other parameters are changed, a different cant angle would probably result after the optimization. And for tasks other than manipulation, its performance can be optimized with other parameters as well as the cant angle.

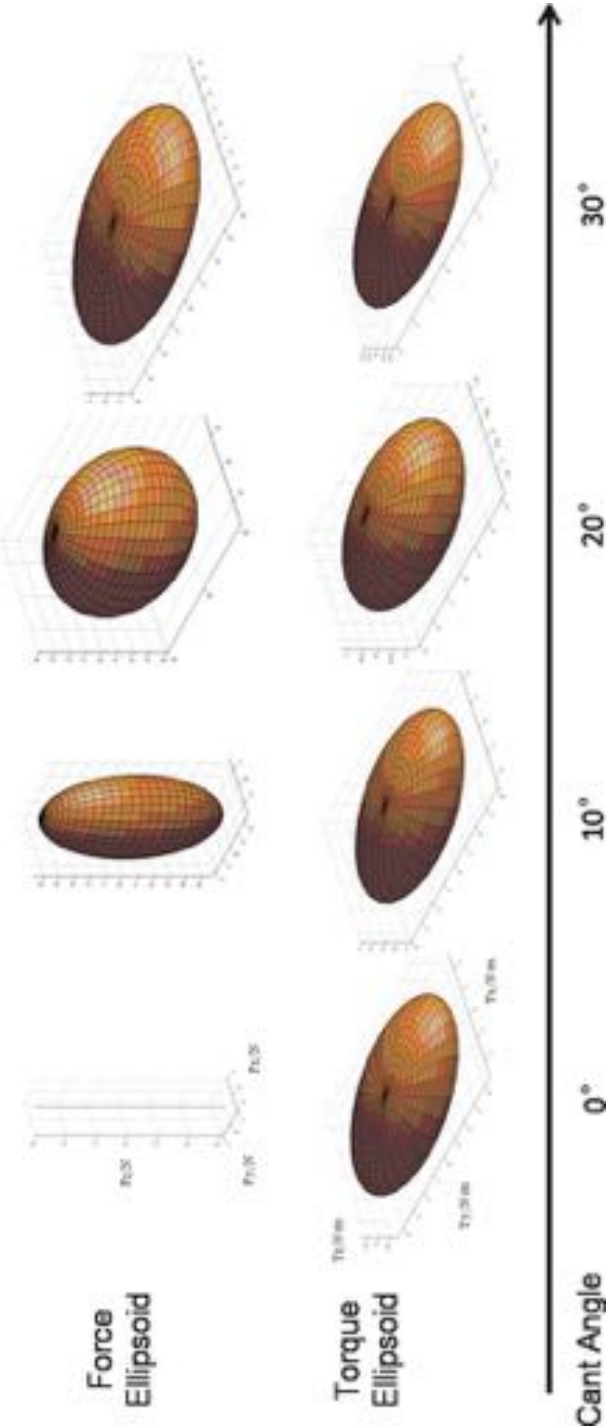


Fig. 12.8 Force/torque ellipsoids at different cant angles

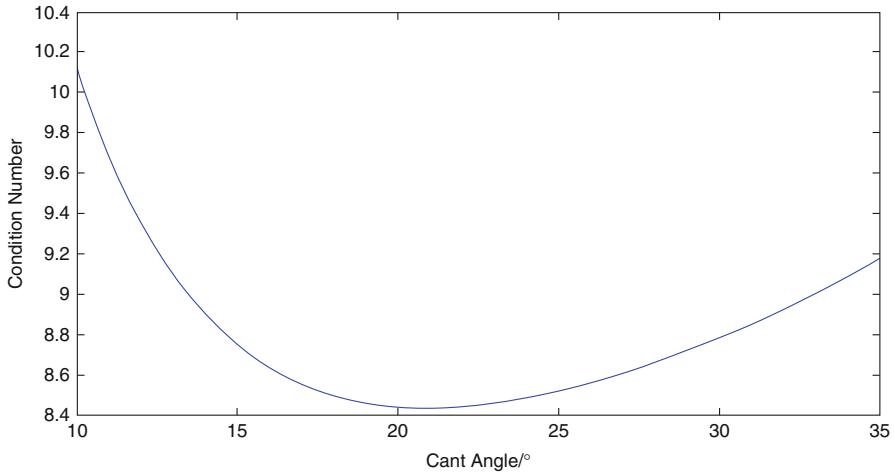


Fig. 12.9 Condition number of the conversion matrix

12.3 Dexterous Hexrotor Control

12.3.1 Dexterous Hexrotor Control System

For human-controlled flight, the pilot controls Dexterous Hexrotor's orientation and position by manipulating the forces and torques acting on the platform. For Dexterous Hexrotor, there are six independent controlled forces and torques acting on itself. Therefore, control systems to help the pilot control Dexterous Hexrotor's orientation and position are developed.

Forces and torques acted on the Dexterous Hexrotor are generated by the thrust of each rotor. And the thrust is proportional to the PWM value put into the motor. Based on the force/torque control equation, to generate desired force/torque, PWM values which are calculated from desired force/torque vector $[F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T$ through thrust mapping need to be sent to each motor.

Multicopters are inherently unstable in the air. Therefore, an attitude controller is needed to stabilize the UAV during flight. A common control system used in quadrotors is as shown in Fig. 12.11.

This system is constructed by radio receiver, attitude controller, IMU (inertial measurement unit), thrust mapping, and motors. With the help of attitude controller stabilizing the UAV, the pilot only needs to control its attitude to change the orientation and position (e.g., pitching for moving forward).

The difference between controlling Dexterous Hexrotor and quadrotor is the thrust mapping. Thrust mapping of quadrotor only maps $[F_z \ \tau_x \ \tau_y \ \tau_z]^T$ to motors. Dexterous Hexrotor's thrust mapping needs to map all force/torque around three axes to motors. So the control system should be as it is shown in Fig. 12.12.

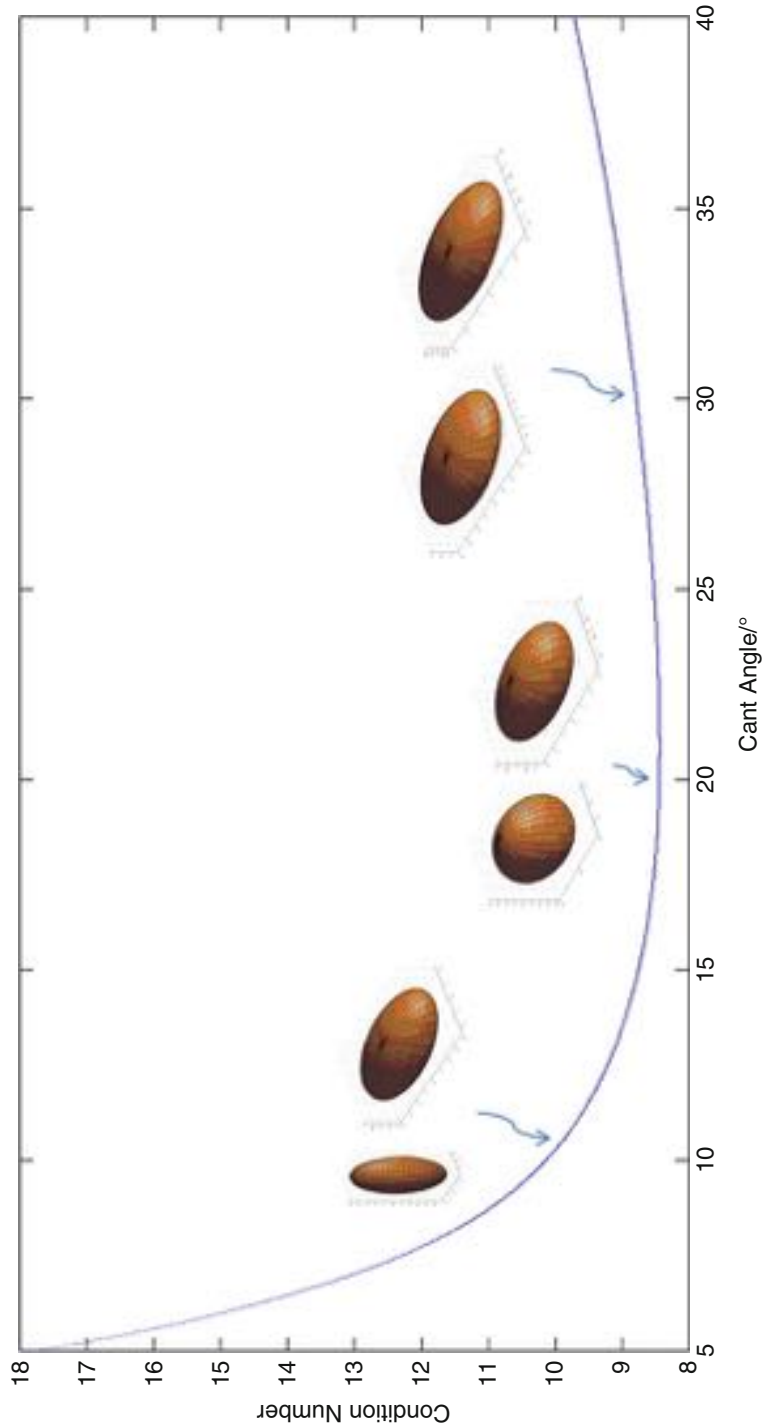


Fig. 12.10 Condition number with force/torque ellipsoids

Fig. 12.11 Quadrotor control system

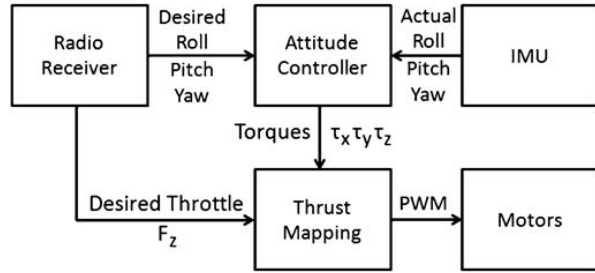
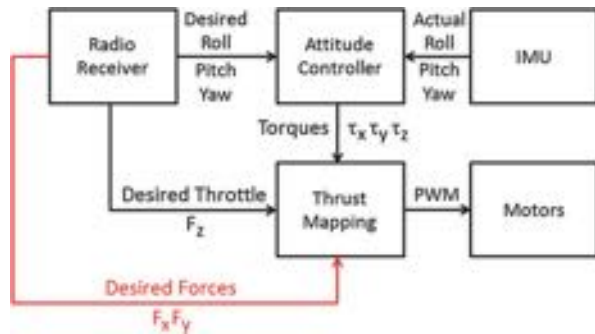


Fig. 12.12 Dexterous Hexrotor control system



Thrust mapping in this system maps $[F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T$ to $[PWM_1 \ PWM_2 \ PWM_3 \ PWM_4 \ PWM_5 \ PWM_6]^T$. It is a mapping from 6-D actuator space to 6-D force/torque space.

Control system of Dexterous Hexrotor has two more channels controlling F_x and F_y than quadrotor. These forces are in the plane perpendicular to the rotor's axis. They can provide horizontal acceleration. These forces can be used to exert forces immediately in the plane, and they also give the pilot another option of moving Dexterous Hexrotor. When the pilot wants to move the quadrotor, it needs to pitch or roll an angle in order to get the translational acceleration. Dexterous Hexrotor can simply generate the translational acceleration by F_x or F_y .

The difference between control system of Dexterous Hexrotor and typical quadrotor is as shown in Table 12.2.

12.3.2 Hexrotor Attitude Controller

The attitude controller is designed to stabilize Dexterous Hexrotor for human-controlled flight. It helps Dexterous Hexrotor maintaining desired attitude during flight by controlling Dexterous Hexrotor roll, pitch, and yaw. This is also a commonly used control structure in quadrotor community.

For stability and also fast response, three double-loop PID controllers for roll, pitch, and yaw have been developed, and they share same structure as it is shown in Fig. 12.13.

Table 12.2 Difference between control system of Dexterous Hexrotor and typical quadrotor

| | Quadrotor | Dexterous Hexrotor |
|--------------------------------------|---|--|
| Pilot controlled variables | Roll, pitch, yaw, Fz | Roll, pitch, yaw, Fx, Fy, Fz |
| Number of pilot controlled variables | 4 | 6 |
| Inputs to thrust mapping | $F_z \ \tau_x \ \tau_y \ \tau_z$ | $F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z$ |
| Number of inputs to thrust mapping | 4 | 6 |
| Outputs of thrust mapping | PWM ₁ PWM ₂ PWM ₃ PWM ₄ | PWM ₁ PWM ₂ PWM ₃ PWM ₄ PWM ₅ PWM ₆ |

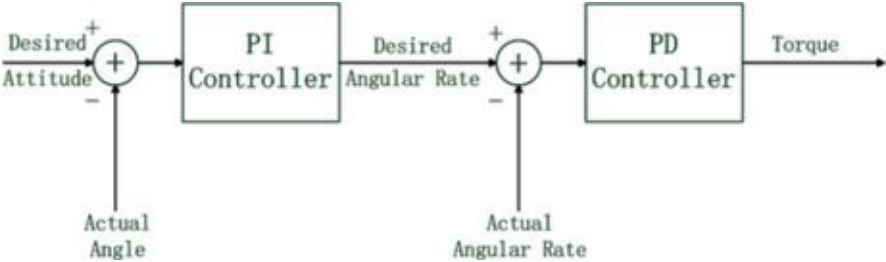


Fig. 12.13 Double-loop PID controller

The outer loop is using a PI controller for controlling the angle. It takes desired angle as input, actual IMU angle as feedback, and outputs desired angular rate to the inner loop.

The inner loop is using a PD controller for controlling angular rate. It takes angular rate as input, actual IMU angular rate as feedback, and outputs torques to adjust the attitude of Dexterous Hexrotor. The PI and PD combination is chosen for a good reason. A single traditional PID controller would work, but slow. An outer loop controlling angle and an inner loop controlling rate would be much better for its fast response. P item is necessary in both loops for fast response. A D term in the inner loop is for improving response time. An I term in the outer loop is for helping Dexterous Hexrotor dealing with persistent external forces, like wind or incorrect center of gravity.

12.4 Dexterous Hexrotor Prototype

A prototype that uses off-the-shelf components, preferably those commonly used by modern research quadrotors, has been built to improve cost, compatibility, and simplicity. The prototype of Dexterous Hexrotor design can be seen in Fig. 12.14.

The motors are mounted on in-house designed and fabricated ABS plastic adapters that cant 20° tangentially to the edge. These plastic adapters are then mounted on the end of each arm as shown in Fig. 12.15.

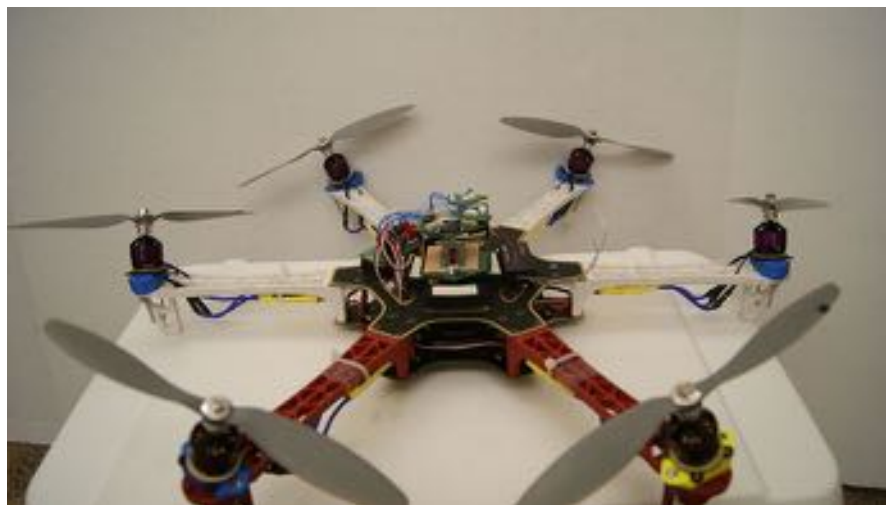


Fig. 12.14 Dexterous Hexrotor prototype built on a commercial frame



Fig. 12.15 Motor mounted on a 20° ABS plastic adapter

Standard 11.1 V lithium polymer battery packs are used to power the 30 A ESCs (electronic speed controllers). Each ESC drives a 1,130 kV brushless DC motor with a three-phase signal converted from the PWM command. The motors are installed with 9" \times 4.7" pusher or puller propellers.

A 2.4 GHz radio transmitter and receiver set are used for human-controlled flight. Four main joysticks are used for roll, pitch, yaw, and throttle. Two side knobs are used to set F_x and F_y values.

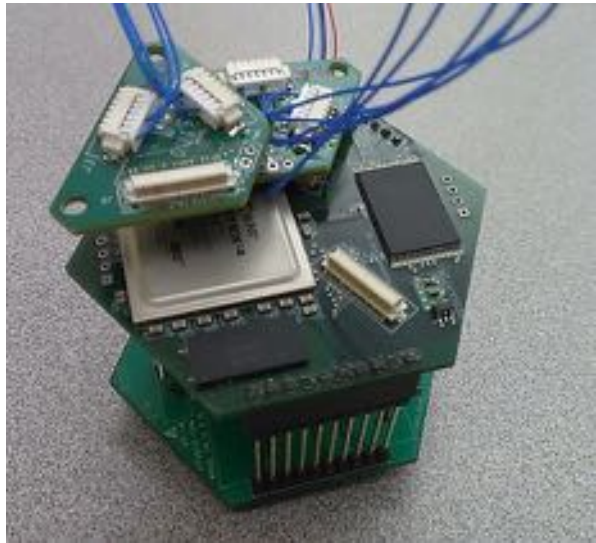
The controller is the RecoNode platform developed in collaborative mechatronics lab. A universal PCB board is used for routing signals between onboard electronics, the RecoNode Platform, IMU, radio receiver, and ESCs as shown in Fig. 12.16.

A Sparkfun 9-DOF sensor stick is used as IMU. It includes an ADXL345 accelerometer, an HMC5883L magnetometer, and an ITG-3200 gyro. It communicates with CPU by I2C interface.

Fig. 12.16 Onboard electronics of Dexterous Hexrotor



Fig. 12.17 RecoNode stack with Virtex4 FX20 CPU and two single-width wedges



The RecoNode is a multiprocessor architecture based on the Virtex 4 FPGA with multiple, hardcore PowerPCs. Capable of up to 1,600 MIPS from four PowerPCs plus hundreds of additional MIPS of special-purpose coprocessing from the FPGA fabric itself, this computational node is very high in performance compared to conventional wireless sensor nodes – roughly 100 times greater computational throughput. The RecoNodes are dual-processor models running at 400 MIPS with a power budget of 0.9 mW/MIPS.

The RecoNode used in Dexterous Hexrotor has two base boards and two same wedges. In Fig. 12.17, from bottom to top: DU105 power board, providing 1.2, 2.5, 3.3, and 5 V for the system; DU100 CPU board, carrying a Virtex4 FX20 CPU running at 100 MHz; DU121 servo wedge, transferring the PWM commands from CPU to ESCs and input signals from radio receiver and IMU to CPU.

12.5 Experiments and Results

12.5.1 6-DOF Force/Torque Test

Force closure in 3-D space requires six controllable degrees of freedom in force/torque space to truly accomplish. To prove Dexterous Hexrotor can apply force/torque in all six degrees of freedom or only one degree of freedom without affecting the others, the Dexterous Hexrotor is bolted on an ATI 6-DOF force sensor and tried to produce forces and torques in all three dimensions. The test setup and ATI force sensor software are as shown in Fig. 12.18.

During the test, unit forces and torques in the force/torque vector $[F_x F_y F_z \tau_x \tau_y \tau_z]^T$ were commanded sequentially for around 5 s each at near hover condition. For 5 s with a five-second off after, each of the script requests a force in X, then Y, and then Z axis. Then it requests a torque about X, then Y, and then Z axis. The result of the test is as shown in Fig. 12.19 and tabulated in Table 12.1.

In this plot, the first 5 s, all motors are off. It is apparent from F_z that at roughly 5 s, Dexterous Hexrotor “takes off” and then the sequence begins. Data was recorded by an ATI 6-DOF force sensor at 10 kSamp/s. Time is in milliseconds.

In Table 12.3, T1 is from 5 to 10 s, which shows half throttle. T2 is from 10 to 15 s, when Dexterous Hexrotor is generating a force in x axis. T3 is from 15 to 20 s, when Dexterous Hexrotor is generating a force in y axis. T4 is from 20 to 25 s, when Dexterous Hexrotor raises its throttle. T5 is from 25 to 30 s, when Dexterous Hexrotor is generating a torque around x axis. T6 is from 30 to 35 s, when Dexterous Hexrotor is generating a torque around y axis. T7 is from 35 to 40 s, when Dexterous Hexrotor is generating a torque around z axis. This data was produced by taking the time average of the central 3 s of each period.

In Table 12.4, error of each force is compared to the magnitude of required force vector. For example, during T1, required force vector in Z axis has a magnitude of 11.56 N. F_x and F_y should be zero during T1, but they are not. So values of F_x and F_y are compared to the magnitude of the force vector and get percentage error of $0.08/11.56 = 0.69\%$ and $0.04/11.56 = 0.34\%$. It's same in T4, T5, T6, and T7, but different in T2 and T3. During T2 and T3, required force vector points along a direction between Z axis and X or Y axis. Since errors are no more than 1 %, it proves it can actually control force in each axis accurately with no coupling to the other axes.

In Table 12.5, error of each torque compared to its peak value is checked. For example, τ_x during T1 should be 0, as it is given a force/torque command of $[0 \ 0 \ F_z \ 0 \ 0 \ 0]^T$, producing only a force vector in Z axis. So comparing the value of τ_x during this period to its peak value happened in T5, its error is $0.01/1.13 = 0.8\%$. Values of torques are much smaller than the magnitude of force vector, so their percentage error looks much larger. But still an error of torques at most 6.6 % can be observed, proving that it can actually control torque in each axis with no coupling to the other axes.

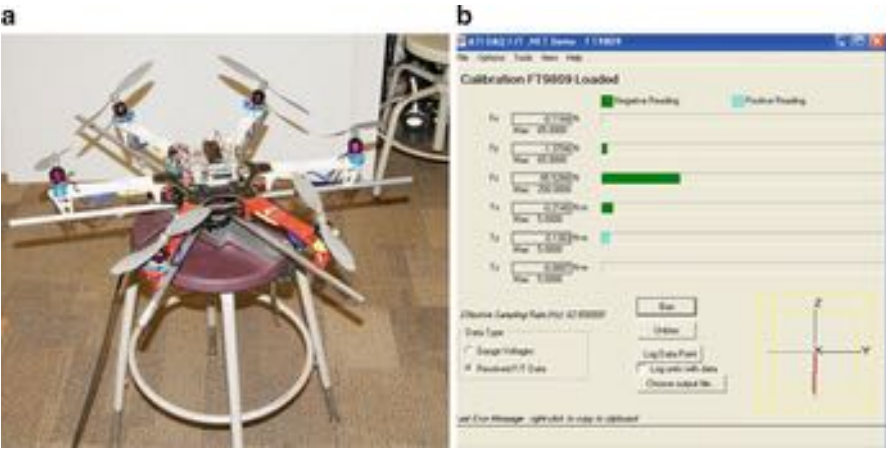


Fig. 12.18 Force/torque test setup: (a) Dexterous Hexrotor bolted on an ATI 6-DOF force sensor. (b) ATI 6-DOF force sensor software

With the ability of controlling force/torque in each axis with no coupling to the other axes, it shows full controllability over six degree of freedoms of Dexterous Hexrotor and linear independence between them.

12.5.2 Exert Forces Test

To test the response time of the Dexterous Hexrotor corresponding to external forces, a staged peg-in-hole task is presented with Dexterous Hexrotor and a normal quadrotor. A frame diagram in Fig. 12.20 shows the experiment’s setup. The peg was held rigidly by the UAV. With the peg trapped halfway in a hole which is attached to the 6-DOF ATI force sensor, the UAV starts to take off and increases its throttle until it can carry its own weight. Then it exerts a horizontal force perpendicular to the axis of the peg.

The result is shown in Fig. 12.21. Changes in force sensor measurements of F_y are direct measurements of the force which Dexterous Hexrotor and quadrotor applied on the hole. A positive pulse and a negative pulse are detected around 500 and 1,500 ms. At the meantime, the attitude of Dexterous Hexrotor was recorded and plotted in the same figure.

When Dexterous Hexrotor is exerting a force in Y axis, its attitude, especially pitch angle, does not change more than 1° . This proves Dexterous Hexrotor is exerting a force in Y axis without pitching.

When the quadrotor is trying to apply the same force, it pitches about -20° and 20° . This is because quadrotor can only generate a horizontal force by tilting an angle and Dexterous Hexrotor can do it by simply changing the rotational velocity of the rotors without tilting.

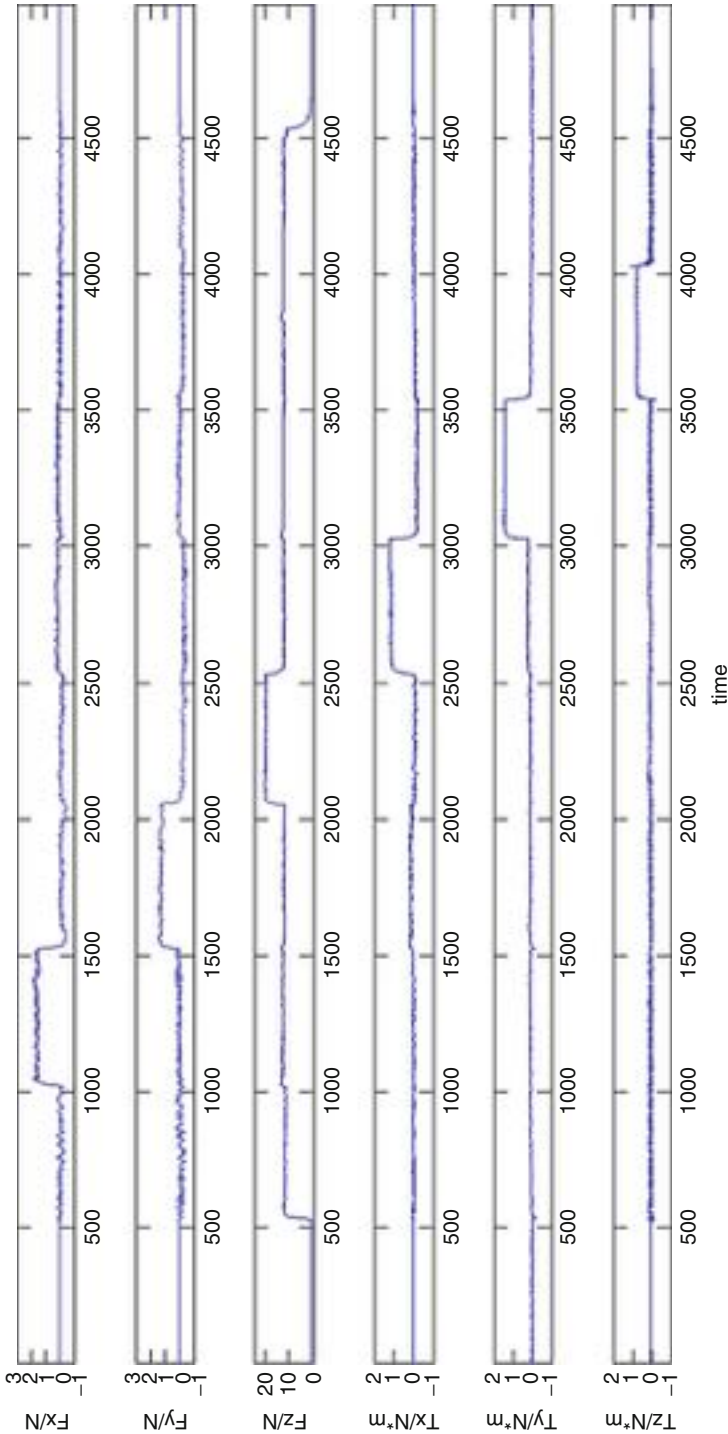


Fig. 12.19 Force/torque plot recorded by an ATI 6-DOF force sensor

Table 12.3 Average value of each period in the force/torque plot

| | T1 | T2 | T3 | T4 | T5 | T6 | T7 |
|------------------|-------|-------|-------|-------|-------|-------|-------|
| $F_x(N)$ | −0.08 | 1.59 | −0.07 | −0.09 | 0.06 | 0.03 | −0.03 |
| $F_y(N)$ | −0.04 | 0.02 | 1.30 | −0.06 | −0.05 | 0.03 | −0.04 |
| $F_z(N)$ | 11.56 | 11.96 | 11.93 | 19.94 | 11.67 | 11.43 | 11.59 |
| $T_x(N \cdot m)$ | −0.01 | 0 | 0.06 | 0 | 1.13 | −0.07 | −0.05 |
| $T_y(N \cdot m)$ | 0.04 | 0.08 | 0.08 | 0.08 | 0.09 | 1.44 | 0.07 |
| $T_z(N \cdot m)$ | 0.04 | 0.04 | 0.04 | 0.03 | 0.03 | 0.05 | 0.75 |

Table 12.4 Percentage error of each force compared to the magnitude of force vector

| (%) | T1 | T2 | T3 | T4 | T5 | T6 | T7 |
|-------|------|------|------|------|------|------|------|
| F_x | 0.69 | – | 0.58 | 0.45 | 0.51 | 0.26 | 0.26 |
| F_y | 0.34 | 0.17 | – | 0.30 | 0.43 | 0.26 | 0.35 |
| F_z | – | – | – | – | – | – | – |

Table 12.5 Percentage error of each torque compared to its peak value

| (%) | T1 | T2 | T3 | T4 | T5 | T6 | T7 |
|-------|-----|-----|-----|-----|-----|-----|-----|
| T_x | 0.8 | 0 | 5.3 | 0 | – | 6.2 | 4.4 |
| T_y | 2.7 | 5.4 | 5.4 | 5.4 | 6.2 | – | 4.8 |
| T_z | 5.3 | 5.3 | 5.3 | 4.0 | 4.0 | 6.6 | – |

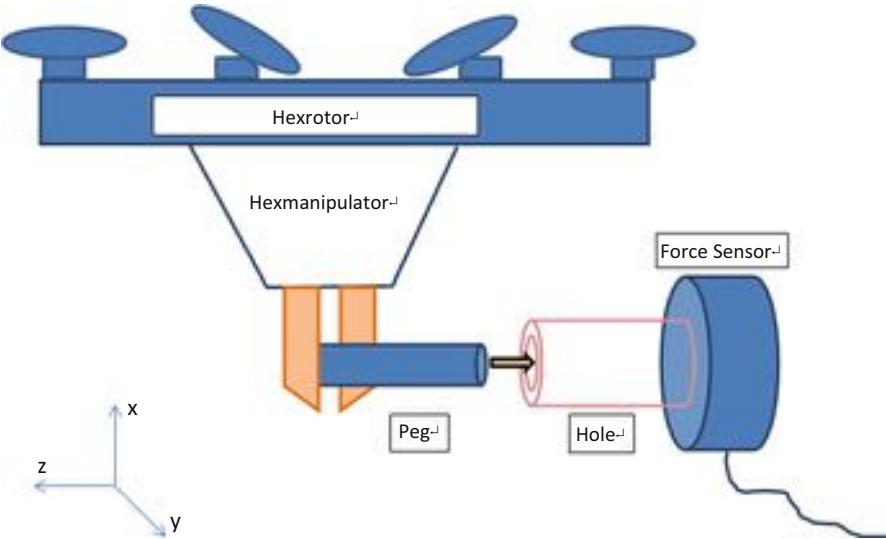


Fig. 12.20 Peg-in-hole setup diagram, with force sensor’s coordinate system indicated

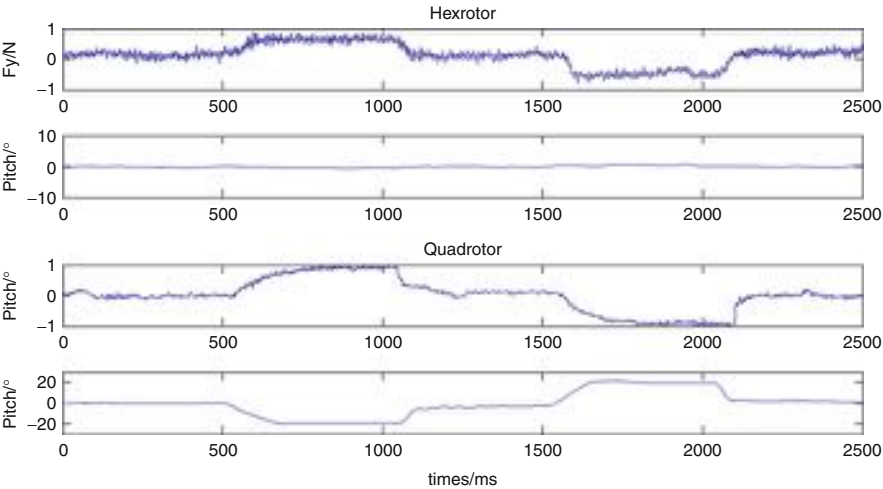


Fig. 12.21 Dexterous Hexrotor and quadrotor measurements of F_y and pitch

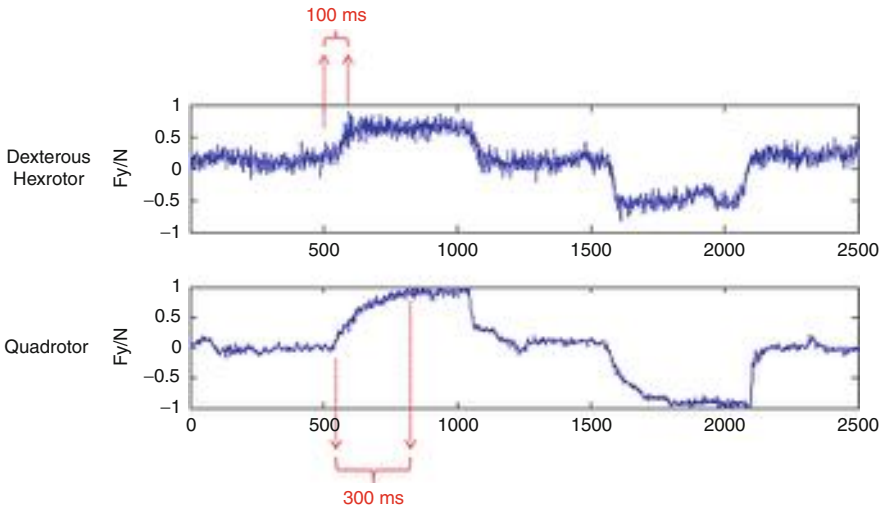


Fig. 12.22 Dexterous Hexrotor and quadrotor measurements of F_y

Therefore, as shown in Fig. 12.22, when it needs to exert an external force, it is obvious that the lag varying speeds of propellers of Dexterous Hexrotor will be much smaller than that caused by tilting an angle of a quadrotor. This is Dexterous Hexrotor’s advantage to typical quadrotors.

Table 12.6 Difference between Dexterous Hexrotor and typical quadrotor

| | Quadrotor | Dexterous Hexrotor |
|------------------------------------|--------------|--------------------|
| Number of actuators | 4 | 6 |
| Cant angle of each actuator | 0° | 0°–90° |
| DOFs in force/torque space | 4 | 6 |
| Controllable DOFs | 4 | 6 |
| Ability to achieve force closure | No | Yes |
| Mobility in 3-D space | Nonholonomic | Holonomic |
| Response of exert horizontal force | Slow | Fast |

12.6 Conclusion

This chapter introduces a truly holonomic aerial rotorcraft that provides force closure for controlled interaction with external structures. The difference between Dexterous Hexrotor and quadrotor is listed in Table 12.5. The key contribution of this chapter is to derive the mapping from actuator space to force/torque space based on decomposed net force/torque and develop a metric for the optimization of Dexterous UAV's performance for manipulation-based tasks. A flight-capable prototype with translational force control has been built and tested (Table 12.6).

It should be noted that Dexterous Hexrotor can “instantaneously resist arbitrary forces” is not strictly true as the Dexterous Hexrotor can only change the torque of its motors instantaneously. The required change in the thrust magnitude is not dependent on torque, for a propeller, but on speed. Therefore, the independent thrust magnitudes and the resulting net force/torque experience a lag due to the inertia of the thrusters. The lag due to the rotor inertia is much smaller than that due to pitching the entire vehicle, as for a conventional quadrotor, and is smaller than the pitching of the variable cant angle. To achieve force closure, we cant all thrusters of Dexterous Hexrotor at an angle and make all thrust vectors nonparallel and vertical. This would definitely influence the power efficiency of the UAV. When typical quadrotor hovers, all its power is used to combat gravity. But for Dexterous Hexrotor, some of its power will be used to cancel out each rotor's thrust. This is the main drawback of nonparallel design.

References

- A. Albers, S. Trautmann, T. Howard, T. Nguyen, M. Frietsch, C. Sauter, CIS&RAM 2010: semi-autonomous flying robot for physical interaction with environment, in *IEEE International Conference on Robotics Automation and Mechatronics*, Singapore, June 2010, pp. 441–446
- M. Bernard, A. Ollero, I. Maza, K. Kondak, J. Intell. Robot. Syst. **57**, 417–449 (2010)
- A. Bicchi, Int. J. Robot. Res. **14**, 319–334 (1995)
- A.M. Dollar, D.R. Bersak, P.E. Pounds, ICRA 2011: grasping from the air: hovering capture and load stability, in *IEEE International Conference on Robotics and Automation*, Shanghai, May 2011, pp. 2491–2498
- R. Holmberg, O. Khatib, J. Robot. Res. **19**, 1066–1074 (2000)

- C. Korpela, P.Y. Oh, TePRA 2011: designing a mobile manipulator using an unmanned aerial vehicle, in *IEEE International Conference on Technologies for Practical Robot Applications*, Boston, Apr 2011
- D. Langkamp, G. Roberts, A. Scillitoe, I. Lunnon, A. Llopis-Pascual, J. Zamecnik, S. Proctor, M. Rodriguez-Frias, M. Turner, A. Lanzon, W. Crowther, IMAV 2011: an engineering development of a novel hexrotor vehicle for 3D applications, in *International Micro Air Vehicle Conference and Competitions*, 't Harde, Sept 2011
- M.T. Mason, J.K. Salisbury, *Robot Hands and the Mechanics of Manipulation* (MIT, Cambridge, 1985)
- D. Mellinger, M. Shomin, N. Michael, V. Kumar, DARS 2010: cooperative grasping and transport using multiple quadrotors, in *International Symposium on Distributed Autonomous Robotic Systems*, Lausanne, Nov 2010. (Springer, Heidelberg, 2013), pp. 545–558
- V.-D. Nguyen, *Int. J. Robot. Res.* **7**(3), 3–16 (1988)
- C.B. Pham, S.H. Yeo, G. Yang, M.S. Kurbanhusen, I.-M. Chen, *Mech. Mach. Theory* **41**, 53–69 (2006)
- F. Reuleaux, *The Kinematics of Machinery* (Dover, New York, 1963)
- E. Rimon, J. Burdick, ICRA '96: on force and form closure, in *IEEE International Conference on Robotics and Automation*, Minneapolis, vol. 2, Apr 1996, pp. 1795–1800
- M. Ryll, H.H. Bühlhoff, P.R. Giordano, ICRA 2012: modeling and control of a quadrotor UAV with tilting propellers, in *IEEE International Conference on Robotics and Automation*, Saint Paul, May 2012, pp. 4606–4613
- S. Stramigioli, A. Keemink, M. Fumagalli, R. Carloni, ICRA 2012: mechanical design of a manipulation system for unmanned aerial vehicles, in *IEEE International Conference on Robotics and Automation*, Saint Paul, May 2012, pp. 3147–3152
- T. Yoshikawa, *J. Robot. Res.* **4**(2), 3–9 (1985)

Section III

UAV Fundamentals

Isaac I. Kaminer

Kimon P. Valavanis and George J. Vachtsevanos

The first two sections of the handbook have introduced the reader to UAVs in general and in particular to their design principles, paving the way for this third section, UAV Fundamentals, which details those fundamentals that constitute the foundational elements of UAV design technologies. This section covers kinematics and dynamics for fixed-wing, rotorcraft, quadrotors and flapping-wing MAVs, as well as *principles* of UAV guidance, navigation, and control.

► [Kinematics and Dynamics of Fixed-Wing UAVs](#) by Dobrokhodov provides a thorough review of the fundamentals required for accurate mathematical modeling of flight of a fixed-wing UAV, including kinematics and dynamics of motion and transformation of forces and moments acting on the airplane. The main objective is to familiarize the reader with the “kinematics–dynamics–actions” triad as it applies to a generic fixed-wing UAV. Emphasis is given to the understanding of reference frames and their dynamics as this is essential for the design of the UAV guidance, navigation, and control systems.

► [Dynamic Model for a Miniature Aerobatic Helicopter](#) by Gavrillets presents a nonlinear dynamic model of a miniature aerobatic helicopter following a component buildup to derive the model using simplified analytical expressions for the component forces and moments. Key parameters are estimated based on flight-test experiments. The derived model is used to design control logic for aerobatic maneuvers performed entirely under computer control.

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

► **Quadrotor Kinematics and Dynamics** by Powers, Mellinger, and Kumar presents an overview of the rigid body dynamics of a quadrotor, starting from describing the Newton–Euler equations that govern the quadrotor motion. This is followed by the derivation of a linear controller based on a linearized model of the dynamics and a nonlinear controller derived from the original dynamic model. Experimental results that illustrate the dynamics and control of small quadrotors are also presented.

► **Dynamics and Control of Flapping Wing MAVs** by Doman, Oppenheimer, and Sigthorsson focuses on presenting methods suitable for hover-capable flapping-wing vehicles. This is a challenging problem as flapping-wing motion is characterized by seven wing motion parameters like wing stroke amplitude, symmetric and asymmetric frequency, bias, upstroke and downstroke angle-of-attack limits, and wing stroke plane inclination. A method is presented that enables the designer to estimate how the seven wing motion parameters change in response to changes in the control mechanism position, followed by a method for estimating the combined effect of fuselage motion and flapping-wing motion. Control design considerations and simulation methods are also discussed.

► **Principles of Guidance, Navigation, and Control of UAVs** by Elkaim, Pradipta Lie, and Gebre-Egziabher presents two easily reconfigurable system architectures for guidance, navigation, and control of small UAVs. The presented system architectures integrate a low-cost inertial measurement unit, a GPS receiver, and a triad of magnetometers to generate a navigation solution which, in turn, is used in the guidance and control algorithms. The full system architecture – the hardware, software, and algorithms – is included for completeness. Hardware in the loop simulation and flight-test results documenting the performance of these two systems is given.

With this background, the reader, novice or expert, is now well prepared to study “specifics” and those constituent technologies, such as sensors and sensing strategies, propulsion, control, and communications, to name a few, which will enable him/her to gain a thorough understanding of the subject or explore key areas of interest.

Vladimir Dobrokhodov

Contents

| | |
|--|-----|
| 14.1 Introduction | 244 |
| 14.2 Reference Frames and Coordinate Transformations | 245 |
| 14.2.1 Kinematics of Moving Frames | 246 |
| 14.2.2 Generalized Motion | 251 |
| 14.2.3 Coordinate Frames | 253 |
| 14.3 Rigid Body Dynamics | 261 |
| 14.3.1 Conservation of Linear Momentum | 261 |
| 14.3.2 Conservation of Angular Momentum | 264 |
| 14.3.3 Complete Set of 6DoF Equations of Motion | 267 |
| 14.4 Forces and Moments Acting on the Airplane | 268 |
| 14.4.1 Gravitation | 268 |
| 14.4.2 Propulsion | 269 |
| 14.4.3 Unsteady Atmosphere | 270 |
| 14.4.4 Aerodynamics | 270 |
| 14.5 Accounting for the Earth Rotation Rate | 274 |
| 14.6 Conclusion | 277 |
| References | 277 |

Abstract

This chapter provides a review of basic knowledge required for accurate mathematical modeling of flight of a fixed-wing UAV. These include the kinematics and dynamics of motion, and the transformation of forces and moments acting on the airplane. The detailed discussion of the “kinematics–dynamics–actions” triad in application to a generic fixed-wing UAV is the main objective of this chapter. Therefore, the presentation starts with an introduction to the coordinate frames,

V. Dobrokhodov
Mechanical and Aerospace Engineering Department, Naval Postgraduate School, Monterey,
CA, USA
e-mail: vldobr@nps.edu

their transformations, and differential rotations. Kinematics of the coordinate frames is what connects states of a fixed-wing UAV and transforms forces and moments acting in different coordinate frames. Understanding of reference frames and their dynamics is essential for the guidance, navigation, and control systems design. Next, the chapter provides a detailed derivation of the equations of motion using the Newtonian approach. Assuming that a fixed-wing UAV can be represented as a rigid body moving in an inertial space allows for the derivation of the linear and angular momentum equations. Starting in an inertial frame, it is shown how the final form of translational and rotational equations of motion becomes written in a body-fixed coordinate frame. The development of both the kinematic and dynamic equations is carried out first in a general vector form; then, using simplifying assumptions applicable to a generic fixed-wing symmetric UAV, the vector equations are expanded into a scalar form to better represent the resulting simplification and the physical meaning of the remaining components. Finally, the chapter presents the principles of defining the external forces and moments acting on a generic fixed-wing airplane. Since the forces and moments found on an airplane act in a number of coordinate frames including inertial, body-fixed, and wind frames, the chapter utilizes the concepts and tools built in the kinematics section to transform the forces and moments into the body-fixed frame. Such transformations complete the presentation of the “kinematics–dynamics–actions” triad.

14.1 Introduction

The chapter objective is to provide an overview of the necessary theoretical material to enable a reliable mathematical modeling of the free and controlled motion of a generic fixed-wing UAV. Although the subject is not new and is well presented in existing literature, the rapid advancements of the last decade in research and development of fixed-wing UAV technologies open new applications that require understanding and a careful application of the existing assumptions. New materials, novel structural designs, new aerodynamic configurations, and advanced onboard instrumentation including miniature sensors, actuators, and tremendous onboard processing power enable much wider operational envelop of fixed-wing UAVs and significantly higher utility of their payloads. Depending on the UAV configuration and its intended operational use, the standard 12 equations of motion might not suffice for the task at hand and require deeper consideration of the UAV components interaction.

This chapter starts with some preliminaries required to describe kinematics of a rigid body motion in three-dimensional (3D) space. Thus, the kinematics of 3D rotation is introduced first. The most commonly used coordinate frames that are utilized in the description of UAV states are presented next. Applying the kinematics of rotating frames to a set of specific coordinate frames builds the basis for a convenient description of the external forces and moments acting on a fixed-wing airplane. Understanding of reference frames and their rotations is essential for the eventual development of the guidance, navigation, and control systems architecture.

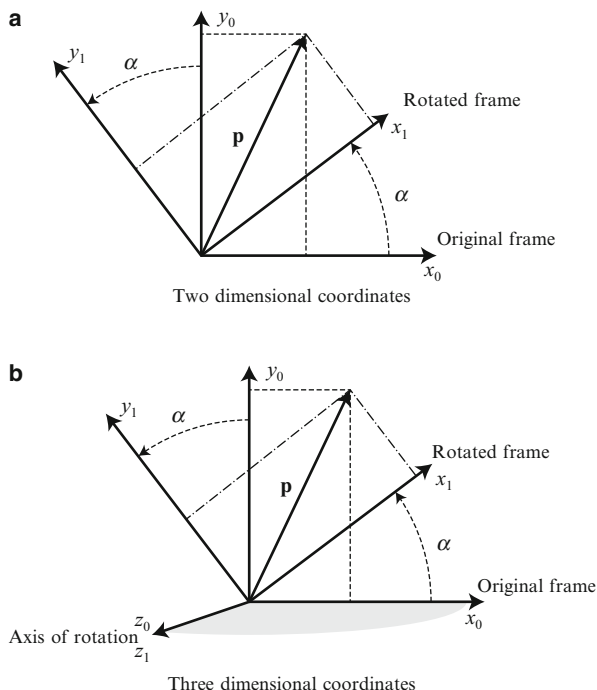
Next, the chapter provides a detailed derivation of the equations of motion using the classical Newtonian approach. Assuming that a fixed-wing UAV can be represented as a rigid body moving in an inertial space allows for the derivation of the linear and angular momentum equations. Starting in an inertial frame, it is shown how the final form of translational and rotational equations of motion can be written in a body-fixed coordinate frame. The development of both the kinematic and dynamic equations is carried out first in a general vector form, and then, using simplifying assumptions applicable to a generic fixed-wing symmetric UAV, the vector equations are expanded into a scalar form to better highlight the physical meaning of their components. Since the external forces and moments found on an airplane act in a number of coordinate frames including inertial, body-fixed, and wind frames, the chapter utilizes the concepts and tools built in the kinematics description to transform the forces and moments into the body-fixed frame. Thus, the complete derivation of linear and angular momentum equations, along with the forces and moments acting on a rigid body, results in the generalized set of 6 degrees of freedom (6DoF) equations of motion.

14.2 Reference Frames and Coordinate Transformations

In order to accurately describe a body motion, it is required to define (i) the forces and moments acting on the body and thus resulting in the body motion and (ii) the coordinate system that can be used as a reference for the motion states definition. It is important to note that there are two types of forces acting on a body in free motion. First, the inertial forces and moments that depend on the velocities and accelerations relative to an inertial reference frame; the classical Newtonian dynamics equations hold only in the inertial frame. The second group consists of the aerodynamic forces and moments resulting from interaction of the body with the surrounding airflow and therefore relative to the air. Since the airflow might not be stationary, it is therefore convenient to describe the resulting aerodynamics in the coordinate frames connected to the body and to the surrounding air. The resulting motion can be conveniently described in terms of the position, velocity, acceleration, and attitude coordinates which comprise the states of the moving body. Some of these states, in turn, need to be defined with respect to a reference frame which choice is defined by the specifics of the UAV application. Thus, the information carried by various reference frames is what facilitates the complete and convenient definition of the free body motion.

Therefore, this section starts with a definition of a coordinate frame and the description of the coordinate frame rotation. The reference frames required to represent the aerodynamic forces and moments and facilitating the solution of the navigation states are introduced next. Communication of the states information occurring during the coordinate frame transformation is presented for the major coordinate frames typical for UAV applications. The section ends with a set of kinematic equations required to represent the transition of linear and angular accelerations.

Fig. 14.1 The same plane rotation considered with respect to two and three axes. (a) Two-dimensional coordinates. (b) Three-dimensional coordinates



14.2.1 Kinematics of Moving Frames

The objective of this subsection is to define a coordinate frame transformation and the associated mathematical formalism. Namely, the direct cosine matrix (DCM) is introduced, and its key properties are presented. The DCM matrix formalism is then followed by a differential rotation that defines the rate of change of the DCM matrix. A fundamental property of the simple summation of angular rates is introduced next. The section ends with a detailed presentation of the coordinate frames used to describe the 6DoF motion of a rigid body. The results of this development are heavily utilized throughout the entire chapter.

An arbitrary motion of a rigid body can be described by a transformation that consists of (Goldstein 1980) translational and rotational components. First, a pure rotation of a rigid body is addressed. Consider a vector \mathbf{p} defined in two orthogonal coordinate frames rotated with respect to their mutual origin by angle α , as shown in Fig. 14.1a.

From this geometrical setup, it can be demonstrated that vector $\mathbf{p} = [x_0, y_0]$ can be uniquely defined in both frames as follows:

$$\begin{aligned} x_1 &= x_0 \cos \alpha + y_0 \sin \alpha \\ y_1 &= -x_0 \sin \alpha + y_0 \cos \alpha \end{aligned} \quad (14.1)$$

where 0 and 1 subscripts refer to the coordinates of \mathbf{p} in the original and rotated frames correspondingly.

Introducing the matrix notation for the linear transformation above results in a simple form that relates the vector \mathbf{p} components in (x_0, y_0) frame to the corresponding components in (x_1, y_1) frame:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = R_0^1 \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad R_0^1 = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}. \quad (14.2)$$

The resulting rotation matrix is called a DCM matrix. The DCM matrix R_0^1 consists of the cosine and sine functions which are the direction cosines between the matching axes of the new and old coordinate systems denoted in the superscript and the subscript correspondingly. Following the same approach, it can be demonstrated that for the case of right-handed coordinate system represented by three orthogonal axes (see Fig. 14.1b), the same right-hand rotation results in transformation

$${}_z R_0^1 = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (14.3)$$

where for clarity, the subscript z denotes the axes of rotation. Proceeding similarly, right-handed rotations of the coordinate frame about the y_0 and x_0 axis give

$${}_y R_0^1 = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix}, \quad {}_x R_0^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}. \quad (14.4)$$

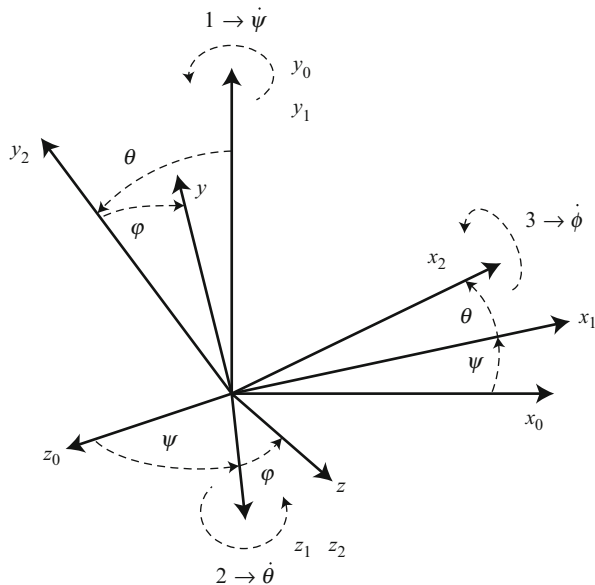
It is worth noting that the DCM transformation has the following easy-to-remember properties that simplify its application (see more details in Rogers 2003):

1. The transformed vector components along the axis of rotation remain unchanged with the rotation about that axis; elements of DCM are either 0 or 1.
2. The remaining elements of DCM are either sin or cos functions of the angle of rotation.
3. The cos elements are on the main diagonal with sin elements on off-diagonal.
4. The negative sin component corresponds to the component rotated “outside” of the quadrant formed by the original frames.
5. Columns (rows) of a DCM matrix form an orthonormal set.

It is straightforward to verify that a DCM matrix corresponding to the right-handed frames has the following properties:

$$\det(R) = 1; \quad R^T = R^{-}; \quad R^T R = I; \quad R = [c_1, c_2, c_3] \Rightarrow c_i \cdot c_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \quad (14.5)$$

Fig. 14.2 Three consecutive rotations



and therefore it belongs to a general class of orthonormal transformation matrices. For a sequence of rotations performed with respect to each orthogonal axis, the resulting transformation can be obtained by a matrix composed of three sequential rotations, called Euler angle rotations, starting from the original frame of reference; see Fig. 14.2.

Formally, this transformation is accomplished by rotating through the ordered sequence of Euler angles $[\psi, \theta, \phi]$, where the numerical indexes define the ordered sequence of rotations and the corresponding axis of rotations:

$$R_{x_0}^y = R_{x_2}^y R_{x_1}^{x_2} R_{x_0}^{x_1} \quad (14.6)$$

It is worth mentioning that the corresponding Euler angles are also widely used to express elementary rotation matrices so that in (14.6), the following notation $R_\phi = R_{x_2}^y, R_\theta = R_{x_1}^{x_2}, R_\psi = R_{x_0}^{x_1}$ is possible. As an example, $R_\psi = R_{x_0}^{x_1}$ defines a rotation of the axis x_0 to x_1 and z_0 to z_1 performed with respect to the axis y_0 by the angle ψ . From now, the same approach to denoting the rotations and vectors is used throughout this chapter. In the case of rotations, the subscript refers to the original frame while the superscript refers to the rotated frame; in the case of vectors denoted in bold face, the subscript defines the frame where the vector is resolved and the superscript refers to a specific meaning of the vector when necessary, and the frames are indicated by the brace as in $\{frame\}$. Therefore, a vector $\mathbf{p}_0 = [x_0, y_0, z_0]^T$ given in $\{0\}$ coordinate frame can be resolved in another coordinate frame $\{1\}$ of arbitrary orientation with respect to the original frame by a transformation matrix R_0^1 composed of three sequential rotations as follows:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R_0^1} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (14.7)$$

$$R_0^1 = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \theta \sin \psi + \sin \phi \sin \theta \cos \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\ \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{bmatrix} \quad (14.8)$$

This matrix, which represents a transformation resulting from three sequential Euler angle rotations, will be used throughout the chapter.

Overall, any rotation matrix has a number of properties. They are summarized here for completeness; an interested reader is referred to references (Goldstein 1980; Murray et al. 1994) for thorough details:

- Rotation matrices are orthogonal.
- The determinant of a rotation matrix is unity.
- Successive rotations can be represented by the ordered product of the individual rotation matrices.
- Rotation matrices are not commutative; hence, in general case, $R_b^c R_a^b \neq R_a^b R_b^c$.
- A nontrivial rotation matrix has only one eigenvalue equal to unity with other two, being a complex conjugate pair with unity magnitude; a trivial rotation is described by an identity matrix.

The time rate of change of the DCM matrix that defines the dynamics of the attitude states is important in derivation of the kinematic equations of motion. As it will be shown shortly, it enables relating the sensor (e.g., given by the rate gyros) measurements obtained in a body-fixed frame to the time derivatives of the Euler angles describing the attitude of a body in an inertial frame.

In general case, the time derivative of a rotation matrix that is considered as a function of time can be obtained based on its key properties. Let $R(t) = R_0^1(t)$ be a rotation matrix given as a function of time. Since $R \cdot R^T = I$, taking the time derivative of both sides yields

$$\frac{d(R \cdot R^T)}{dt} = \dot{R} \cdot R^T + R \cdot \dot{R}^T = \dot{R} \cdot R^T + (\dot{R} \cdot R^T)^T = \mathbf{0}.$$

First, it can be observed that $\dot{R} \cdot R^T = S$ is a skew symmetric matrix. Next, let $\mathbf{p}_1 = R(t)\mathbf{p}_0$, where \mathbf{p}_0 is a vector of constant length rotated over time with an angular velocity vector $\boldsymbol{\Omega}$. Comparing two expressions of the absolute time derivatives of \mathbf{p}_1

$$\dot{\mathbf{p}}_1 = \dot{R}(t)\mathbf{p}_0 = S(t)R(t) \cdot \mathbf{p}_0 = S(t) \cdot \mathbf{p}_1,$$

$$\dot{\mathbf{p}}_1 = \boldsymbol{\Omega} \times \mathbf{p}_1 = S(\boldsymbol{\Omega}(t)) \cdot \mathbf{p}_1$$

leads to

$$\dot{R} = S(\mathbf{\Omega})R, \quad S(\mathbf{\Omega}) = \dot{R}R^T. \quad (14.9)$$

Thus, the skew symmetric matrix $S(\mathbf{\Omega})$ in (14.9) is used to represent the vector cross product between the vectors $\mathbf{\Omega}$ and the $R(t)\mathbf{p}_0$. The matrix $S(\mathbf{\Omega})$, where vector $\mathbf{\Omega} = [\omega_x, \omega_y, \omega_z]^T$ is represented by its components, can be written in the form

$$S(\mathbf{\Omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix}.$$

Another useful general property of angular velocities is called the *angular velocities addition theorem* (Rogers 2003). The theorem states that for angular velocity vectors coordinated in a common frame, the resulting angular velocity of the cumulative rotation is a plain sum of the contributing rotations. Now, if a rotating frame is given by a set of time varying Euler angles $[\psi, \theta, \phi]$ defined with respect to a stationary frame, then it is straightforward to determine the components of the angular velocity vector $\mathbf{\Omega} = [\omega_x, \omega_y, \omega_z]^T$ as though measured in the rotating frame. Starting from an initial stationary frame (see Fig. 14.2) and using two intermediate frames whose relative angular velocities are defined by the Euler angle rates $[\dot{\psi}, \dot{\theta}, \dot{\phi}]$ and utilizing the angular velocities addition theorem, the following kinematic equation can be obtained:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = R_\phi R_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}. \quad (14.10)$$

Substituting the corresponding DCM matrices from (14.7) results in

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (14.11)$$

Inverting the last equation results in equation

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \frac{\sin \theta}{\cos \theta} & \cos \phi \frac{\sin \theta}{\cos \theta} \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \frac{1}{\cos \theta} & \cos \phi \frac{1}{\cos \theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (14.12)$$

that defines the derivatives of the Euler angles in terms of the angles themselves and the angular velocity vector $\mathbf{\Omega} = [\omega_x, \omega_y, \omega_z]^T$ as it was measured in the rotated frame. These equations define the rotational kinematics of a rigid body; they contribute to the final set of 6DoF equations of motion.

Analysis of the (14.12) shows that four elements of the inverted matrix become singular when the second rotation angle θ approaches $\pi/2$. This problem is usually called a *kinematic singularity* or a *gimbal lock* in navigation and is one of the

issues associated with the use of Euler angles for the attitude determination. For differently ordered Euler rotation sequences, the kinematic singularity will occur at a different point. Therefore, one way to avoid the singularity is to switch or change the Euler angle sequences when approaching the singularity. Next, depending on the available computing power, the integration of the kinematic equation (14.12) can be computationally expensive because it involves calculation of trigonometric functions. Furthermore, it can be observed that the Euler angle-based DCM matrix is redundant; it requires only 3 out of 9 elements of the DCM matrix to uniquely define the Euler angles. These shortcomings usually result in applying different parameters describing the attitude and its dynamic transformation.

In applications to the fixed-wing UAV attitude determination, the Rodrigues–Hamilton parameter, or quaternion, is one of the most widely used alternatives (Goldstein 1980). Utilizing the quaternion approach is very powerful because it gives a singularity-free attitude determination at any orientation of a rigid body. Next, since it can be shown that the equations of motion of a rigid body are linear differential equations in the components of quaternion, then it (linearity) is a desirable property, especially when developing estimation and control algorithms. Furthermore, the quaternion is a relatively computationally efficient approach since it does not involve trigonometric functions to compute the attitude matrix and has only one redundant parameter, as opposed to the six redundant elements of the attitude matrix. However, it is also worth noting that the quaternion and Euler angle techniques are both widely used in various UAV applications; the equations connecting both representations are well developed, thus enabling complementary definition of the DCM matrix and Euler angles through the parameters of quaternion and vice versa. An interested reader is referred to an extensive historical survey of attitude representations (Shuster 1993) and references (Rogers 2003; Goldstein 1980; Murray et al. 1994) for more details in the alternative methods of attitude determination.

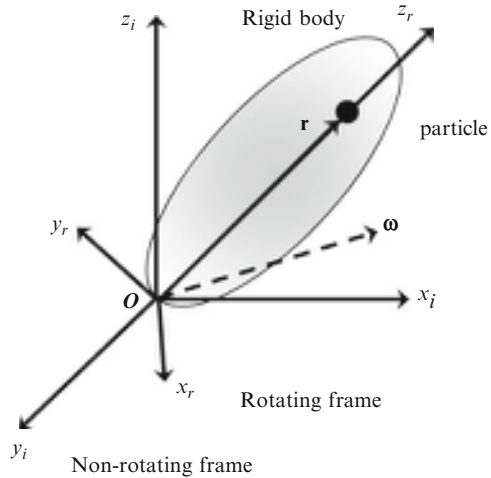
14.2.2 Generalized Motion

In the development of dynamic equations of motion, it will be necessary to calculate the absolute time derivative of a vector defined in coordinate frames that rotate and move with respect to each other. In an application to the UAV kinematics, this can be justified by a necessity to relate the absolute time derivative of a position vector in inertial space (inertial velocity) that is defined based on the measurements taken in a body frame. Similarly, the second time derivative defines the body inertial acceleration.

Consider two coordinate frames $\{F_i\}$ and $\{F_r\}$, where i stands for an inertial not rotating frame and r stands for the rotating frame. The first objective is to calculate the derivative of a unity vector \mathbf{r}_r defined in $\{F_r\}$ attached to a rigid body rotating with respect to the $\{F_i\}$ with angular speed $\boldsymbol{\omega}$; see Fig. 14.3. Denote the rotation from $\{F_r\}$ to $\{F_i\}$ as R :

$$\mathbf{r}_i = R\mathbf{r}_r$$

Fig. 14.3 Deriving the time derivative of a vector



Taking the derivative results in

$$\dot{\mathbf{r}}_i = \dot{\mathbf{R}}\mathbf{r}_r + \mathbf{R}\dot{\mathbf{r}}_r = \dot{\mathbf{R}}\mathbf{r}_r = S(\boldsymbol{\omega})\mathbf{r}_r = \boldsymbol{\omega} \times \mathbf{r}_r, \quad (14.13)$$

where the time derivative $\dot{\mathbf{r}}_r$ is zero due to the rigid body assumption.

Next, using the same setup, calculate the absolute time derivative of an arbitrary time varying vector \mathbf{r} defined in $\{F_r\}$. Defining the vector in terms of its components in both frames and taking its time derivative in the inertial frame result in

$$\mathbf{r} = r_{xi}\mathbf{i} + r_{yi}\mathbf{j} + r_{zi}\mathbf{k} = r_{xr}\mathbf{l} + r_{yr}\mathbf{m} + r_{zr}\mathbf{n}.$$

Taking the absolute time derivative of both expressions gives

$$\begin{aligned} \frac{d\mathbf{r}}{dt} &= \dot{r}_{xi}\mathbf{i} + \dot{r}_{yi}\mathbf{j} + \dot{r}_{zi}\mathbf{k} \\ \frac{d\mathbf{r}}{dt} &= \dot{r}_{xr}\mathbf{l} + \dot{r}_{yr}\mathbf{m} + \dot{r}_{zr}\mathbf{n} + r_{xr}\dot{\mathbf{l}} + r_{yr}\dot{\mathbf{m}} + r_{zr}\dot{\mathbf{n}}. \end{aligned}$$

Applying (14.13) allows rewriting the last equation as

$$\frac{d\mathbf{r}}{dt} = \frac{\delta\mathbf{r}}{\delta t} + \boldsymbol{\omega} \times \mathbf{r} = \dot{\mathbf{r}}_r + \boldsymbol{\omega} \times \mathbf{r}, \quad (14.14)$$

which expresses the derivative of the vector \mathbf{r} in the inertial frame $\{F_i\}$ in terms of its change ($\dot{\mathbf{r}}_r$) calculated in a rotating frame $\{F_r\}$ and its relative rotation defined by the angular velocity $\boldsymbol{\omega}$. The result in (14.14) is known as the *Coriolis theorem*. The second derivative of \mathbf{r} defines the body acceleration and is used in the development of the dynamic equations. It is obtained in a similar manner by recursively applying the Coriolis theorem, thus leading to

$$\begin{aligned}
\ddot{\mathbf{r}} &= \frac{\delta}{\delta t}(\dot{\mathbf{r}}_r + \boldsymbol{\omega} \times \mathbf{r}) + \boldsymbol{\omega} \times (\dot{\mathbf{r}}_r + \boldsymbol{\omega} \times \mathbf{r}) \\
&= \ddot{\mathbf{r}}_r + \frac{\delta \boldsymbol{\omega}}{\delta t} \times \mathbf{r} + \boldsymbol{\omega} \times \frac{\delta \mathbf{r}}{\delta t} + \boldsymbol{\omega} \times \dot{\mathbf{r}}_r + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}), \\
\ddot{\mathbf{r}} &= \ddot{\mathbf{r}}_r + \dot{\boldsymbol{\omega}} \times \mathbf{r} + 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_r + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}).
\end{aligned} \tag{14.15}$$

Similarly to (14.14) where $\dot{\mathbf{r}}_r$ denotes the local derivative taken in a rotating frame $\{F_r\}$, the $\frac{\delta \boldsymbol{\omega}}{\delta t}$ refers to the derivative of angular velocity $\boldsymbol{\omega}$ taken in $\{F_r\}$ frame. However, it can be easily demonstrated

$$\frac{d\boldsymbol{\omega}}{dt} = \frac{\delta \boldsymbol{\omega}}{\delta t} + \boldsymbol{\omega} \times \boldsymbol{\omega} = \frac{\delta \boldsymbol{\omega}}{\delta t} \tag{14.16}$$

that the derivative of $\boldsymbol{\omega}$ is independent on the coordinate frame. In turn, this justifies omitting the subscript in $\boldsymbol{\omega}$ referring to $\{F_r\}$. The last two terms in (14.15) are commonly known as *Coriolis* and the *centripetal accelerations* correspondingly. The following chapter heavily relies on the results in (14.14) and (14.15) when it develops the dynamic equations of motion.

14.2.3 Coordinate Frames

Deriving equations of motion of a fixed-wing UAV requires a definition of coordinate frames where forces and moments acting on the airplane can be conveniently defined and where the states including the position, velocity, acceleration, and attitude can be suitably described. It is worth noting that with the latest advances in power technologies and novel materials, a mission of long duration becomes a reality. As an example, solar power technology is one of the alternatives that can make the 24/7 flight of a fixed-wing solar powered autonomous soaring gliders feasible. Thus, the long duration and great operational distances might require considering the UAV flight operations with respect to the rotating Earth. This consideration would extend the set of coordinate frames used in long-endurance UAV applications. The primary reason for such an extension would lie in the necessity to resolve the inertial angular velocity of rotation of a body frame in a “true” inertial frame where the Earth rotation can be resolved. Those frames would include the Earth-centered inertial, Earth-centered Earth-fixed, and a geodetic coordinate frames. Consequently, the angular velocities addition theorem mentioned above would be used to resolve the angular velocity vector of body rotation with respect to the inertial frame as a vector sum of angular velocities of the intermediate frames. An interested reader is referred to (Rogers 2003) for more details in the coordinate frames and transformations used in inertial navigation.

Therefore, in addition to the body and wind frames that define dynamics of the body–fluid interaction, this subsection defines the following set of coordinate frames used in various UAV applications:

1. Earth-centered inertial frame $\{i\}$
2. Earth-centered Earth-fixed frame $\{e\}$

3. Geodetic coordinate system $\{\lambda, \varphi, h\}$
4. Tangent plane coordinate system $\{u\}$
5. Body-carried frame $\{n\}$
6. Body-fixed frame $\{b\}$
7. Wind frame $\{w\}$

Depending on the duration of flight and operational range, both dictated by a specific UAV application, the first four frames can be considered inertial frames, with the remaining three frames body fixed. The positioning of inertial and body frames is related by a plain translation, while the orientation of body frames relates to each other by pure rotations. Details of the frames definition and their relations are the subject of this section.

14.2.3.1 Earth-Centered and Geodetic Coordinate Frames

It is convenient to consider two coordinate frames connected to the Earth. The *Earth-centered Earth-fixed* (ECEF) coordinate system is fixed to the Earth, and therefore, it rotates at the Earth's sidereal rate with respect to the *Earth-centered inertial* (ECI) frame that represents nonrotating inertial frame. The ECI frame is usually denoted $\{i\}$ while the ECEF frame is denoted $\{e\}$. Both frames are right-handed orthogonal and have their origins at the center of the Earth. The ECI frame has its z_i axis aligned with the direction of the Earth's rotation vector and x_i and y_i axes placed in the equatorial plane with x_i fixed in some celestial reference direction, for example, a line connecting the Sun's center and the Earth's position at vernal equinox (Kaplan 1981). The ECEF has x_e and y_e axes placed in the equatorial plane and z_e axis aligned with z_i axis; see Fig. 14.4 where the Earth is modeled as a spheroid. The x_e axis is usually attached to the intersection of the Greenwich meridian and the equator, and the y_e axis completes the right-hand system.

It is worth noting that the ECEF axes definition may vary; however, the definition always states the attachment of two vectors to the direction of the Earth rotation and the Greenwich meridian as the inherent Earth properties. The sidereal rate Ω_i^e is the vector defining the ECEF rotation with respect to the ECI; the latter one is often called the true inertial frame. If necessary, for the purpose of UAV flight description, the magnitude Ω_i^e of the rate can be approximated by one full rotation in 23h56'4.099'', thus resulting in $\Omega_i^e = 15.04106718$ deg/h. Therefore, the transformation from ECI to ECEF frame is a plain rotation around the z_i axis defined by a single rotation by an angle $\Omega_i^e \cdot t$, where t is the time interval.

The *local geodetic* $\{\lambda, \varphi, h\}$ frame is usually associated with the ECEF frame, see Fig. 14.4. It has the same origin placed at the center of the Earth. The frame defines the orientation of a line normal to Earth's surface and passing through the point of interest. The orientation of the line is defined by two angles, λ (geodetic latitude) and φ (geodetic longitude), with the height h above the Earth's surface; eventually these three parameters, along with the components of the UAV velocity vector, become the major navigation states. For most UAV applications, it is sufficiently accurate to model Earth's surface as an oblate spheroid with given r_e (equatorial) and r_p (polar radiuses) or one of the radiuses and the e (ellipticity) (Kaplan 1981). Last revisited in 2004, the datum of World Geodetic System (WGS-84) provides

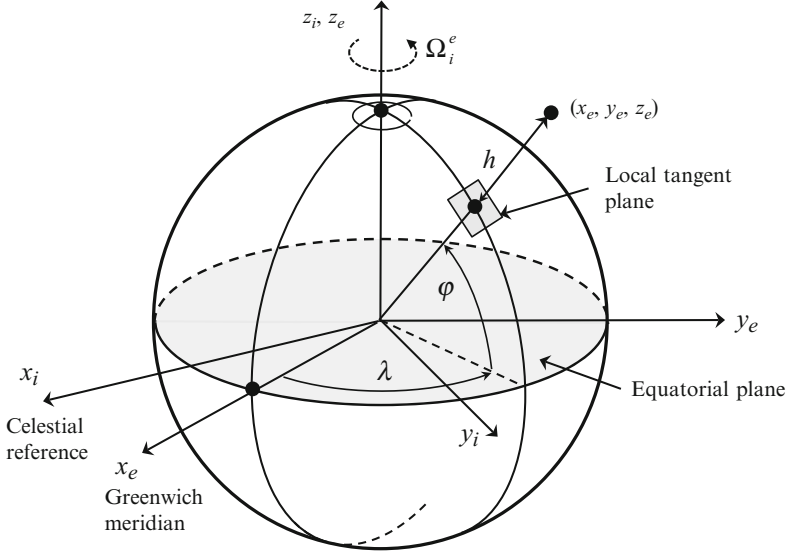


Fig. 14.4 ECI, ECEF, and geodetic coordinate frames

the following parameters for the oblate spheroid modeling: $r_e = 6,378,137.00$ m, $r_p = 6,356,752.314$ m. The resulting transformation from the geodetic $\{\lambda, \varphi, h\}$ to the ECEF frame is as follows:

$$\begin{aligned} x_e &= (r_\lambda + h) \cos \varphi \cos \lambda \\ y_e &= (r_\lambda + h) \cos \varphi \sin \lambda \\ z_e &= ((1 - \varepsilon^2)r_\lambda + h) \sin \varphi \end{aligned} \quad (14.17)$$

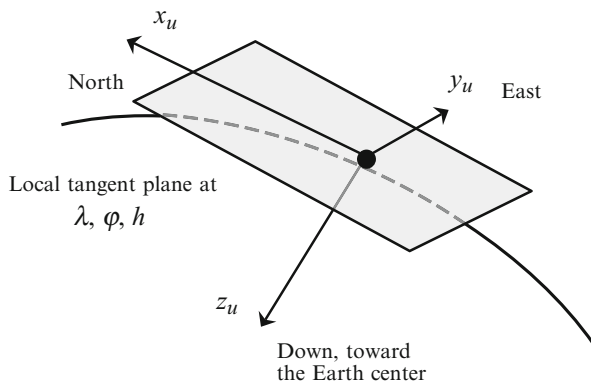
where ε the eccentricity of oblate ellipsoid is defined as

$$\begin{aligned} r_\lambda &= \frac{r_e}{\sqrt{1 - \varepsilon^2 \sin^2 \varphi}}; \\ \varepsilon &= \sqrt{1 - \frac{r_p^2}{r_e^2}}. \end{aligned}$$

14.2.3.2 Local Tangent Plane Coordinate System

The origin of the *local tangent plane* (LTP) is fixed to the surface of the Earth with two of its axes attached to the plane tangent to the surface; see Fig. 14.5. The frame is usually marked with the subscript $\{u\}$ and serves the purpose of an inertial frame in most short-duration low-speed UAV applications. The frame's x_u and y_u axes are in the tangent plane and most often aligned with the north and east directions correspondingly; the z_u axis completes the right-hand coordinate system, thus pointing down. Quite often the order and alignment of the LTP frame principal

Fig. 14.5 Definition of the local tangent plane; NED



axes change. In such cases, the LTP coordinate system explicitly specifies its type; in the nominal case presented above, it can be also defined as an NED frame indicating the north–east down alignment of the coordinate axes.

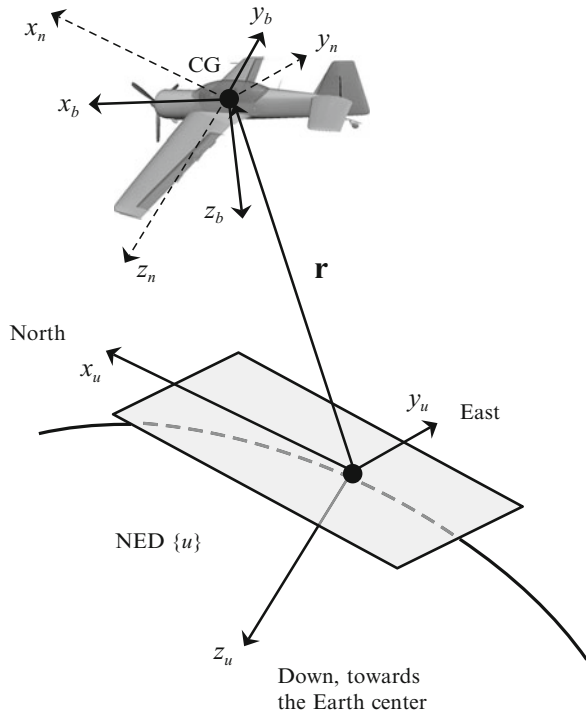
When the origin of the LTP frame is defined in terms of its geodetic latitude, longitude, and altitude above the ground surface, then the equations (14.17) can be applied to define the kinematics of navigation states.

14.2.3.3 Body-Carried and Fixed Frames

In flight dynamics, the body-attached reference frames usually have their origin at the center of gravity (CG) of an airplane; therefore, these frames are moving. The *body-carried* frame $\{n\}$ is an orthogonal frame originated at the CG of the UAV. All its axes are permanently stabilized and aligned with the LTP frame axes as it was connected to the CG; see Fig. 14.6; the frame is usually utilized in defining the navigation equations thus assigning its subscript $\{n\}$. This frame is connected to the LTP frame by means of a plain translation $\mathbf{r} = [r_n, r_e, r_d]^T$.

The *body-fixed* frame is an orthogonal frame defined with respect to the body-carried frame. Its origin is at the CG of UAV, and its axes are rigidly connected to the body; therefore, the frame also rotates with the body. The frame is usually marked with the subscript $\{b\}$. The alignment of the $\{b\}$ frame axes is a degree of freedom that can exploit the body symmetry. It can be proven (Goldstein 1980) that for every rigid body, there is always an orthogonal coordinate system, usually called principal, in which the cross products of inertia terms are zero. Assuming that a typical UAV has at least one plane of symmetry (geometric and mass symmetry) results in two of the body-fixed axes laying in the plane of symmetry. When the axes are aligned along the principal axes of inertia of the body, as will be shown in the following chapter, the dynamic equations of motion become significantly simpler. In a majority of fixed-wing UAV configurations, the axes of $\{b\}$ frame match the principal axes of inertia. The typical orientation of the body-fixed axes is as follows (see Fig. 14.6): if the UAV has a vertical plane of symmetry, then x_b and z_b lie in that plane of symmetry; x_b points toward the direction of flight and z_b points downward and y_b points right, thus completing the right-hand system.

Fig. 14.6 Definition of the body-fixed frame with respect to LTP frame



As the body moves, its attitude is defined with reference to the body-carried frame $\{n\}$ by three consecutive Euler rotations by ψ (yaw), θ (pitch), and ϕ (roll) angles. See their graphical illustration in Fig. 14.2 where frames $\{0\}$ and $\{1\}$ relate to the frames $\{n\}$ and $\{b\}$ correspondingly. The formal definition of the Euler angles in the application to an airplane attitude specification is presented here for completeness:

- ψ – yaw is the angle between x_n and the projection of x_b on the local horizontal plane.
- θ – pitch is the angle between the local horizon and the x_b axis measured in the vertical plane of symmetry of the UAV.
- ϕ – roll is the angle between the body-fixed y_b axis and the local horizon measured in the plane $y_b z_b$.

As it follows from the attitude representation section, the DCM matrix R_u^b transforming the body-carried $\{n\}$ to the body-fixed $\{b\}$ frame can be constructed as in (14.8). Now the subscripts ($u \rightarrow b$) denote the rotation from the LTP $\{u\}$ to the body-fixed frame; $\{u\}$ and $\{n\}$ frames are always aligned by the definition of body-carried frame.

The application of the rotation matrix (14.8) immediately follows from the need to describe the UAV translational motion in an inertial frame of reference by utilizing the inertial velocity measurements taken in the body-fixed frame at

$\text{CG} - \mathbf{V}_b^g = [u, v, w]^T$. To this end, consider Fig. 14.6 where vector $\mathbf{r} = [r_n, r_e, r_d]^T$ denotes the position of an airplane CG with respect to the LTP (NED) frame attached to the Earth. Relating the translational velocity and position and accounting for the fact that *body-carried* frame $\{n\}$ is stabilized with respect to the nonrotating $\{u\}$ frame results in

$$\begin{aligned} \frac{d\mathbf{r}}{dt} &= R_b^u \mathbf{V}_b^g \\ \frac{d}{dt} \begin{bmatrix} r_n \\ r_e \\ r_d \end{bmatrix} &= R_b^u \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \end{aligned} \quad (14.18)$$

The relation between the Euler angles defining the relation between the stabilized $\{n\}$ frame and the body-fixed frame $\{b\}$ was already derived in (14.11) and (14.12). They define the dynamics of Euler angles defined in an inertial frame with respect to the rates measured in the body-fixed frame. Thus, the kinematic equations (14.12) and (14.18) represent the dynamics of translational and rotational coordinates and therefore are part of the final set of equations of motion.

14.2.3.4 Wind Frame

Aerodynamic forces and moments resulting from the body–air interaction as the airframe moves through the air depend on the body orientation with respect to the surrounding air. In other words, they depend on the vector representing the wind. The velocity vector of the possibly moving air (wind) resolved in the inertial frame $\{u\}$ is denoted \mathbf{V}_u^a ; see Fig. 14.7. The magnitude of \mathbf{V}_u^a is called an *airspeed* V_a , as opposed to the velocity vector defined in LTP with respect to the ground – *ground speed* vector \mathbf{V}_u^g . The orientation of the wind frame $\{w\}$ defined by the direction of \mathbf{V}_u^a with respect to the body-fixed $\{b\}$ is defined by two angles.

To generate the lift force in flight, the wing of the UAV must be oriented at a usually positive angle α with respect to the \mathbf{V}_u^a vector. This angle is called the *angle of attack*. The angle of attack α is also one of the key parameters that define

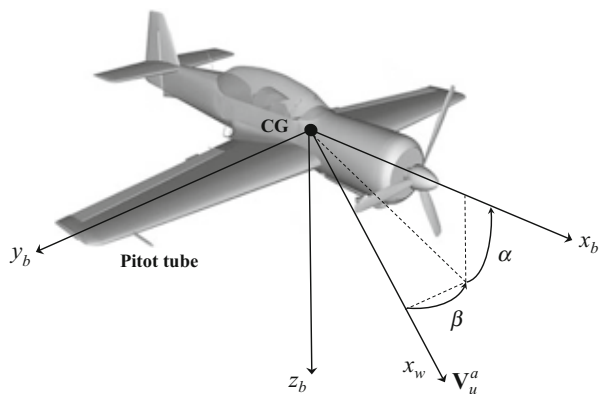


Fig. 14.7 Wind frame and body-fixed frames. Definition of the angle of attack and the sideslip

the longitudinal stability of an airplane. Therefore, quite often, the coordinate frame that results from a single rotation from the body-fixed $\{b\}$ frame on angle α is called a stability frame (Beard and McLain 2012; Etkin and Reid 1995). As illustrated in Fig. 14.7, the angle of attack is defined by the projection of \mathbf{V}_u^a into a vertical plane of symmetry of the UAV (spanned by axes x_b, z_b in frame $\{b\}$) and the longitudinal axis x_b of the UAV. It is positive when a leading edge of the wing rotates upward with respect to the \mathbf{V}_u^a . In turn, the angle between the velocity vector \mathbf{V}_u^a projected into the “wing-level” plane (spanned by axes x_b, y_b in frame $\{b\}$) and the longitudinal axis x_b of UAV is called the *sideslip angle*. It is denoted by β .

Then the transformation from the body-fixed frame $\{b\}$ to the wind frame $\{w\}$ is given by

$$\begin{aligned} R_b^w &= \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \\ &= \begin{bmatrix} \cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}. \end{aligned} \quad (14.19)$$

The inverse transformation from the wind frame $\{w\}$ to the body-fixed frame $\{b\}$ is the transpose of (14.19): $R_w^b = (R_b^w)^T$.

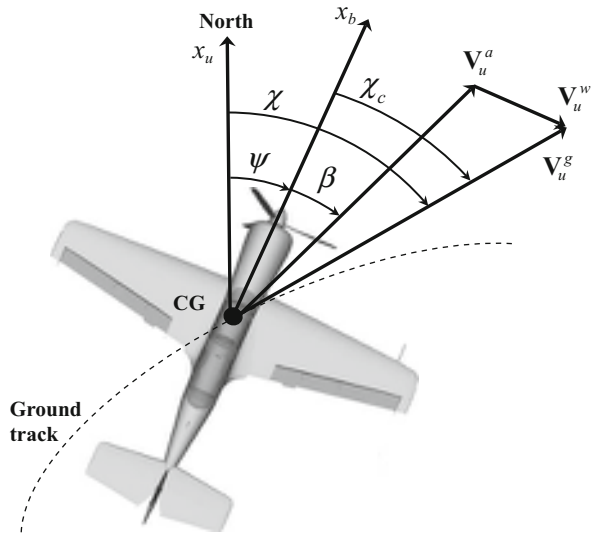
The importance of the wind frame in application to the UAVs flying in wind conditions that might contribute up to 100% of the nominal airplane speed cannot be overestimated. As an example, imagine an autonomous glider that is designed to utilize the wind energy to sustain the long duration of flight. Therefore, it is necessary to understand the difference between the airspeed, represented by the air velocity vector \mathbf{V}_u^a and the ground speed \mathbf{V}_u^g , both resolved with respect to the LTP frame. Consider the graphical representation of the relation between these vectors in Fig. 14.8. In the presence of constant wind, these velocities are related by the equation that is often called the *wind triangle*:

$$\mathbf{V}_u^a = \mathbf{V}_u^g - \mathbf{V}_u^w \quad (14.20)$$

where \mathbf{V}_u^w is the wind velocity defined in the LTP frame.

The objective of the following development is to define the relations among these velocities defined in three different frames, while being measured or estimated by the sensors installed in the body-fixed and in the inertial frame. First, define the components of all three vectors in the body-fixed frame $\{b\}$. Let the UAV velocity in the LTP (inertial) frame expressed in the body frame be $\mathbf{V}_b^g = [u, v, w]^T$, and let the wind velocity in the LTP frame expressed in body frame be $\mathbf{V}_b^w = [u_w, v_w, w_w]^T$. Observe that \mathbf{V}_u^a can be expressed in $\{w\}$ frame as $\mathbf{V}_w^a = [V_a \ 0 \ 0]^T$, and let $\mathbf{V}_b^a = [u_a \ v_a \ w_a]^T$ be its components expressed in the body frame. Utilizing the definition of the angles of attack and sideslip relating the wind frame to the body-fixed frame and the “wind triangle” (14.20) expressed in the body frame results in the following:

Fig. 14.8 Wind triangle in 2D plane and the yaw (ψ), sideslip (β), and course (χ) angles



$$\mathbf{V}_b^a = \mathbf{V}_b^g - \mathbf{V}_b^w = \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix}$$

$$\mathbf{V}_b^a = \begin{bmatrix} u_a \\ v_a \\ w_a \end{bmatrix} = R_w^b \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix}.$$

$$\begin{bmatrix} u_a \\ v_a \\ w_a \end{bmatrix} = V_a \begin{bmatrix} \cos \alpha \cos \beta \\ \sin \beta \\ \sin \alpha \cos \beta \end{bmatrix} \quad (14.21)$$

This last equation relates the airspeed components of \mathbf{V}_b^a resolved in the body frame with the airspeed and the angles of attack and sideslip. In turn, if the wind components resolved in the body frame are known, then inverting the last equation allows for calculation of the airspeed V_a and the α , β angles:

$$V_a = \sqrt{u_a^2 + v_a^2 + w_a^2}, \quad \alpha = a \tan \left(\frac{w_a}{u_a} \right), \quad \beta = a \sin \left(\frac{v_a}{V_a} \right). \quad (14.22)$$

14.2.3.5 Summary of Kinematics

This section developed the fundamental kinematic equations that not only define the kinematics of states and contribute to the final set of 6DoF equations of motion but also serve as the basis for the design of the guidance and navigation tasks. There are numerous publications describing the kinematics of moving frames. Most of the publications originate in the area of classical mechanics and rigid body

dynamics (Murray et al. 1994; Goldstein 1980). The publications in the area of flight dynamics and control always contain material addressing the attitude representation techniques and differential rotations and thus can be a good source of reference information. The most recent and thorough presentation of these topics can be found in (Beard and McLain 2012) where the authors specifically address the kinematics and dynamics of small UAVs.

14.3 Rigid Body Dynamics

This section addresses the development of the dynamics of a rigid body. The discussion is based on the application of the Newton's laws in the cases of linear and angular motion. In particular, the Newton's second law of motion states that the sum of all external forces acting on a body in an inertial frame must be equal to the time rate of change of its linear momentum. On the other hand, the sum of the external moments acting on a body must be equal to the time rate of change of its angular momentum. The application of these laws is the objective of this chapter.

Thus, a fixed-wing UAV is considered as the rigid body, and its dynamics is defined with respect to the body-fixed coordinate system. The approach considers a typical fixed-wing UAV operating in a small region of the Earth thus justifying the assumption of LTP frame as an inertial frame. Relations necessary to translate the inertial forces to the body-fixed frame are also presented. Before proceeding to the derivation, it is necessary to present some assumptions typical for the fixed-wing UAVs:

A1. The mass of the UAV remains constant during flight.

A2. The UAV is a rigid body.

The relations derived in this chapter are general and can be applied to any rigid body; however, the treatment of the aerodynamic forces and moments acting on the body will be specific to the aerodynamically controlled fixed-wing UAVs.

14.3.1 Conservation of Linear Momentum

First, assume that a rigid body consists of a set of i “isolated” elementary particles with mass m_i exposed to the external force \mathbf{F}_u^i while being connected together by the internal forces \mathbf{R}_u^i . Since the set of N particles comprises a rigid body structure (see **A2**), the net force exerted by all the particles is $\sum_i^N \mathbf{R}_u^i = 0$. The set of external forces acting on the body is a combination of the gravity force acting in an inertial frame $\{u\}$ and the aerodynamic and propulsion forces defined with respect to the body-fixed frame $\{b\}$ but expressed in the inertial frame $\{u\}$. Thus, the linear momentum of a single particle expressed in an inertial frame obeys the equality

$$\mathbf{F}_u^i + \mathbf{R}_u^i = \frac{d}{dt}(m_i \mathbf{V}_u^i), \quad (14.23)$$

where the time derivative is taken in an inertial frame. Summing up all the N particles comprising the body gives the linear momentum equation of the entire body

$$\sum_{i=1}^N \mathbf{F}_u^i = \sum_{i=1}^N \frac{d}{dt} (m_i \mathbf{V}_u^i). \quad (14.24)$$

The left-hand side of this equation represents the sum of all forces (gravitational, propulsion, and aerodynamic) expressed in an inertial frame, with the right side depending on the velocity of the body defined in an inertial frame. First, observe that the individual inertial velocities \mathbf{V}_u^i are not independent. Since they comprise a rigid body, their vectors can be represented as a sum of \mathbf{V}_b^g the inertial velocity vector resolved in $\{b\}$ frame with respect to CG plus the velocity vector induced by the body rotation with respect to the CG and defined by the radius vector \mathbf{r}_b^i of the i th particle in $\{b\}$ frame; thus $\mathbf{V}_u^i = \mathbf{V}_b^g + \boldsymbol{\omega} \times \mathbf{r}_b^i$. Next, using an assumption **A1** and utilizing the result in (14.14) for the total velocity of the i th particle in an inertial frame, allows for the calculation of the absolute time derivatives in an inertial frame in the following form:

$$\begin{aligned} \sum_{i=1}^N \mathbf{F}_u^i &= \sum_{i=1}^N \frac{d}{dt} (m_i \mathbf{V}_u^i) = \sum_{i=1}^N \frac{d}{dt} (m_i (\mathbf{V}_b^g + \boldsymbol{\omega} \times \mathbf{r}_b^i)) \\ &= \sum_{i=1}^N m_i \frac{d\mathbf{V}_b^g}{dt} + \sum_{i=1}^N m_i \frac{d}{dt} (\boldsymbol{\omega} \times \mathbf{r}_b^i) \\ &= \sum_{i=1}^N m_i \frac{d\mathbf{V}_b^g}{dt} + \frac{d}{dt} \left[\boldsymbol{\omega} \times \sum_{i=1}^N m_i \mathbf{r}_b^i \right]. \end{aligned} \quad (14.25)$$

Here $\boldsymbol{\omega}$ represents the angular velocity of the UAV body resolved in the inertial frame; see (14.14). Defining \mathbf{r}_b^{cg} the vector of CG location in $\{b\}$ frame as $m\mathbf{r}_b^{cg} = \sum_{i=1}^N m_i \mathbf{r}_b^i$, where $m = \sum_{i=1}^N m_i$ is the total mass of the body, simplifies the linear momentum equation:

$$\sum_{i=1}^N \mathbf{F}_u^i = m \frac{d\mathbf{V}_b^g}{dt} + m \frac{d}{dt} [\boldsymbol{\omega} \times \mathbf{r}_b^{cg}].$$

Resolving all external forces in $\{b\}$ frame and assuming that the location of CG does not change with time and applying the result in (14.14) to the absolute derivatives of the vectors \mathbf{V}_b^g and $\boldsymbol{\omega}$ result in

$$\mathbf{F}_b = m(\dot{\mathbf{V}}_b^g + \dot{\boldsymbol{\omega}} \times \mathbf{r}_b^{cg} + \boldsymbol{\omega} \times \mathbf{V}_b^g + \boldsymbol{\omega} \times [\boldsymbol{\omega} \times \mathbf{r}_b^{cg}]), \quad (14.26)$$

where the vector quantities can be represented in the following scalar form:

$\mathbf{F}_b = [X, Y, Z]^T$ – the externally applied forces expressed in the body frame
 $\mathbf{V}_b^g = [u, v, w]^T$ – the inertial velocity components defined in the body frame
 $\boldsymbol{\omega} = [p, q, r]^T$ – the inertial angular velocity resolved in the body frame
 $\mathbf{r}_b^{cg} = [x_{cg}, y_{cg}, z_{cg}]^T$ – the body referenced location of the center of gravity.

The translation of the inertial forces to the body frame is justified by the convenience of calculating the local body frame derivatives of the \mathbf{V}_b^g and $\boldsymbol{\omega}$ expressed in the body frame; the first one results in $\dot{\mathbf{V}}_b^g$, while the derivative of $\boldsymbol{\omega}$ is independent on the coordinate frame; see (14.16).

Utilizing the double vector product identity $\boldsymbol{\omega} \times [\boldsymbol{\omega} \times \mathbf{r}_b^{cg}] = (\boldsymbol{\omega} \cdot \mathbf{r}_b^{cg}) \cdot \boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \boldsymbol{\omega}) \cdot \mathbf{r}_b^{cg}$ allows for the expansion of the linear momentum equation in the scalar form as follows:

$$\begin{aligned}
 X &= m [\dot{u} + qw - rv + \dot{q}z_{cg} - \dot{r}y_{cg} + (qy_{cg} + rz_{cg})p - (q^2 + r^2)x_{cg}] \\
 Y &= m [\dot{v} + ru - pw + \dot{r}x_{cg} - \dot{p}z_{cg} + (rz_{cg} + px_{cg})q - (r^2 + p^2)y_{cg}] \\
 Z &= m [\dot{w} + pv - qu + \dot{p}y_{cg} - \dot{q}x_{cg} + (px_{cg} + qy_{cg})r - (p^2 + q^2)z_{cg}].
 \end{aligned} \tag{14.27}$$

The last set of equations allows for an arbitrary choice of the body frame origin $\{b\}$ with respect to the CG. However, if the origin of the body-fixed frame $\{b\}$ is chosen at the CG, the last set of equations can be significantly simplified by substituting $\mathbf{r}_b^{cg} = [0, 0, 0]^T$, thus leading to

$$\mathbf{F}_b = m (\dot{\mathbf{V}}_b^g + \boldsymbol{\omega} \times \mathbf{V}_b^g). \tag{14.28}$$

Expanding the cross product results in the following simplified form of the linear momentum equation:

$$\begin{aligned}
 X &= m [\dot{u} + qw - rv] \\
 Y &= m [\dot{v} + ru - pw] \\
 Z &= m [\dot{w} + pv - qu].
 \end{aligned} \tag{14.29}$$

Resolving equations (14.29) with respect to the derivatives (accelerations in the body frame) leads to the standard form of differential equations suitable for immediate mathematical modeling:

$$\begin{aligned}
 \dot{u} &= \frac{X}{m} + [rv - qw] \\
 \dot{v} &= \frac{Y}{m} + [pw - ru] \\
 \dot{w} &= \frac{Z}{m} + [qu - pv].
 \end{aligned} \tag{14.30}$$

14.3.2 Conservation of Angular Momentum

Applying the law of conservation of angular momentum to an i th particle in a moving frame is very similar to the approach used above. Consider an i th particle subjected to the internal (\mathbf{M}_u^i) and external ($\mathbf{r}_u^i \times \mathbf{F}_u^i$) moments acting on the body in inertial frame. Similar to the linear momentum case, the sum of internal moments acting on the particle of a rigid body (A2) should be equal to zero $\left(\sum_{i=1}^N \mathbf{M}_u^i = 0\right)$, while the external moments arise from the inertial gravity and the body-attached forces, such as aerodynamic and propulsion. Thus, the conservation of angular momentum calculated across the entire rigid body results in

$$\sum_{i=1}^N (\mathbf{M}_u^i + \mathbf{r}_b^i \times \mathbf{F}_u^i) = \sum_{i=1}^N \mathbf{M}_u^i + \sum_{i=1}^N \mathbf{r}_b^i \times \mathbf{F}_u^i = \sum_{i=1}^N \mathbf{r}_b^i \times \frac{d}{dt}(m_i \mathbf{V}_u^i), \quad (14.31)$$

where the sum of internal moments cancel each other out. Then, applying the Coriolis theorem (14.14) leads to

$$\begin{aligned} \sum_{i=1}^N \mathbf{r}_b^i \times \frac{d}{dt}(m_i \mathbf{V}_u^i) &= \sum_{i=1}^N m_i \mathbf{r}_b^i \times \frac{d}{dt}(\mathbf{V}_u^i) \\ &= \sum_{i=1}^N m_i \mathbf{r}_b^i \times \left(\frac{\delta \mathbf{V}_b^g}{\delta t} + \boldsymbol{\omega} \times \mathbf{V}_b^g + \frac{\delta \boldsymbol{\omega}}{\delta t} \times \mathbf{r}_b^i + \boldsymbol{\omega} \times [\boldsymbol{\omega} \times \mathbf{r}_b^i] \right) \\ &= \sum_{i=1}^N m_i \mathbf{r}_b^i \times \left(\frac{\delta \mathbf{V}_b^g}{\delta t} + \boldsymbol{\omega} \times \mathbf{V}_b^g \right) + \sum_{i=1}^N m_i \mathbf{r}_b^i \times \left[\frac{\delta \boldsymbol{\omega}}{\delta t} \times \mathbf{r}_b^i \right] \\ &\quad + \sum_{i=1}^N m_i \mathbf{r}_b^i \times [\boldsymbol{\omega} \times [\boldsymbol{\omega} \times \mathbf{r}_b^i]]. \end{aligned} \quad (14.32)$$

The first term can be expanded by utilizing the definition of the CG:

$$\begin{aligned} \sum_{i=1}^N m_i \mathbf{r}_b^i \times \left(\frac{\delta \mathbf{V}_b^g}{\delta t} + \boldsymbol{\omega} \times \mathbf{V}_b^g \right) &= m \mathbf{r}_{cg} \times \left(\frac{\delta \mathbf{V}_b^g}{\delta t} + \boldsymbol{\omega} \times \mathbf{V}_b^g \right) \\ &= \begin{bmatrix} m [y_{cg}(\dot{w} + pv - qu) - z_{cg}(\dot{v} + ru - pw)] \\ m [z_{cg}(\dot{u} + qw - rv) - x_{cg}(\dot{w} + pv - qu)] \\ m [x_{cg}(\dot{v} + ru - pw) - y_{cg}(\dot{u} + qw - rv)] \end{bmatrix}. \end{aligned} \quad (14.33)$$

Utilizing the double vector product identity allows for the expansion of the second term as follows:

$$\begin{aligned}
 \sum_{i=1}^N m_i \mathbf{r}_b^i \times \left(\frac{\delta \boldsymbol{\omega}}{\delta t} \times \mathbf{r}_b^i \right) &= \sum_{i=1}^N m_i \left(\frac{\delta \boldsymbol{\omega}}{\delta t} (\mathbf{r}_b^i \cdot \mathbf{r}_b^i) - \mathbf{r}_b^i \left(\frac{\delta \boldsymbol{\omega}}{\delta t} \cdot \mathbf{r}_b^i \right) \right) \\
 &= \begin{bmatrix} \sum_{i=1}^N m_i ((y_i^2 + z_i^2) \dot{p} - (y_i \dot{q} + z_i \dot{r}) x_i) \\ \sum_{i=1}^N m_i ((z_i^2 + x_i^2) \dot{q} - (z_i \dot{r} + x_i \dot{p}) y_i) \\ \sum_{i=1}^N m_i ((x_i^2 + y_i^2) \dot{r} - (x_i \dot{p} + y_i \dot{q}) z_i) \end{bmatrix} = \begin{bmatrix} I_{xx} \dot{p} + I_{xy} \dot{q} + I_{xz} \dot{r} \\ I_{yx} \dot{p} + I_{yy} \dot{q} + I_{yz} \dot{r} \\ I_{zx} \dot{p} + I_{zy} \dot{q} + I_{zz} \dot{r} \end{bmatrix} \\
 &= \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I} \cdot \dot{\boldsymbol{\omega}}. \tag{14.34}
 \end{aligned}$$

The equation (14.34) is obtained by recognizing the moments of inertia and their symmetrical properties:

$$\begin{aligned}
 I_{xx} &= \sum_{i=1}^N m_i (y_i^2 + z_i^2) & I_{yy} &= \sum_{i=1}^N m_i (z_i^2 + x_i^2) & I_{zz} &= \sum_{i=1}^N m_i (x_i^2 + y_i^2) \\
 I_{xy} &= I_{yx} = - \sum_{i=1}^N m_i x_i y_i & I_{xz} &= I_{zx} = - \sum_{i=1}^N m_i x_i z_i & I_{yz} &= I_{zy} = - \sum_{i=1}^N m_i y_i z_i
 \end{aligned}$$

Combining them into a matrix form defines the inertia tensor \mathbf{I} that allows the conversion of the entire double vector product into a very compact form as in (14.34). The diagonal terms of \mathbf{I} are called the moments of inertia. The off-diagonal terms are called the products of inertia; they define the inertia cross coupling. The moments of inertia are directly proportional to the UAV's tendency to resist angular acceleration with respect to a specific axis of rotation. For a body with axes of symmetry, the inertia tensor can be resolved (Goldstein 1980) with zero off-diagonal terms that significantly simplify its form and the final equations of angular momentum.

The last term in (14.32) utilizes twice the same double cross product expansion, thus leading to

$$\begin{aligned} \sum_{i=1}^N m_i \mathbf{r}_b^i \times [\boldsymbol{\omega} \times [\boldsymbol{\omega} \times \mathbf{r}_b^i]] &= \sum_{i=1}^N m_i \mathbf{r}_b^i \times ((\boldsymbol{\omega} \cdot \mathbf{r}_b^i) \boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \boldsymbol{\omega}) \mathbf{r}_b^i) \\ &= \begin{bmatrix} I_{yz}(q^2 - r^2) + I_{xz}pq - I_{xy}pr \\ I_{xz}(r^2 - p^2) + I_{xy}rq - I_{yz}pq \\ I_{xy}(p^2 - q^2) + I_{yz}pr - I_{xz}qr \end{bmatrix} + \begin{bmatrix} (I_{zz} - I_{yy})rq \\ (I_{xx} - I_{zz})rp \\ (I_{yy} - I_{xx})qp \end{bmatrix}. \end{aligned} \quad (14.35)$$

Resolving the total inertial moment acting on the UAV in body frame and denoting the components as $\mathbf{M}_b = [L, M, N]^T$ and combining the results in (14.33)–(14.35) lead to the following complete angular momentum equations in an expanded form:

$$\begin{aligned} L &= I_{xx}\dot{p} + I_{xy}\dot{q} + I_{xz}\dot{r} \\ &\quad + I_{yz}(q^2 - r^2) + I_{xz}pq - I_{xy}pr + (I_{zz} - I_{yy})rq \\ &\quad + m[y_{cg}(\dot{w} + pv - qu) - z_{cg}(\dot{v} + ru - pw)] \\ M &= I_{yx}\dot{p} + I_{yy}\dot{q} + I_{yz}\dot{r} \\ &\quad + I_{xz}(r^2 - p^2) + I_{xy}rq - I_{yz}pq + (I_{xx} - I_{zz})rp \\ &\quad + m[z_{cg}(\dot{u} + qw - rv) - x_{cg}(\dot{w} + pv - qu)] \\ N &= I_{zx}\dot{p} + I_{zy}\dot{q} + I_{zz}\dot{r} \\ &\quad + I_{xy}(p^2 - q^2) + I_{yz}pr - I_{xz}qr + (I_{yy} - I_{xx})qp \\ &\quad + m[x_{cg}(\dot{v} + ru - pw) - y_{cg}(\dot{u} + qw - rv)]. \end{aligned} \quad (14.36)$$

If the origin of the body-fixed frame $\{b\}$ is chosen at the CG ($\mathbf{r}_b^{cg} = [0, 0, 0]^T$), then in the case of a typical UAV with a vertical plane of symmetry spanned by the body-fixed axes x_b, z_b , the two pairs of the off-diagonal terms of \mathbf{I} matrix become zero, namely, $I_{xy} = I_{yx} = 0$ and $I_{yz} = I_{zy} = 0$. This significantly simplifies the above equations:

$$\begin{aligned} L &= I_{xx}\dot{p} + (I_{zz} - I_{yy})rq + I_{xz}(\dot{r} + pq) \\ M &= I_{yy}\dot{q} + (I_{xx} - I_{zz})rp + I_{xz}(r^2 - p^2) \\ N &= I_{zz}\dot{r} + (I_{yy} - I_{xx})qp + I_{zx}(\dot{p} - qr). \end{aligned} \quad (14.37)$$

These equations represent the complete rotational dynamics of a typical fixed-wing UAV modeled as a rigid body with a longitudinal plane of symmetry.

14.3.3 Complete Set of 6DoF Equations of Motion

The final set of 6DoF equations of motion describing the kinematics and dynamics of a generic UAV with a longitudinal plane of symmetry modeled as a rigid body can be summarized as follows:

$$\begin{aligned} X &= m [\dot{u} + qw - rv] \\ Y &= m [\dot{v} + ru - pw] \\ Z &= m [\dot{w} + pv - qu] \end{aligned} \quad (14.38)$$

$$\begin{aligned} L &= I_{xx} \dot{p} + (I_{zz} - I_{yy})rq + I_{xz}(\dot{r} + pq) \\ M &= I_{yy} \dot{q} + (I_{xx} - I_{zz})rp + I_{xz}(r^2 - p^2) \\ N &= I_{zz} \dot{r} + (I_{yy} - I_{xx})qp + I_{xz}(\dot{p} - qr) \end{aligned} \quad (14.39)$$

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{R}^u \mathbf{V}_b^g = \\ &\begin{bmatrix} \cos \theta \cos \psi & -\cos \theta \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \mathbf{V}_b^g \end{aligned} \quad (14.40)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \frac{\sin \theta}{\cos \theta} & \cos \phi \frac{\sin \theta}{\cos \theta} \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \frac{1}{\cos \theta} & \cos \phi \frac{1}{\cos \theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (14.41)$$

Analysis of the above differential equations shows that these equations are nonlinear and coupled, that is, each differential equation depends upon variables which are described by other differential equations. In general case, their analytical solutions are not known, and they can only be solved numerically. There are 12 states describing the free motion of a rigid body subject to external forces ($\mathbf{F}_b = [X, Y, Z]^T$) and moments ($\mathbf{M}_b = [L, M, N]^T$). In the control system design, these variables are called state variables because they completely define the state of a rigid body at any instance of time. The state variables are summarized in Table 14.1 for completeness.

What remains in the description of 6DoF equations of motion is to define the external forces and moments acting on the airplane. This will be the objective of the next section.

Table 14.1 State variables of the 6DoF equations of motion

| State variable | Definition |
|-------------------------------------|--|
| $\mathbf{r} = [r_x, r_y, r_z]^T$ | Vector of inertial position of the UAV and its components |
| $\mathbf{V}_b^g = [u, v, w]^T$ | Vector of inertial velocity components resolved in the body-fixed frame |
| $[\phi, \theta, \psi]$ | Euler angles that define the attitude of body-fixed frame with respect to the inertial frame |
| $\boldsymbol{\omega} = [p, q, r]^T$ | The inertial angular rates resolved in the body-fixed frame |

14.4 Forces and Moments Acting on the Airplane

The objective of this section is to present a generalized approach to defining the external forces and moments acting on a fixed-wing UAV as functions of its states. The primary forces and moments acting on an airplane are the gravitational, thrust of the propulsion system, aerodynamic, and disturbances due to the flight in unsteady atmosphere. The most challenging task here is in defining the aerodynamic forces and moments resulting from the air–body interaction. Although the aerodynamic description of airfoils defining a fixed wing is not a new subject, the varieties of shapes, aspect ratios, and aerodynamic configurations of modern fixed-wing UAVs do not allow thorough presentation of all configurations. As an example, possible aerodynamic configurations of aerodynamic surfaces include tandem, variable span wings, joined wings, twin boom, and V-tail configuration, just to name a few. However, a generalization is possible. An interested reader is referred to the most relevant survey (Mueller and DeLaurier 2003) of aerodynamics of small UAV that describes the modeling approaches and their limitations.

14.4.1 Gravitation

Assuming that the flight altitude is negligible in comparison to the radius of the Earth, it is sufficient to consider the gravity’s magnitude constant. Then, the effect of the Earth’s gravitation can be naturally modeled in the body-carried frame by the force applied to the CG of the UAV of mass m ; the gravitational force is proportional to the gravitational constant g and is called the weight of the UAV

$$\mathbf{F}_u^{gr} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}. \quad (14.42)$$

Before substituting this force expression into the equations of motion (14.39), it needs to be resolved in the body frame. The inertial to body rotation R_u^b enables this transformation:

$$\mathbf{F}_b^{gr} = R_u^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = mg \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \theta \cos \phi \end{bmatrix}. \quad (14.43)$$

Assuming that the frame $\{b\}$ origin is chosen at the CG and since the gravitational force acts through the CG of the airplane, the corresponding moment contribution is zero, $\mathbf{M}_b^{gr} = [0, 0, 0]^T$.

14.4.2 Propulsion

The configuration of the propulsion system of modern fixed-wing UAVs varies greatly. The architectures can be categorized by the number of engines, their type, and their installation arrangement in the airframe. A thorough review of the existing configurations along with some future projections and trends in the modern and future UAV systems can be found in reference (OSD 2001). However, what is common across all possible configurations is that the vector of thrust in all systems is set parallel to the existing axes of symmetry; the thrust vectoring is not a common feature of fixed-wing UAVs yet.

The thrust is naturally represented in the body-fixed reference system. The direction of the thrust vector \mathbf{F}_b^{tr} is usually fixed and lies in the plane of symmetry or is parallel to it; however, it may not be aligned with the longitudinal x_b -axis. If the orientation of the thrust vector \mathbf{F}_b^{tr} varies in its reference to the airframe, then a separate coordinate system analogous to the wind axes should be defined, thus introducing the required rotation of the thrust vector to the body-fixed coordinate system. It is a common design requirement that the installation of multiple engines should not introduce any unbalanced moments, thus not inducing any loss of control efforts for the UAV stabilization. For the analysis of a nominal flight regime, the thrust vector \mathbf{F}_b^{tr} is considered fixed with respect to the body-fixed frame.

For the sake of simplicity, consider a typical fixed-wing UAV architecture where the installation of one or multiple engines results in the cumulative thrust vector \mathbf{F}_b^{tr} passing through the CG and the only moment being the torque generated primarily by the reactive force from the rotating propeller; depending on the type and the power of the propulsion system, there might be three more components (Illman 1999) of the torque, namely, the spiraling slipstream, the gyroscopic precession, and the asymmetric propeller loading (“P-factor”). Thus, in the case of a typical UAV, the net force X_{tr} of thrust in x_b direction and the moment L around x_b axis can be considered proportional to the thrust control command δ_{tr} . Moreover, thrust characteristics of most conventional engines are always functions of the air density and the airspeed. Thus, the contributing force and moment resulting from the propulsion system can be presented as follows:

$$\mathbf{F}_b^{tr} = \begin{bmatrix} F_{tr}(V_a, h, \delta_{tr}) \\ 0 \\ 0 \end{bmatrix}; \mathbf{M}_b^{tr} = \begin{bmatrix} M_{tr}(V_a, h, \delta_{tr}) \\ 0 \\ 0 \end{bmatrix}. \quad (14.44)$$

A particular example of modeling the propulsion force for the case of a micro UAV can be found in Beard and McLain (2012).

14.4.3 Unsteady Atmosphere

In the previous discussion of the wind frame, it was assumed that the wind \mathbf{V}_u^w defined in the LTP frame is constant; thus, the velocities are related by the “wind triangle” equation $\mathbf{V}_u^a = \mathbf{V}_u^g - \mathbf{V}_u^w$.

The most common approach (McRuer et al. 1999) in wind modeling is to consider two components contributing to the wind. The first component $\mathbf{V}_u^{wsteady}$ defines the steady wind resolved in the inertial frame, and therefore it can be presented by the measurements in the LTP frame. The second component \mathbf{V}_b^{wgust} is stochastic, which represents the short-period disturbances or gusts resolved in the body-fixed frame. Since the equations of motion are written in the body-fixed frame, then

$$\mathbf{V}_b^w = R_u^b \mathbf{V}_u^{wsteady} + \mathbf{V}_b^{wgust}. \quad (14.45)$$

From the components of the wind and the UAV velocity, both resolved in the body frame, it is therefore possible to find the body frame components of the air velocity as

$$\mathbf{V}_b^a = \begin{bmatrix} u_a \\ v_a \\ w_a \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} - R_u^b \begin{bmatrix} u^{wsteady} \\ v^{wsteady} \\ w^{wsteady} \end{bmatrix} - \begin{bmatrix} u^{wgust} \\ v^{wgust} \\ w^{wgust} \end{bmatrix}. \quad (14.46)$$

These body frame components of the air velocity enable straightforward calculation of the airspeed and the angles of attack and sideslip as in (14.22).

Modeling of the stochastic and steady components of wind is based primarily on a history of experimental observations expressed using linear filters. The most widely used techniques are represented by von Karman and Dryden wind turbulence models (Hoblit 2001). Both methods are well supported with their numerical implementations.

14.4.4 Aerodynamics

Aerodynamic forces and moments depend on the interaction of an aircraft with the airflow, which may also be in motion relative to the Earth. However, for the purpose of representing the nominal aerodynamic effects, the large-scale motion of the atmosphere is not critical and therefore will be considered constant; in fact, it will only affect the navigation of the UAV.

The small perturbation theory (Ashley and Landahl 1985) is one of the approaches used in describing the aerodynamic interaction of a given aerodynamic shape with airflow. The perturbation in aerodynamic forces and moments are

functions of variations in state variables and control inputs. The control inputs here are the deflections of the control surfaces of an airplane that modify the airflow around the body, thus generating the desired aerodynamic effects. The nomenclature of the control surfaces and their control mechanization depends on the particular aerodynamic composition of the airplane. Nevertheless, the principles describing the effects of the control surface deflection on the generated forces and moments are the same. Consider the following control effectors of a classical aerodynamic configuration: the elevator, the aileron, and the rudder (see Fig. 14.9). In this configuration, the ailerons are used to control the roll angle ϕ , the elevator is used to control the pitch angle θ , and the rudder controls the yaw angle ψ .

Their deflections are denoted as δ_a for the aileron, δ_e for the elevator, and δ_r for the rudder. The positive deflection of a control surface is defined by applying the right-hand rule to the rotation axis of the surface. The positive direction of the aileron, elevator, and rudder deflections is also depicted in Fig. 14.9.

Deflection of the control surfaces modifies the pressure distribution around the control surfaces or the entire body thus producing corresponding forces. The forces acting with respect to the CG of the body result in aerodynamic moments. For example, deflecting the elevator primarily changes the pitching moment acting on the airplane. In turn, this results in changing the angle of attack of the wing that increases the lifting power of the airplane. The calculation of aerodynamic characteristics of one or more lifting surfaces with variable deflections of the control surfaces at various attitudes with respect to the airflow can be accomplished by utilizing well-developed linear panel methods (Hess 1990; Henne 1990) conveniently implemented in various software packages (Fearn 2008; Kroo 2012).

The panel methods capture the effect of pressure distribution in the form of parameterized forces and moments versus the angles of attack and sideslip, and airspeed; they play a role of states here. For example, considering the longitudinal plane, the effect of aerodynamic pressure acting on a fixed wing can be modeled using a total force \mathbf{F}_b^Σ and pitching moment \mathbf{M}_b^Σ acting on the wing both resolved in the body frame. It is common to project the total force to the wind axes, thus resulting in the lift F_{lift} and drag F_{drag} force components. Figure 14.10

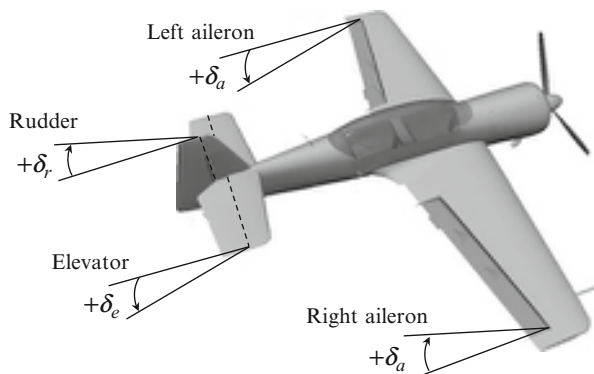
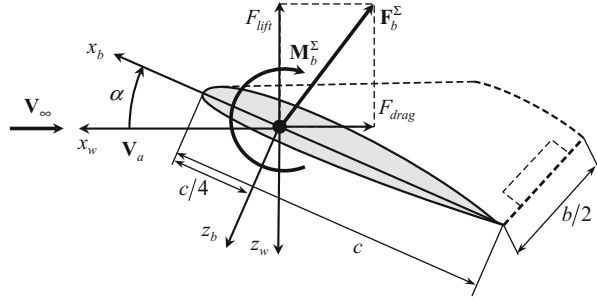


Fig. 14.9 Control surfaces of a classical aerodynamic configuration

Fig. 14.10 Definition of the lift and drag forces and the pitching moment in the wind frame



demonstrates the approach to modeling aerodynamic effects in the wind and body-fixed frames with respect to the vector of free airstream \mathbf{V}_∞ .

As shown in Fig. 14.10, the lift F_{lift} and drag F_{drag} forces act in the wind frame and are applied at the aerodynamic center of the lifting surface that is located at the quarter-chord point (c is the length of the mean aerodynamic chord). The pitching component M of moment \mathbf{M}_b^Σ acts around the aerodynamic center. Then, the values of forces and moments are represented in a form connecting a number of surface-specific parameters and the states in the following form:

$$\begin{aligned} F_{lift} &= 1/2 \rho V_a^2 S C_L \\ F_{drag} &= 1/2 \rho V_a^2 S C_D, \\ M &= 1/2 \rho V_a^2 S c C_m \end{aligned} \quad (14.47)$$

where C_L, C_D, C_m are the nondimensional aerodynamic coefficients (to be parameterized), S is the planform area of the wing surface, and c and b are the mean aerodynamic chord and the wing span. The same approach is applied to each of the aerodynamic surfaces comprising the airplane. Then by using the parallel axis theorem (Goldstein 1980), the elementary moments from all the lifting and control surfaces can be transferred to the CG of the rigid body, thus resolving the actions with respect to one unifying center.

It is common practice to consider the total aerodynamic forces and moments in projections to the longitudinal and lateral planes of the airplane. The benefit of this approach is in the simplicity of representing the aerodynamic effects and in providing a natural ground for the nonlinear model decomposition at the next step of the control system design. Thus, the longitudinal forces and moments (14.47) consist of lift, drag, and pitching moment acting in the vertical plane of symmetry. The lateral side force F_{side} , yawing N , and rolling L moments are caused by the asymmetric airflow around the airplane and control surfaces deflection; the asymmetry can be caused by the side wind or intentional deflection of the rudder. For the majority of fixed-wing UAVs, the key states that define the parameterization

Table 14.2

Parameterization of longitudinal and lateral aerodynamics

| Longitudinal channel | Lateral channel |
|---|---|
| $F_{drag} = 1/2\rho V_a^2 S C_D(\alpha, q, \delta_e)$ | $F_{side} = 1/2\rho V_a^2 S C_Y(\beta, p, r, \delta_r, \delta_a)$ |
| $F_{lift} = 1/2\rho V_a^2 S C_L(\alpha, q, \delta_e)$ | $L = 1/2\rho V_a^2 S b C_l(\beta, p, r, \delta_r, \delta_a)$ |
| $M = 1/2\rho V_a^2 S c C_m(\alpha, q, \delta_e)$ | $N = 1/2\rho V_a^2 S b C_n(\beta, p, r, \delta_r, \delta_a)$ |

of the aerodynamic coefficient are the angle of attack α , the sideslip β , body rates $[p, q, r]$, and the controls which are the surface deflections $[\delta_e, \delta_r, \delta_a]$. The most general functional form of the longitudinal and lateral aerodynamics can be presented as follows (Table 14.2):

Without delving deeper into the intricacies of aerodynamic parameterization, it is sufficient to demonstrate the final form of forces and moments defined in the wind coordinate frame:

- Longitudinal plane

$$\begin{aligned}
 F_{drag} &= 1/2\rho V_a^2 S \left(C_{D_0} + C_D^\alpha \alpha + C_D^q \frac{c}{2V_a} \cdot q + C_D^{\delta_e} \cdot \delta_e \right) \\
 F_{lift} &= 1/2\rho V_a^2 S \left(C_{L_0} + C_L^\alpha \alpha + C_L^q \frac{c}{2V_a} \cdot q + C_L^{\delta_e} \cdot \delta_e \right) \\
 M &= 1/2\rho V_a^2 S c \left(C_{M_0} + C_M^\alpha \alpha + C_M^q \frac{c}{2V_a} \cdot q + C_M^{\delta_e} \cdot \delta_e \right)
 \end{aligned} \tag{14.48}$$

- Lateral plane

$$\begin{aligned}
 F_{side} &= 1/2\rho V_a^2 S \left(C_{Y_0} + C_Y^\beta \beta + C_Y^p \frac{b}{2V_a} \cdot p + C_Y^r \frac{b}{2V_a} \cdot r + C_Y^{\delta_r} \cdot \delta_r + C_Y^{\delta_a} \cdot \delta_a \right) \\
 L &= 1/2\rho V_a^2 S b \left(C_{l_0} + C_l^\beta \beta + C_l^p \frac{b}{2V_a} \cdot p + C_l^r \frac{b}{2V_a} \cdot r + C_l^{\delta_r} \cdot \delta_r + C_l^{\delta_a} \cdot \delta_a \right) \\
 N &= 1/2\rho V_a^2 S b \left(C_{n_0} + C_n^\beta \beta + C_n^p \frac{b}{2V_a} \cdot p + C_n^r \frac{b}{2V_a} \cdot r + C_n^{\delta_r} \cdot \delta_r + C_n^{\delta_a} \cdot \delta_a \right)
 \end{aligned} \tag{14.49}$$

The presented parameterization is a simple linear approximation of the aerodynamics given by the Taylor series expansion taken with respect to the given trim conditions. The coefficients $C_{f/m}^{state}$ are the nondimensional partial derivatives of the corresponding forces and moments (denoted in the subscript) defined with respect to the corresponding state or control (denoted in the superscript). The coefficients with zero in the subscript denote the forces and moments calculated when all states, including the control surface deflection, are zero; for example, C_{l_0} denotes the roll moment coefficient estimated at $\beta = p = r = 0$ and $\delta_r = \delta_a = 0$. The common naming convention suggests that those derivatives defined with respect to states $[\alpha, \beta, p, q, r]$ are called the stability derivatives and those with respect to controls $[\delta_e, \delta_r, \delta_a]$ are called the control derivatives. The static stability of an aircraft

with respect to disturbances in some variable is directly reflected in the sign of a particular derivative. For example, the sign of C_M^α should be negative to guarantee static stability in pitching motion, while the sign of C_N^β should be positive for the directional static stability.

Each of the presented coefficients is usually a function of states. The precision requirement of the linear parameterization greatly depends on the operational envelop of the UAV and its intended use; the higher the maneuverability of the UAV, the more terms necessary to accurately represent the aerodynamics. Each of the coefficients has very intuitive physical meaning and is usually studied separately. An interested reader is referred to Beard and McLain (2012) for a detailed discussion of the aerodynamic coefficients of small and micro fixed-wing UAVs.

One last step needs to be performed before the aerodynamics (14.48) and (14.49) defined in the wind coordinates can be plugged into the equations of motion (14.38) and (14.39) resolved in the body-fixed frame. The transformation (14.19) from the wind to the body frame serves this purpose.

Therefore, the total forces and moments acting on the fixed-wing UAV can be presented as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_u^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} F_{tr}(V_a, h, \delta_{tr}) \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2} \rho V_a^2 S \cdot R_w^b \begin{bmatrix} C_D(\alpha, q, \delta_e) \\ C_Y(\beta, p, r, \delta_r, \delta_a) \\ C_L(\alpha, q, \delta_e) \end{bmatrix} \quad (14.50)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} M_{tr}(V_a, h, \delta_{tr}) \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2} \rho V_a^2 S \cdot \begin{bmatrix} bC_l(\beta, p, r, \delta_r, \delta_a) \\ cC_M(\alpha, q, \delta_e) \\ bC_n(\beta, p, r, \delta_r, \delta_a) \end{bmatrix}. \quad (14.51)$$

14.5 Accounting for the Earth Rotation Rate

The complete set of 6DoF equations of motion presented above is an approximation of the rigid body kinematics and dynamics and is valid as long as the assumption of the flat Earth model satisfies the task at hand. During the high-speed flight or in long-duration and extended range missions, the precision of the derived states will suffer from omitting the sidereal rate of the rotating Earth. The key reason for the error is in the accumulation over time of the Coriolis and centripetal accelerations induced by the rotating Earth. Thus, the following derivation outlines how the Earth rotation can be accounted for in the definition of the inertial velocity and acceleration vectors.

First, consider the ECI as the true inertial frame $\{i\}$. Next, by using the simplifying properties of defining the free motion of a rigid body with respect to the CG and utilizing the Coriolis theorem, resolve the absolute time derivative of the CG position vector \mathbf{r}_i^{cg} in the true inertial frame as follows:

$$\mathbf{V}_i^{cg} = \dot{\mathbf{r}}_i^{cg} = \mathbf{V}_e^{cg} + \boldsymbol{\Omega}_i^e \times \mathbf{r}_i^{cg}. \quad (14.52)$$

Taking the second time derivative and assuming that the sidereal rate of the Earth rotation is constant ($\dot{\boldsymbol{\Omega}}_i^e = 0$) results in

$$\begin{aligned}\dot{\mathbf{V}}_i^{cg} &= \dot{\mathbf{V}}_b^{cg} + \boldsymbol{\omega} \times \mathbf{V}_e^{cg} + \boldsymbol{\Omega}_i^e \times \dot{\mathbf{r}}_i^{cg} = \dot{\mathbf{V}}_b^{cg} + \boldsymbol{\omega} \times \mathbf{V}_e^{cg} + \boldsymbol{\Omega}_i^e \times (\mathbf{V}_e^{cg} + \boldsymbol{\Omega}_i^e \times \mathbf{r}_i^{cg}) \\ &= \dot{\mathbf{V}}_b^{cg} + (\boldsymbol{\omega} + \boldsymbol{\Omega}_i^e) \times \mathbf{V}_e^{cg} + \boldsymbol{\Omega}_i^e \times (\boldsymbol{\Omega}_i^e \times \mathbf{r}_i^{cg})\end{aligned}\quad (14.53)$$

In (14.53), the $\boldsymbol{\omega}$ denotes the vector of inertial angular velocity resolved in the body frame, and \mathbf{V}_b^{cg} is the same as $\mathbf{V}_b^g = \mathbf{V}_b^g$. The equation (14.52) updates the kinematic dead reckoning equation in (14.39), while the vector of inertial acceleration in (14.53) should be used in the application of the second Newtonian law.

Applying the angular velocities addition theorem, the vector $\boldsymbol{\omega}$ can be represented as a sum of the angular velocity vector $\boldsymbol{\omega}_n^b$ of body frame $\{b\}$ resolved in the body-carried frame $\{n\}$, the angular velocity vector $\boldsymbol{\omega}_e^n$ of body-carried frame $\{n\}$ resolved in ECEF frame $\{e\}$, and the sidereal rate of the Earth rotation vector $\boldsymbol{\Omega}_i^e$ resolved in the true inertial frame $\{i\}$. Thus, the last equation can be also written as

$$\dot{\mathbf{V}}_i^{cg} = \dot{\mathbf{V}}_b^{cg} + (\boldsymbol{\omega}_n^b + \boldsymbol{\omega}_e^n + 2\boldsymbol{\Omega}_i^e) \times \mathbf{V}_e^{cg} + \boldsymbol{\Omega}_i^e \times (\boldsymbol{\Omega}_i^e \times \mathbf{r}_i^{cg}). \quad (14.54)$$

What remains is to define the elements of (14.54) that enable calculation of the vector cross products.

As before, the term $2\boldsymbol{\Omega}_i^e \times \mathbf{V}_e^{cg}$ is the Coriolis, and the term $\boldsymbol{\Omega}_i^e \times (\boldsymbol{\Omega}_i^e \times \mathbf{r}_i^{cg})$, is the centripetal accelerations. The angular velocity vector $\boldsymbol{\omega}_e^n$, can be obtained from the geodetic latitude ($\dot{\phi}$) and longitude ($\dot{\lambda}$) rates, which in turn can be calculated from the NED components of $\mathbf{V}_e^{cg} = [V_N, V_E, V_D]^T$. The transformation of rates of the geodetic system ($\dot{\phi}, \dot{\lambda}$) to the body-carried stabilized frame $\{n\}$ can be obtained similarly to (14.10) by a left-handed rotation around the east axis through the latitude angle ϕ

$$\boldsymbol{\omega}_e^n = \begin{bmatrix} 0 \\ -\dot{\phi} \\ 0 \end{bmatrix} + \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\lambda} \\ 0 \\ 0 \end{bmatrix}. \quad (14.55)$$

The rate of change of latitude and longitude can be calculated from the V_n northern and V_e eastern components of the velocity vector as follows:

$$\dot{\phi} = \frac{V_n}{r_m + h}, \quad \dot{\lambda} = \frac{V_E}{(r_n + h) \cos \varphi}, \quad (14.56)$$

where h is the height of CG above the reference oblate spheroid and

$$r_m = \frac{r_e(1 - \varepsilon^2)}{(1 - \varepsilon^2 \sin^2 \varphi)^{\frac{3}{2}}}, \quad r_n = \frac{r_e}{\sqrt{1 - \varepsilon^2 \sin^2 \varphi}}$$

are the estimates of the reference spheroid radius in the meridian and normal directions at given latitude and longitude. Substituting (14.56) into the (14.55) results in the estimate of ω_e^n as follows:

$$\omega_e^n = \left[\frac{V_E}{r_n + h}, -\frac{V_N}{r_m + h}, -\frac{V_E}{r_n + h} \tan \phi \right]^T. \quad (14.57)$$

Observe that the Earth sidereal rotation vector Ω_i^e has only one component in ECEF frame $\Omega_i^e = [0, 0, \Omega_i^e]^T$. Resolving for convenience Ω_i^e in the $\{n\}$ frame by a single φ rotation produces

$$\Omega_i^n = [\Omega_i^e \cos \phi, 0, -\Omega_i^e \sin \phi]^T, \quad (14.58)$$

thus completing the definition of all terms in (14.54). Obviously, the result of substituting of all the vectors into (14.54) is cumbersome; however it demonstrates how the Earth sidereal rate can be accounted for.

The corresponding linear and angular momentum equations can be obtained by applying the second Newtonian law; the procedure is similar to the simplified case presented above and resulted in the (14.26) and (14.32). Utilizing the same set of assumptions (A1–A2) and resolving all the external forces and moments with respect to the CG in the body frame results in the same angular momentum equation; however, the kinematic and the linear momentum equations need to be modified. Applying the second Newtonian law to the linear motion of the CG and accounting for a new result in (14.52) and (14.53) gives

$$\mathbf{V}_i^{cg} = R_b^e \mathbf{V}_b^{cg} + \Omega_i^e \times \mathbf{r}_i^{cg} \quad (14.59)$$

$$\mathbf{F}_b = m [\dot{\mathbf{V}}_b^{cg} + (\omega + \Omega_b^e) \times R_b^e \mathbf{V}_b^{cg} + \Omega_b^e \times (\Omega_b^e \times R_b^e \mathbf{r}_i^{cg})], \quad (14.60)$$

where \mathbf{F}_b , as before, is the sum of all externally applied forces applied at CG resolved in the body frame. Equations (14.59) and (14.60) are the new relations derived in a true inertial frame $\{i\}$, thus accounting for the rotating Earth.

To give a reader a sense of numerical significance of the resulting acceleration, the following numerical example compares the contribution of the Coriolis and the centripetal terms with an assumption that a UAV is at the constant altitude in the wing-level flight due east and is not maneuvering; therefore, $\omega_n^b = 0$ and $\mathbf{V}_e^{cg} = [0, V_E, 0]^T$. In these conditions, the centripetal term becomes equal to the Coriolis term at the speed of 914 m/s. In turn, when at the equator latitude, the third vertical component of the Coriolis acceleration is about 0.27 m/s^2 , that is, 2.7 % of the acceleration due to gravity (9.8 m/s^2). Thus, the applicability of the simplifying flat Earth assumption becomes justified for a case of a short-duration and relatively low-speed flight of an airplane. Therefore, this new set of equations should be used when accurate modeling is required for a UAV moving faster than 600 m/s over the Earth or when long distance and duration navigation is considered.

14.6 Conclusion

The objective of this chapter was to provide a review of the theoretical material required to enable accurate mathematical representation of the free and controlled motion of a generic fixed-wing UAV modeled as a rigid body. The key building blocks presented were the coordinate frames and their transformations, kinematics of rotation, dynamics of motion, and the definition of forces and moments acting on the airplane. The kinematics of spatial rotation is what connects the three building blocks of the “kinematics–dynamics–actions” triad. In addition to the 6DoF equations of motion describing the kinematics and dynamics of a rigid body motion, the tools and methods developed in this chapter contribute significantly into the UAV flight dynamics, system identification, control, guidance, and navigation.

References

- H. Ashley, M. Landahl, *Aerodynamics of Wings and Bodies (Dover Books on Aeronautical Engineering)* (Dover Publications, New York, 1985)
- R. Beard, T. McLain, *Small Unmanned Aircraft: Theory and Practice* (Princeton University Press, Princeton, 2012)
- B. Etkin, L.D. Reid, *Dynamics of Flight: Stability and Control*, 3rd edn. (Wiley, New York, 1995)
- R.L. Fearn, Airfoil aerodynamics using panel methods. *Math. J. (Wolfram Research)* **10**(4), 15 (2008)
- H. Goldstein, *Classical Mechanics*, 2nd edn. (Addison-Wesley, Reading, 1980)
- P. Henne, *Applied Computational Aerodynamics (Progress in Astronautics and Aeronautics)* (AIAA, Washington, DC, 1990)
- J.L. Hess, Panel methods in computational fluid dynamics. *Ann. Rev. Fluid Mech.* **22**, 255–274 (1990)
- F. Hoblit, *Gust Loads on Aircraft: Concepts and Applications* (AIAA Education Series, Washington, DC, 2001)
- P.E. Illman, *The Pilot's Handbook of Aeronautical Knowledge* (McGraw-Hill Professional, New York, 1999)
- G.H. Kaplan, *The IAU Resolutions on Astronomical Constants, Time Scales, and the Fundamental Reference Frames*, vol. circular no. 163 (United States Naval Observatory, Washington, DC, 1981)
- I. Kroo, LinAir 4. A nonplanar, multiple lifting surface aerodynamics program. Desktop Aeronautics (2012). <http://www.desktop.aero/linair.php>. Accessed 10 Apr 2012
- D. McRuer, I. Ashkenas, D. Graham, *Aircraft Dynamics and Automatic Control* (Princeton University Press, Princeton, 1999)
- T.J. Mueller, J.D. DeLaurier, Aerodynamics of small vehicles. *Ann. Rev. Fluid Mech* **35**, 89–111 (2003)
- R. Murray, Z. Li, S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, vol. 1 (CRC Press, Boca Raton, 1994)
- OSD, *Unmanned Aerial Vehicles Roadmap 2000–2025* (Office of the Secretary of Defence, Washington, DC, 2001)
- R.M. Rogers, *Applied Mathematics in Integrated navigation Systems*, 2nd edn. (AIAA, Reston, 2003)
- M.D. Shuster, A survey of attitude representations. *J. Astronaut. Sci.* **41**(4), 439–517 (1993)

Vladislav Gavrilets

Contents

15.1 Introduction 280

15.2 Helicopter Parameters 282

15.3 Equations of Motion 284

15.4 Component Forces and Moments 285

 15.4.1 Main Rotor Forces and Moments 285

 15.4.2 Engine, Governor, and Rotor Speed Model 295

 15.4.3 Fuselage Forces 298

 15.4.4 Vertical Fin Forces and Moments 299

 15.4.5 Horizontal Stabilizer Forces and Moments 300

 15.4.6 Tail Rotor 300

15.5 Actuator Models 304

References 305

Abstract

A nonlinear dynamic model of a miniature aerobatic helicopter is presented in detail. A component buildup was applied in devising the model, using simplified analytical expressions for the component forces and moments. Flight-test experiments were used to estimate several key parameters, such as the equivalent torsional stiffness in the hub. The model was used to design control logic for aerobatic maneuvers performed entirely under computer control.

V. Gavrilets
Flight Control and Navigation Group, Rockwell Collins Control Technologies, Warrenton,
VA, USA
e-mail: vgavrile@rockwellcollins.com

15.1 Introduction

Miniature helicopters with hingeless rotors are extremely agile compared to their full-scale counterparts due to fast shrinking of moments of inertia with size and therefore growth of control authority Mettler (2002). These vehicles can be used for tasks requiring aggressive maneuvering, such as pursuit and evasion in urban or mountainous environment, or aerial stunts for movie industry. Precise autonomous execution of aggressive maneuvers, coupled with online motion planning, can further expand utility of agile miniature rotorcraft. This created the need for an adequate nonlinear model of an aerobatic miniature helicopter.

An extensive body of literature has been devoted to the dynamics of full-scale helicopters, and step-by-step procedures for creating a first-principles dynamic model were outlined (Padfield 1996; Bramwell 2001; Talbot et al. 1982). Frequency-domain system identification techniques are widely used to evaluate accuracy of linearized models at different operating conditions (NASA Ames Research Center 2000; Tischler 1996). Due to a wide range of flow conditions and their complex interaction with the wake, cross-axis coupling in the hub, and sometimes interaction of the structural modes with the rigid body modes, the models of full-scale helicopters used in simulators tend to be complex and contain a large number of states.

Recent years have seen an increase in modeling efforts for small-scale helicopters. Mettler (2002) performed a comprehensive study of the features common to the dynamics of small-scale helicopters and applied frequency-domain methods for identification of linearized models. Most of the small-scale helicopters have Bell-Hiller stabilizer bars, which slow down rotor dynamics and mitigate gust response. It was shown (Mettler 2002) that this creates relatively low-frequency lateral and longitudinal pendulum-like flapping modes, which involve fuselage, rotor, and stabilizer bar. The same effect was observed on a full-size UH-1H helicopter, equipped with a stabilizer bar with mechanical dampers instead of aerodynamic surfaces. The modes are lightly damped and must be accounted for to develop a high-bandwidth control system required for autonomous aggressive maneuvering.

LaCivita et al. (2002b) developed a technique for integrating first-principles modeling and frequency-domain system identification to yield a nonlinear model for small-scale helicopters. The resulting model of the Yamaha R-50 helicopter employed 30 states and was used for H_∞ -based control system design for the helicopter in steady-state hovering or slow forward and backward flight (LaCivita et al. 2002a). The authors further compared accuracy of full linearized models and reduced-order linear models and came to similar conclusions with those previously noted by Mettler (2002), which advocated the use of first-order models for tip-path-plane flapping dynamics.

Abbeel et al. (2006) built a nonlinear model of a small-scale aerobatic helicopter, using a combination of first-principles modeling and data fitting based on reinforcement learning. The model used fewer states than the one described in this chapter: it included rigid body states and main rotor speed only. The authors used the model and apprenticeship learning algorithms to develop control laws that enabled tracking

of highly aggressive aerobatic trajectories (Abbeel et al. 2010) and autonomous autorotation landing (Abbeel et al. 2008). Under this approach, time histories from aerobatic flights conducted by an expert pilot were used to refine the model and tailor control law parameters to a particular task.

This chapter presents a development of a first-principles-based nonlinear dynamic model of a miniature helicopter, which proved sufficiently accurate to predict helicopter dynamics in a wide range of conditions, including aerobatic flight. This work is closely based on a modeling chapter in the author's doctoral dissertation (Gavrilets 2003), devoted to human-inspired approach to autonomous helicopter aerobatics. Most of the model parameters were measured directly, and several were estimated using data from flight-test experiments. Analytical linearization of the model with respect to forward speed yields simple models, directly used for design of feedback control laws for autonomous execution of aerobatic maneuvers. The helicopter used in the study is X-Cell .60 SE by Miniature Aircraft (1999), equipped with a 0.90 size engine and an electronic governor. This helicopter features a particularly stiff hub, which gives large control authority to cyclic actuators. The coupling in the hub for this helicopter was also shown to be negligible, which further simplified model development.

The helicopter flew a fully autonomous mission that included an axial roll, a hammerhead, and a Split-S and sequences of these maneuvers (Gavrilets et al. 2002). An artist representation of a Split-S maneuver is shown in Fig. 15.1. The state trajectories in both open loop and closed loop flights were well predicted by the nonlinear model that used a single set of parameters, invariant of the task being performed.

The work on the dynamic model was performed using a custom-designed and built avionics suite, which was concurrently developed. The avionics suite is described in detail in the author's Ph.D. dissertation (Gavrilets 2003).

The model has typical rigid body states with the quaternion attitude representation used in order to enable simulation of extreme attitudes (Rolfe and Staples 1986), two states for the lateral and longitudinal flapping angles, one for the rotor speed, and one for the integral of the rotor speed tracking error. This last state comes from the governor action, modeled with a proportional-integral feedback from the rotor speed tracking error to the throttle command. The model covers a large portion of the X-Cell's natural flight envelope: from hover flight to about 20 m/s forward flight. The maximum forward speed corresponds to an advance ratio $\mu = 0.15$, which is considered as relatively low (Padfield 1996), and permits a number of assumptions (e.g., thrust perpendicular to the rotor disk; see Chen (1979)). The cross-coupling effects in the rotor hub were also shown to be negligible for this helicopter, which further simplified model development. The mathematical model was developed using basic helicopter theory, accounting for the particular characteristics of a miniature helicopter. Most of the parameters were measured directly, and several were estimated using data collected from simple flight-test experiments, involving step and pulse responses in various actuator inputs. No formal system identification procedures are required for the proposed model structure. The model's accuracy was verified using comparison between model predicted responses and responses

Fig. 15.1 Miniature helicopter undergoing a Split-S maneuver. The same maneuver was performed fully autonomously under control laws, developed based on the dynamic model described in this chapter (Drawing courtesy of Popular Mechanics magazine)



collected during flight-test data. The model was also “flown” in simulator by an expert RC pilot to determine how well it reproduces the piloted flying qualities.

Analytical linearization of the model with respect to forward speed was used to derive simple linear models. These were subsequently used for a model-based design of the controllers used for the automatic execution of aerobatic maneuvers. The actual aggressive trajectories flown by the helicopter were adequately predicted by the simulation based on the developed nonlinear model.

The remainder of the chapter contains a full list of model parameters with the numerical values, dynamics equations of motion, and expressions for forces and moments exerted on the helicopter by its components. Flight-test data used to validate the model for various flight regimes, including aerobatics, is provided throughout the chapter.

15.2 Helicopter Parameters

The physical helicopter parameters used for the model are given in Table 15.1. The moments of inertia around the aircraft body axes passing through the vehicle center of gravity were determined using torsional pendulum tests (Harris 1996). The cross-axis moments of inertia are hard to measure without a balancing device, and since they are usually small, they were neglected. The X-Cell main and tail rotors, as well as the stabilizer bar, have symmetric airfoils. The lift curve slopes of these surfaces were estimated according to their respective aspect ratios (Kueth and Chow 1986).

Table 15.1 Parameters of MIT instrumented X-Cell 60 SE helicopter

| Parameter | Description |
|---|--|
| $m = 8.2 \text{ kg}$ | Helicopter mass |
| $I_{xx} = 0.18 \text{ kg m}^2$ | Rolling moment of inertia |
| $I_{yy} = 0.34 \text{ kg m}^2$ | Pitching moment of inertia |
| $I_{zz} = 0.28 \text{ kg m}^2$ | Yawing moment of inertia |
| $K_\beta = 54 \text{ N-m/rad}$ | Hub torsional stiffness |
| $\gamma_{fb} = 0.8$ | Stabilizer bar Lock number |
| $B_{\delta_{lat}}^{\text{nom}} = 4.2 \text{ rad/rad}$ | Lateral cyclic to flap gain at nominal rpm |
| $A_{\delta_{lon}}^{\text{nom}} = 4.2 \text{ rad/rad}$ | Longitudinal cyclic to flap gain at nominal rpm |
| $K_\mu = 0.2$ | Scaling of flap response to speed variation |
| $\Omega_{\text{nom}} = 167 \text{ rad/s}$ | Nominal m.r. speed |
| $R_{\text{mr}} = 0.775 \text{ m}$ | m.r. radius |
| $c_{\text{mr}} = 0.058 \text{ m}$ | m.r. chord |
| $a_{\text{mr}} = 5.5 \text{ rad}^{-1}$ | m.r. blade lift curve slope |
| $C_{D_0}^{\text{mr}} = 0.024$ | m.r. blade zero lift drag coefficient |
| $C_{T_{\text{max}}}^{\text{mr}} = 0.0055$ | m.r. max thrust coefficient |
| $I_{\beta_{\text{mr}}} = 0.038 \text{ kg m}^2$ | m.r. blade flapping inertia |
| $R_{\text{tr}} = 0.13 \text{ m}$ | t.r. radius |
| $c_{\text{tr}} = 0.029 \text{ m}$ | t.r. chord |
| $a_{\text{tr}} = 5.0 \text{ rad}^{-1}$ | t.r. blade lift curve slope |
| $C_{D_0}^{\text{tr}} = 0.024$ | t.r. blade zero lift drag coefficient |
| $C_{T_{\text{max}}}^{\text{tr}} = 0.05$ | t.r. max thrust coefficient |
| $n_{\text{tr}} = 4.66$ | Gear ratio of t.r. to m. r. |
| $n_{\text{es}} = 9.0$ | Gear ratio of engine shaft to m. r. |
| $\delta_r^{\text{trim}} = 0.1 \text{ rad}$ | t.r. pitch trim offset |
| $S_{\text{vf}} = 0.012 \text{ m}^2$ | Effective vertical fin area |
| $C_{L_{\alpha}}^{\text{vf}} = 2.0 \text{ rad}^{-1}$ | Vertical fin lift curve slope |
| $\epsilon_{\text{vf}}^{\text{tr}} = 0.2$ | Fraction of vertical fin area exposed to t.r. induced velocity |
| $S_{\text{ht}} = 0.01 \text{ m}^2$ | Horizontal fin area |
| $C_{L_{\alpha}}^{\text{ht}} = 3.0 \text{ rad}^{-1}$ | Horizontal tail lift curve slope |
| $P_{\text{eng}}^{\text{idle}} = 0.0 \text{ W}$ | Engine idle power |
| $P_{\text{eng}}^{\text{max}} = 2,000.0 \text{ W}$ | Engine max power |
| $K_p = 0.01 \text{ s/rad}$ | Proportional governor gain |
| $K_i = 0.02 \text{ 1/rad}$ | Integral governor gain |
| $S_x^{\text{fus}} = 0.1 \text{ m}^2$ | Frontal fuselage drag area |
| $S_y^{\text{fus}} = 0.22 \text{ m}^2$ | Side fuselage drag area |
| $S_z^{\text{fus}} = 0.15 \text{ m}^2$ | Vertical fuselage drag area |
| $h_{\text{mr}} = 0.235 \text{ m}$ | m.r. hub height above c.g. |
| $l_{\text{tr}} = 0.91 \text{ m}$ | t.r. hub location behind c.g. |
| $h_{\text{tr}} = 0.08 \text{ m}$ | t.r. height above c.g. |
| $l_{\text{ht}} = 0.71 \text{ m}$ | Stabilizer location behind c.g. |

The effective torsional stiffness in the hub was estimated from angular rate responses to step commands in cyclic, as described in Sect. 15.4.1.3.

Note that in the table, “m.r.” stands for the main rotor, and “t.r.” stands for the tail rotor.

15.3 Equations of Motion

The rigid body equations of motion for a helicopter are given by the Newton-Euler equations shown below. Here the cross products of inertia are neglected:

$$\dot{u} = vr - wq - g \sin \theta + (X_{mr} + X_{fus}) / m$$

$$\dot{v} = wp - ur + g \sin \phi \cos \theta + (Y_{mr} + Y_{fus} + Y_{tr} + Y_{vf}) / m$$

$$\dot{w} = uq - vp + g \cos \phi \cos \theta + (Z_{mr} + Z_{fus} + Z_{ht}) / m$$

$$\dot{p} = qr(I_{yy} - I_{zz}) / I_{xx} + (L_{mr} + L_{vf} + L_{tr}) / I_{xx}$$

$$\dot{q} = pr(I_{zz} - I_{xx}) / I_{yy} + (M_{mr} + M_{ht}) / I_{yy}$$

$$\dot{r} = pq(I_{xx} - I_{yy}) / I_{zz} + (-Q_e + N_{vf} + N_{tr}) / I_{zz}$$

The set of forces and moments acting on the helicopter are organized by components: $()_{mr}$ for the main rotor, $()_{tr}$ for the tail rotor, $()_{fus}$ for the fuselage (includes fuselage aerodynamic effects), $()_{vf}$ for the vertical fin, and $()_{ht}$ for the horizontal stabilizer. These forces and moments are shown along with the main helicopter variables in Fig. 15.2. Q_e is the torque produced by the engine to counteract the aerodynamic torque on the main rotor blades. The helicopter blades rotate clockwise when viewed from above; therefore, $Q_e \geq 0$. In the above equations, it was assumed that the fuselage center of pressure coincides with the c.g.; therefore, the moments created by the fuselage aerodynamic forces were neglected.

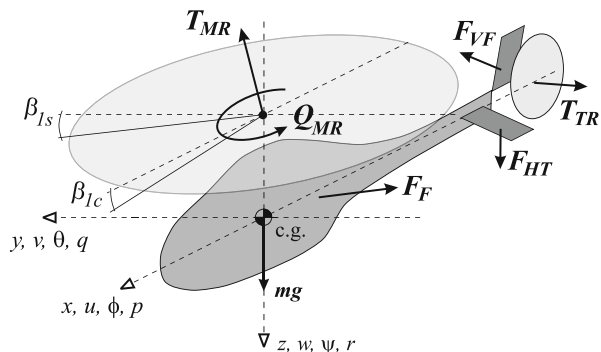


Fig. 15.2 Moments and forces acting on helicopter

The rotational kinematic equations were mechanized using quaternions (Rolfe and Staples 1986). The inertial velocities are derived from the body-axis velocities by a coordinate transformation (flat-Earth equations are used) and integrated to obtain inertial position. A fourth-order Runge-Kutta integration method is used, with an integration step of 0.01 s.

15.4 Component Forces and Moments

15.4.1 Main Rotor Forces and Moments

15.4.1.1 Thrust

For the main rotor thrust, it was assumed that the inflow is steady and uniform. According to Padfield (1996, p. 126), the time constant for settling of the inflow transients at hover is given by

$$\tau_\lambda = \frac{0.849}{4\lambda_{\text{trim}}\Omega_{\text{mr}}} \quad (15.1)$$

Induced velocity at hover trim condition can be determined from simple momentum theory:

$$V_{\text{imr}} = \sqrt{\frac{mg}{2\rho\pi R_{\text{mr}}^2}} = 4.2 \text{ m/s} \quad (15.2)$$

The tip speed of the main rotor is $V_{\text{mr}}^{\text{tip}} = \Omega_{\text{mr}} R_{\text{mr}} = 125.7 \text{ m/s}$, from which the inflow ratio is $\lambda_{\text{imr}} = V_{\text{imr}}/V_{\text{mr}}^{\text{tip}} = 0.033$. Therefore, the time it takes for the inflow to settle is $\tau_\lambda = 0.038 \text{ s}$, which is significantly faster than the rigid body dynamics. During the maneuvers requiring large thrust variations, the time constant may change substantially. However, as shown in the section on the main rotor flapping dynamics, the X-Cell cyclic control authority is dominated by the hub torsional stiffness, which makes the modeling of the inflow transients less critical.

A momentum theory-based iterative scheme given by Padfield (1996, p. 123) was adapted to compute the thrust coefficient and inflow ratio as a function of airspeed, rotor speed, and collective setting. Flapping angles were neglected in the computation of the rotor thrust. The blades of the main rotor have no twist. The influence of the cyclics and the roll rate on thrust are of second order for advance ratio range $\mu < 0.15$ and were neglected as well. An empirically determined maximum thrust coefficient was introduced, since momentum theory does not take into account the effect of blade stall.

The thrust coefficient is given by (omitting the “mr” index)

$$C_T = \frac{T}{\rho (\Omega R)^2 \pi R^2} \quad (15.3)$$

where T is the main rotor thrust. Then, the following system of equations can be solved iteratively:

$$\lambda_0 = \frac{C_T}{2\eta_w \sqrt{\mu^2 + (\lambda_0 - \mu_z)^2}} \quad (15.4)$$

$$C_T^{\text{ideal}} = \frac{a\sigma}{2} \left(\theta_0 \left(\frac{1}{3} + \frac{\mu^2}{2} \right) + \frac{\mu_z - \lambda_0}{2} \right) \quad (15.5)$$

$$C_T = \begin{cases} C_T^{\text{ideal}} & \text{if } -C_T^{\text{max}} \leq C_T^{\text{ideal}} \leq C_T^{\text{max}} \\ -C_T^{\text{max}} & \text{if } C_T^{\text{ideal}} < -C_T^{\text{max}} \\ C_T^{\text{max}} & \text{if } C_T^{\text{max}} < C_T^{\text{ideal}} \end{cases} \quad (15.6)$$

$$C_T^{\text{max}} = \frac{T^{\text{max}}}{\rho (\Omega R)^2 \pi R^2} \quad (15.7)$$

Here

$$\mu = \frac{\sqrt{(u - u_{\text{wind}})^2 + (v - v_{\text{wind}})^2}}{\Omega R} - \text{advance ratio}$$

$$\mu_z = \frac{w - w_{\text{wind}}}{\Omega R} - \text{normal airflow component}$$

$$\sigma = \frac{2c}{\pi R} - \text{solidity ratio}$$

a – lift curve slope

θ_0 – commanded collective angle

η_w – coefficient of nonideal wake contraction

$T^{\text{max}} = 2.5 \text{ mg}$ – maximum rotor thrust

Based on momentum theory, the rotor wake far downstream contracts by a factor of 2 (Padfield 1996, p. 116). A coefficient η_w was introduced to account for nonideal wake contraction and the power lost due to the nonuniform velocity and pressure distribution in the wake. An approximate value for this coefficient was determined to be $\eta_w = 0.9$. Hence, the iterative scheme given in Padfield (1996, p. 123) is modified as follows. First, define the zero function:

$$g_0 = \lambda_0 - \frac{C_T}{2\eta_w \Lambda^{1/2}}, \text{ where}$$

$$\Lambda = \mu^2 + (\lambda_0 - \mu_z)^2$$

and thrust coefficient C_T is given by Eq. (15.5). Apply Newton's iterative scheme:

$$\lambda_{0_{j+1}} = \lambda_{0_j} + f_j h_j (\lambda_{0_j})$$

$$h_j = - \left(\frac{g_0}{dg_0/d\lambda_0} \right)_{\lambda_0=\lambda_{0_j}}$$

An explicit expression for h_j :

$$h_j = - \frac{(2\eta_w \lambda_{0_j} \Lambda^{1/2} - C_T) \Lambda}{2\eta_w \Lambda^{3/2} + \frac{a\sigma}{4} \Lambda - C_T (\mu_z - \lambda_{0_j})}$$

Padfield (1996, p. 123) suggests a constant value of the convergence rate coefficient $f_j = 0.6$.

Note that at hover the denominator of Eq. (15.4) is zero when the vertical velocity is equal to the inflow velocity. This condition corresponds to a vortex-ring state, which cannot be modeled adequately by the momentum theory. Instead, the denominator is numerically separated from zero. In general, this condition is avoided in flight because it leads to a loss of control. One has to keep in mind that the simulation does not adequately represent the helicopter dynamics when vortex-ring conditions exist on either the main or the tail rotor. Furthermore, strictly speaking, the momentum theory applies only to a fully developed steady-state flow in ascending flight. Empirical corrections for descending flight, cited by Padfield (1996, p. 118), could be used to make thrust prediction somewhat more accurate.

The momentum theory approach was previously shown to be adequate for estimating steady-state main rotor thrust both at hover and in fast forward flight. Results of the wind tunnel tests with a 5-ft-diameter rotor are given by Harris (1972) and summarized by Bramwell (2001, pp. 109–114). It was shown that the blade lift curve slope coefficient a can be determined from experiments such that the momentum theory accurately predicts thrust for a wide range of advance ratios and collective pitch angles. To test the applicability of momentum theory-based thrust calculation to transient response, flight data was gathered for collective pitch pulse responses at hover. At hover, the vertical acceleration can be represented by a linear relation:

$$a_z = Z_w w + Z_{\text{col}} \delta_{\text{col}} \quad (15.8)$$

The vertical speed damping stability derivative Z_w and the collective pitch control derivative Z_{col} can be obtained analytically by linearization of the momentum theory equations (Padfield 1996, pp. 219, 229). For hover,

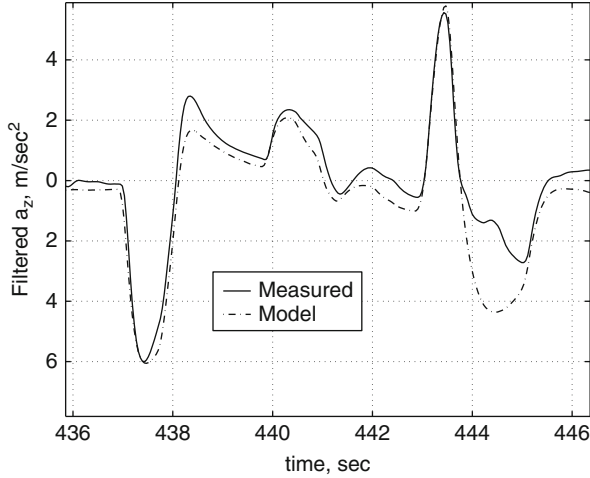


Fig. 15.3 Modeling vertical acceleration response at hover

$$Z_w = -\frac{\rho (\Omega R) \pi R^2}{m} \frac{2a\sigma\lambda_0}{16\lambda_0 + a\sigma} \quad (15.9)$$

$$Z_{col} = -\frac{\rho (\Omega R)^2 \pi R^2}{m} \frac{8}{3} \frac{a\sigma\lambda_0}{16\lambda_0 + a\sigma} \quad (15.10)$$

The same value for the blade lift curve slope was used as the one determined for the particular airfoil used in the tests summarized by Harris (1972), $a = 5.5$. This value is consistent with the high aspect ratio of the main rotor blades, if the blades are considered as a wing (Kuethe and Chow 1986). The mean values were subtracted from the collective command and the vertical acceleration measurement; the signals were filtered with a first-order low-pass filter with time constant of 0.2 s. The digital models of the analog low-pass filter and the servomechanism were applied to the collective command for consistency. Figure 15.3 shows the comparison of the computed vertical acceleration from Eq. (15.8) and actual acceleration. As can be seen, the model, based on linearization of the momentum theory, agrees well with flight data.

15.4.1.2 Torque

The main rotor torque can be approximated as a sum of induced torque due to generated thrust and torque due to profile drag on the blades (Padfield 1996, p. 116):

$$C_Q = \frac{Q}{\rho (\Omega R)^2 \pi R^3} = C_T (\lambda_0 - \mu_z) + \frac{C_{D_0}\sigma}{8} \left(1 + \frac{7}{3}\mu^2\right) \quad (15.11)$$

where C_Q is the torque coefficient and C_{D_0} is the profile drag coefficient of the main rotor blade. The profile drag is not significantly affected by changes in the collective setting. Thus, the yawing moment produced by the main rotor is given by

$$Q_{mr} = C_Q \rho (\Omega R)^2 \pi R^3 \quad (15.12)$$

Profile drag coefficient of the main rotor blade was estimated as $C_{D_0} = 0.024$. Underestimation of the profile drag coefficient would lead to overprediction of the main rotor speed in windmilling flight conditions, which occur during autorotation and some agile maneuvers.

Except for the hover condition, the rotor in-plane force, which contributes to the drag and side force, is substantially smaller than the drag provided by the fuselage and the side force from the fuselage and the empennage. This force was neglected in the calculations; effectively it was lumped with the fuselage forces, estimated in Sect. 15.4.3. The moments due to the in-plane force are much smaller than those due to the blade flapping, since the torsional stiffness of the hub retention is high on the X-Cell.

15.4.1.3 Main Rotor Moments and Flapping Dynamics

The main rotor flapping angle β can be represented as a Fourier series of the blade azimuth angle ψ , with only the first three coefficients retained (Padfield 1996, p. 32):

$$\beta(\psi) = a_0 + a_1 \cos \psi + b_1 \sin \psi \quad (15.13)$$

Flapping of the teetering stabilizer bar can be represented by a similar equation without the constant term since no coning takes place:

$$\beta_s(\psi) = a_{1s} \sin \psi + b_{1s} \cos \psi \quad (15.14)$$

Stabilizer bar flapping contributes to the change of the main rotor blade pitch angle through a mechanical linkage:

$$\theta(\psi) = \theta_0 + \theta_{lon} \sin \psi + \theta_{lat} \cos \psi + k_s \beta_s \quad (15.15)$$

The swashplate deflections change the cyclic pitch angle of both the main rotor and the stabilizer bar. Coupled second-order differential equations can be developed for Fourier coefficients of the main rotor and stabilizer bar flapping. It can be shown (Padfield 1996, pp. 33–35) that the undamped natural frequency of the flapping motion is close to the rotor speed Ω_{mr} and the damping ratio can be approximated by $\gamma/8$, where γ is the Lock number of the blades being considered (main rotor or stabilizer bar). The Lock number represents the ratio of aerodynamic to inertial forces and is defined as

$$\gamma = \frac{\rho c a R^4}{I_\beta} \quad (15.16)$$

For the main rotor blades, the Lock number is relatively high, $\gamma_{mr} \approx 3.7$; therefore, the flapping motion is well damped. For a step response, this corresponds to the settling time (to within 5 % of the steady-state value) of $24/\gamma\Omega = 0.039$ s. For the stabilizer bar, with its small aerodynamic surfaces, the Lock number is low, $\gamma_{tb} \approx 0.8$, and the corresponding settling time is 0.144 s. Earlier work on modeling of small-scale rotorcraft with Bell-Hiller stabilizer bars (Mettler et al. 2002b; Mettler 2002; LaCivita et al. 2002b) showed that the main rotor and stabilizer bar flapping dynamics can be lumped and represented by tip-path plane (TPP) flapping dynamics with only two states. This result was based on frequency-domain identification and comparison of reduced and full order transfer functions for attitude dynamics. Furthermore, coupling of the lumped flapping dynamics and rigid body pitch and roll motions leads to pronounced second-order characteristics (von Grunhagen et al. 1996; Talbot et al. 1982; Mettler 2002; LaCivita et al. 2002b). These modes are lightly damped and should be explicitly accounted for in designing high-bandwidth attitude or rate control systems (Mettler et al. 2000, 2002a). The lateral and longitudinal flapping dynamics were represented by the first-order equations:

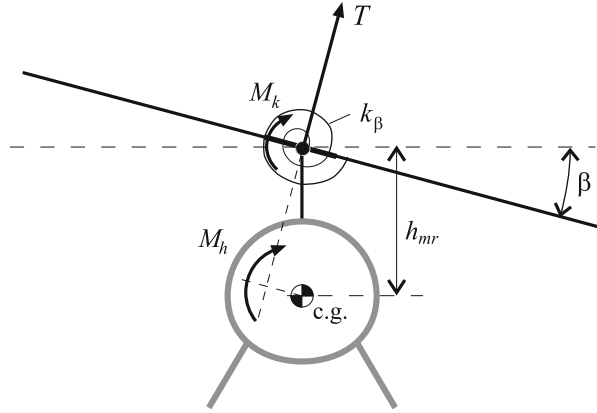
$$\dot{b}_1 = -p - \frac{b_1}{\tau_e} - \frac{1}{\tau_e} \frac{\partial b_1}{\partial \mu_v} \frac{v - v_w}{\Omega R} + \frac{B_{\delta_{lat}}}{\tau_e} \delta_{lat} \quad (15.17)$$

$$\dot{a}_1 = -q - \frac{a_1}{\tau_e} + \frac{1}{\tau_e} \left(\frac{\partial a_1}{\partial \mu} \frac{u - u_w}{\Omega R} + \frac{\partial a_1}{\partial \mu_z} \frac{w - w_w}{\Omega R} \right) + \frac{A_{\delta_{lon}}}{\tau_e} \delta_{lon} \quad (15.18)$$

where $B_{\delta_{lat}}$ and $A_{\delta_{lon}}$ are effective steady-state lateral and longitudinal gains from the cyclic inputs to the main rotor flap angles; δ_{lat} and δ_{lon} are the lateral and longitudinal cyclic control inputs (pilot stick or control system outputs); u_w , v_w , and w_w are the wind components along, respectively, X, Y, and Z helicopter body axes; and τ_e is the effective rotor time constant for a rotor with the stabilizer bar. Frequency-domain identification showed that the pitch and roll cross-coupling flapping coefficients are approximately an order of magnitude less than the direct coefficients for the X-Cell (Mettler 2002) and were neglected. This result holds for large-amplitude inputs as well. For example, during an axial roll maneuver, the pitch rate remains close to zero with no pilot compensation, similarly to the roll rate during a loop. This natural decoupling of the cyclic responses makes the X-Cell a particularly attractive helicopter for aerobatics.

The dominant rotor moments are the control moments produced by the rotor flapping. In the following pages, the moments in the roll direction are described (resulting from the lateral TPP flapping b_1). Figure 15.4 shows the rotor moments that are acting on the fuselage. The first contribution results from the restraint in the blade attachment to the rotor head. The restraint can be approximated using a linear torsional spring with a constant stiffness coefficient K_β , resulting in a roll moment $M_{k,lat} = K_\beta b_1$. The second contribution results from the tilting of the thrust vector. Assuming that the thrust vector is perpendicular to the TPP, the thrust vector will tilt proportionally to the rotor flapping angles. The moment arm is the distance h_{mr}

Fig. 15.4 Rotor moments acting on the helicopter fuselage



between the rotor head and the helicopter center of gravity, resulting in a lateral moment $M_{h,lat} = T h_{mr} b_1$. The total main rotor rolling moment, entering the rigid body equations of motion, is represented by Eq. (15.19):

$$L_{mr} = (K_\beta + T h_{mr}) b_1 \quad (15.19)$$

Similarly, the pitching moment is given by Eq. (15.20):

$$M_{mr} = (K_\beta + T h_{mr}) a_1 \quad (15.20)$$

To determine the parameters entering the flapping equations, one can split up the problem into “slow” and “fast” dynamics. First, one can notice that the dihedral derivatives are important only at the low-frequency spectrum of the dynamics. At high frequencies (above 0.5 Hz), the transfer functions from cyclic inputs to angular rates can be approximated by second-order transfer functions, derived by omitting the translational flapping derivatives in Eqs. (15.17) and (15.18) and combining it with Eqs. (15.19) and (15.20):

$$\frac{q}{\delta_{lon}} \approx \frac{A_{\delta_{lon}}}{\tau_e} \frac{\omega_{nq}^2}{s^2 + 1/\tau_e s + \omega_{nq}^2} \quad (15.21)$$

$$\frac{p}{\delta_{lat}} \approx \frac{B_{\delta_{lat}}}{\tau_e} \frac{\omega_{np}^2}{s^2 + 1/\tau_e s + \omega_{np}^2} \quad (15.22)$$

Pitching dynamics in fast forward flight is significantly influenced by the horizontal tail, which provides a stabilizing effect, and the main rotor flapping due to vertical speed, which provides a destabilizing effect. Therefore, the longitudinal cyclic to pitch rate transfer function given in Eq. (15.21) is valid in low-speed flight only. Here the undamped natural frequencies of the longitudinal and lateral fuselage-rotor modes are

$$\omega_{nq} = \sqrt{\frac{T_{mr}h_{mr} + K_\beta}{I_{yy}}}$$

$$\omega_{np} = \sqrt{\frac{T_{mr}h_{mr} + K_\beta}{I_{xx}}}$$

Note that for hover and straight and level flight, $T_{mr} \approx mg$. The distance between the main rotor hub and the helicopter center of gravity can be measured. The moments of inertia were determined with the torsional pendulum tests (Harris 1996). The natural frequencies of the lightly damped second-order systems can be easily determined by counting oscillation periods in a recorded step response, thereby providing an estimate of the hub torsional stiffness. These parameters are given in Table 15.1. An approximate value of the damping time constant for the flapping motion (Mettler 2002) is given in Eq. (15.23):

$$\tau_e = \frac{16}{\gamma_{fb}\Omega_{mr}} \approx 0.1 \text{ s} \quad (15.23)$$

Note that the damping is proportional to the stabilizer bar Lock number, making it small. The steady-state cyclic to rate gains depend on the swashplate gearing. Experiments have also indicated that the values of $B_{\delta_{lat}}$ and $A_{\delta_{lon}}$ grow with the rotor speed. This effect was approximated as a function of effective dynamic pressure, or square of the rotor speed:

$$B_{\delta_{lat}} = B_{\delta_{lat}}^{\text{nom}} \left(\frac{\Omega}{\Omega_{\text{nom}}} \right)^2 \text{ rad/rad}$$

$$A_{\delta_{lon}} = A_{\delta_{lon}}^{\text{nom}} \left(\frac{\Omega}{\Omega_{\text{nom}}} \right)^2 \text{ rad/rad}$$

$$\Omega_{\text{nom}} = 167 \text{ rad/s}$$

These gains were determined by matching DC gain of the angular rate responses to steps in cyclics. Final verification of the derived parameters is provided by simulation of the linear systems described in Eqs. (15.21) and (15.22) and comparison with the flight-test data. Figure 15.5 shows the actual and simulated roll rates for a segment including an axial roll maneuver. The helicopter undergoes negative rotor loading during inverted portion of the maneuver, leading to 15% lower roll rate in that segment than predicted by the simplified linear model. Figure 15.6 shows the actual and simulated pitch rates for a segment of 15 m/s forward flight with pulse commands on longitudinal cyclic. Note that the linearized models of angular rate dynamics are adequate for both small- and large-amplitude motion.

The flapping due to translational velocity is described by the flapping derivatives $\partial a_1/\partial \mu$ and $\partial b_1/\partial \mu_v$. The longitudinal flapping due to the forward speed increase

Fig. 15.5 Actual and model roll rate response during axial roll maneuver

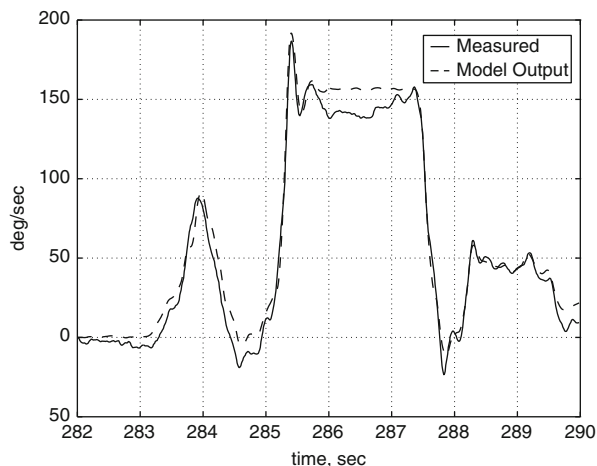
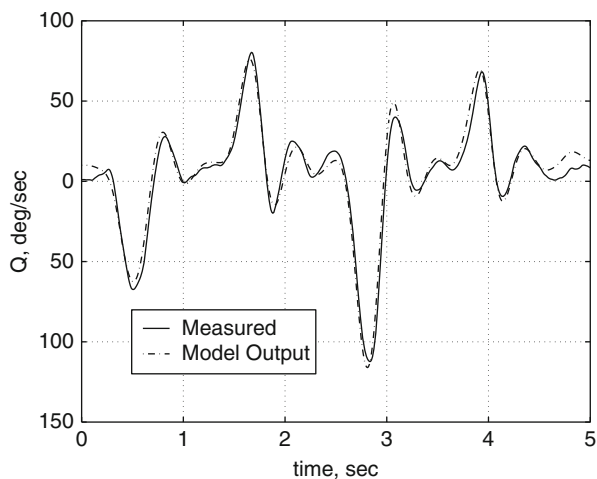


Fig. 15.6 Actual and model pitch rate response in low-speed flight



is caused by an increased lift on the advancing blade with respect to the retreating blade, which turns into a flap-back moment on the main rotor due to the 90° gyroscopic phase lag. A theoretical value for the steady-state longitudinal flapping for a rotor without a stabilizer bar is given by Bramwell (2001, p. 107)

$$a_1 = \frac{2\mu (4\delta_{col}/3 - \lambda_0)}{1 + 3\mu^2/2} \approx 2\mu (4\delta_{col}/3 - \lambda_0) \quad (15.24)$$

While this expression is valid for a teetering rotor at hover, theoretical approximation for a hingeless rotor without a stabilizer bar is very close (Padfield 1996). The stabilizer bar dramatically reduces flapping response to gusts, and

Eq. (15.24) cannot be used for predicting dihedral effect on a rotor equipped with one. Since this derivative plays a primary role in the frequency and damping of the phugoid mode, which is very slow, it is difficult to estimate with the frequency-domain identification methods (Mettler 2002). An open loop excitation would have to last for much longer than it takes for the helicopter to diverge; therefore, the necessary pilot's feedback would bias an estimate of the derivative (Ljung 1999). A scaling coefficient was introduced in Eq. (15.24), which was linearized to yield

$$\frac{\partial a_1}{\partial \mu} = 2K_\mu \left(\frac{4\delta_{\text{col}}}{3} - \lambda_0 \right) \quad (15.25)$$

A rough estimate for the scaling coefficient K_μ can be obtained by matching the steady-state cyclic input in forward flight at constant speed (maintained with the velocity-tracking feedback controller) with that predicted by the simulation in the same conditions. An estimate for X-Cell 60 yielded $K_\mu = 0.2$, which implies that the stabilizer bar reduces steady-state flapping response to forward speed by a factor of 5. From Eq. (15.24) and the rotor symmetry, one can conclude that the longitudinal and lateral dihedral derivatives are equal in magnitude and, in both cases, cause the rotor to flap away from the incoming air:

$$\frac{\partial b_1}{\partial \mu_v} = -\frac{\partial a_1}{\partial \mu} \quad (15.26)$$

Positive Z-axis velocity causes higher lift on advancing blade, which results in a flap back of the rotor; this effect is captured by the stability derivative $\partial a_1 / \partial \mu_z$ in Eq. (15.17). An analytical estimate of the derivative is adapted (Bramwell 2001, p. 159) to accommodate backward flight and scaled by the same coefficient K_μ to reflect the effect of the stabilizer bar:

$$\frac{\partial a_1}{\partial \mu_z} = K_\mu \frac{16\mu^2}{(1 - \mu^2/2)(8|\mu| + a\sigma)} \text{sign } \mu \approx K_\mu \frac{16\mu^2}{8|\mu| + a\sigma} \text{sign } \mu \quad (15.27)$$

15.4.1.4 Rotor Forces

For small advance ratio flight ($\mu < 0.15$), one can assume that the thrust vector is perpendicular to the TPP. The small flapping angles (below 10°) allow one to use linear approximation for the main rotor force components along the helicopter body axes. As was stated above, the in-plane rotor force was lumped with the fuselage forces and is not accounted for in the equations below:

$$X_{\text{mr}} = -T_{\text{mr}}a_1$$

$$Y_{\text{mr}} = T_{\text{mr}}b_1$$

$$Z_{\text{mr}} = -T_{\text{mr}}$$

15.4.2 Engine, Governor, and Rotor Speed Model

The rotor speed dynamics is modeled by the following equation:

$$\dot{\Omega} = \dot{r} + \frac{1}{I_{\text{rot}}} [Q_e - Q_{\text{mr}} - n_{\text{tr}} Q_{\text{tr}}] \quad (15.28)$$

where Q_e is the engine torque (positive clockwise), $Q_{\text{mr}} = C_Q \rho (\Omega R)^2 \pi R^3$ is the main rotor torque (positive counterclockwise), Q_{tr} is the tail rotor torque, n_{tr} is the tail rotor gear ratio, I_{rot} is the total rotating inertia referenced to the main rotor speed, and Ω is the rotor speed. The engine torque depends on the throttle setting δ_t and rotor speed and is usually represented by engine maps or lookup tables. The maps for the engine were not available, and a simplified representation of the engine torque is suggested. Assume that engine power is proportional to the throttle setting:

$$P_e = P_e^{\max} \delta_t \quad (15.29)$$

where $0 < \delta_t < 1$. Then, the torque is

$$Q_e = \frac{P_e}{\Omega} \quad (15.30)$$

The engine torque response to throttle changes can be considered instantaneous, since the time lags associated with air intake, fuel flow, and combustion are very small compared to vehicle dynamics.

In the absence of manufacturer data, the governor can be modeled as a proportional-integral feedback controller, maintaining commanded rotor speed by changing the throttle:

$$\begin{aligned} \delta_t &= K_p \cdot (\Omega_c - \Omega) + K_i \cdot \omega_i \\ \dot{\omega}_i &= \Omega_c - \Omega \end{aligned} \quad (15.31)$$

where Ω_c is the rotor speed command and K_p and K_i are proportional and integral feedback gains. The anti-windup logic resets the integrator state value ω_i in case computed throttle command is saturated. Throttle servo dynamics is much faster than the rotor speed dynamics and was neglected in the model.

To determine the parameters of the given engine/governor model, time response to the rotor speed step command was analyzed. Consider linearization of Eqs. (15.28)–(15.31) around a nominal operating point, for example, hovering flight, neglecting the yawing acceleration and the tail rotor torque. The states of the linear system will be the rotor speed deviation from the nominal ω and an integral of the rotor speed tracking error ω_i . The inputs are ω_c , a variation of the rotor speed command, and δ_c , a variation of the collective angle from the trim setting. The resulting linear system is given in Eq. (15.32):

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \omega \\ \omega_i \end{bmatrix} = & \begin{bmatrix} -\frac{1}{\Omega I_{\text{rot}}} (3Q_{\text{mr}}^0 + P_e^{\text{max}} K_p) & \frac{P_e^{\text{max}} K_i}{\Omega I_{\text{rot}}} \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ \omega_i \end{bmatrix} \\ & + \begin{bmatrix} \frac{P_e^{\text{max}} K_p}{\Omega I_{\text{rot}}} & -\frac{1}{I_{\text{rot}} C_Q} \frac{\partial C_Q}{\partial \delta_c} Q_{\text{mr}}^0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_c \\ \delta_c \end{bmatrix} \end{aligned} \quad (15.32)$$

From Eq. (15.11), the main rotor torque at hover Q_{mr}^0 can be computed, and for the X-Cell with the parameters given in Table 15.1, $Q_{\text{mr}}^0 \approx 6.3 \text{ Nm}$. The characteristic polynomial of the system is given in Eq. (15.33):

$$\chi(\lambda) = \lambda^2 + \lambda \frac{3Q_{\text{mr}}^0 + P_e^{\text{max}} K_p}{\Omega I_{\text{rot}}} + \frac{P_e^{\text{max}} K_i}{\Omega I_{\text{rot}}} \quad (15.33)$$

To estimate the coefficients of the characteristic polynomial, the following test was performed (Sprague et al. 2001). The helicopter was kept at hover at 1,600 rpm, and a 100 rpm step input in rotor speed was commanded to the governor from a remote control. The rotor speed measurement was not available directly in the instrumentation package. Instead, the sound of the engine was recorded with a handheld camcorder. Next, a time-frequency decomposition analysis (Feron et al. 1998) was applied to determine frequency content of the engine noise as a function of time. In such an analysis, a signal of limited duration and frequency, called a wavelet, defined by its central frequency and width, is convoluted in the time domain with the data. The output in the time domain will have a larger magnitude when the wavelet's central frequency is present in the signal than when the central frequency is missing. The spectral content of the signal as a function of time can then be determined by repeating this computation over a range of frequencies. Many frequency bands appear in the sound spectrum of the engine noise, representing harmonics. Figure 15.7 (top) shows the result of the Morlet wavelet time-frequency analysis performed on the engine noise during a step change in RPM setting; two harmonics are indicated. The same behavior appears on each harmonic, providing the opportunity to fine-tune the model to a number of step responses, thereby reducing measurement error due to external noise. The governor/engine system was approximated with a second-order system, whose response to a commanded RPM step input appears at the bottom of Fig. 15.7. The system's damping ratio is $\zeta = 0.63$, and its natural frequency is $\omega_n = 1.3 \text{ rad/s}$. By matching the coefficients of the characteristic polynomials, one can obtain the expressions given in Eq. (15.34):

$$\begin{aligned} 2\zeta\omega_n &= \frac{3Q_{\text{mr}}^0 + P_e^{\text{max}} K_p}{\Omega I_{\text{rot}}} \\ \omega_n^2 &= \frac{P_e^{\text{max}} K_i}{\Omega I_{\text{rot}}} \end{aligned} \quad (15.34)$$

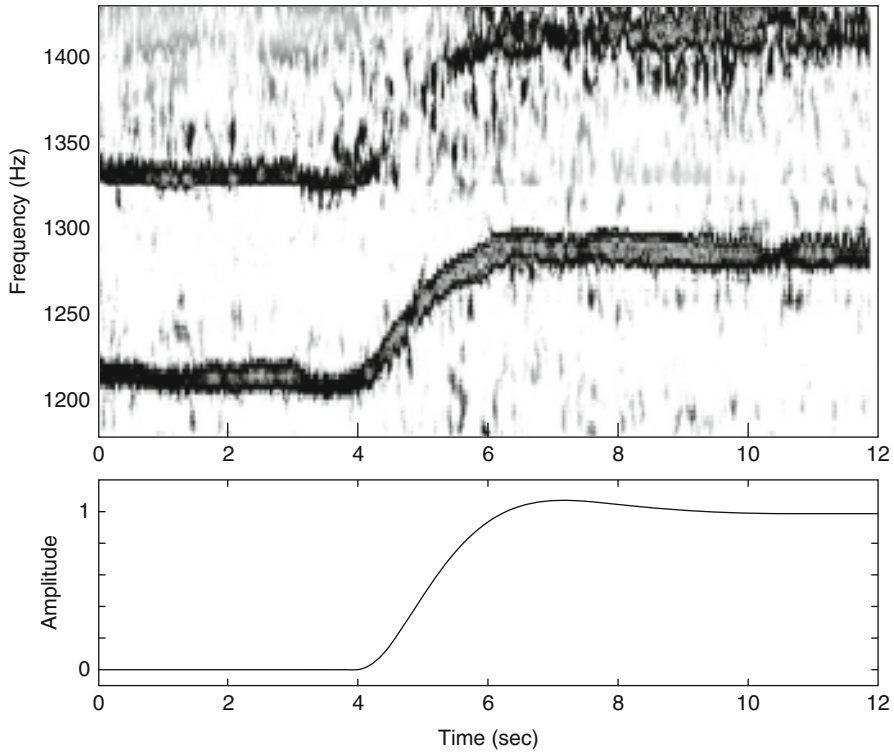


Fig. 15.7 (top) Two frequency bands in the engine noise spectrum; (bottom) simulated response to rotor speed step command

A total kinetic energy of all rotating components is $2I_{\beta_{mr}}\Omega^2 + I_{es}(n_{es}\Omega)^2 + 2I_{\beta_{tr}}(n_{tr}\Omega)^2 = (2I_{\beta_{mr}} + I_{es}n_{es}^2 + 2I_{\beta_{tr}}n_{tr}^2)\Omega^2$, where $I_{\beta_{mr}}$ and $I_{\beta_{tr}}$ are, respectively, the main and the tail rotor blade inertias, I_{es} is the inertia of the engine shaft and all components rotating at the engine speed, n_{tr} is the tail rotor gear ratio, and n_{es} is the engine gear ratio. Therefore, the rotating inertia referenced to the main rotor speed can be represented as $I_{rot} = 2I_{\beta_{mr}} + I_{es}n_{es}^2 + 2I_{\beta_{tr}}n_{tr}^2$. The most important contribution comes from the main rotor blades. The tail rotor inertia, after scaling with the gear ratio squared, amounts to 5 % of the main rotor inertia. The rotating inertia referenced to the engine speed is harder to estimate, but an upper bound can be found by estimating the total mass of rotating components (0.2 kg) and its effective radius of inertia (0.04 m). One thus arrives at an estimate for I_{rot} equal to 2.5 inertias of the main rotor blade. Using this value, and matching coefficients of the characteristic polynomial according to Eq. (15.34), estimates for the proportional and integral governor gains were obtained. The model could be further refined if real-time rotor speed data were available, or the actual governor gain and the engine maps were available from the manufacturer datasheets.

This engine/governor model is simplified. However, it reflects the trends which are important in some very aggressive maneuvers which involve large and rapid variation of the aerodynamic torque on the rotor. First, tight governor feedback keeps the rotor speed close to the nominal setting. Second, an increase in the aerodynamic torque leads to a temporary decrease in rotor speed and a lagged application of the yawing torque to the airframe. The reverse is true for windmilling flight, in which the rotor extracts energy from the air, and leads to an increase in rotor speed, and lagged decrease in torque applied to the airframe. The inaccuracies in the model can make feedforward compensation of the main rotor torque with the tail rotor difficult to tune; a tight yaw rate feedback to the tail rotor pitch is much more effective and is routinely used by R/C pilots in the form of a yaw rate gyro.

15.4.3 Fuselage Forces

For hover flight and forward speeds well below the induced velocity at hover (≈ 4.5 m/s for X-Cell), the rotor downwash is deflected by the forward and side velocity. This deflection creates a force opposing the movement. One can express the X and Y drag forces created by the fuselage in this flight regime by

$$X_{\text{fus}} = S_x^{\text{fus}} \frac{1}{2} \rho V_{\text{imr}}^2 \frac{u}{V_{\text{imr}}}$$

$$Y_{\text{fus}} = S_y^{\text{fus}} \frac{1}{2} \rho V_{\text{imr}}^2 \frac{v}{V_{\text{imr}}}$$

where S_x^{fus} and S_y^{fus} are effective drag areas of the fuselage in the X and Y directions.

When the forward speed is higher than the rotor induced velocity, the fuselage drag can be modeled as the drag of a flat plate exposed to dynamic pressure. In this case, the perturbations to the fuselage forces can be expressed as

$$X_{\text{fus}} = S_x^{\text{fus}} \frac{1}{2} \rho U_e^2 \frac{u}{U_e}$$

$$Y_{\text{fus}} = S_y^{\text{fus}} \frac{1}{2} \rho U_e^2 \frac{v}{U_e}$$

where U_e is the trim airspeed.

Considering the above equations, fuselage forces can be approximated by

$$V_{\infty} = \sqrt{u_a^2 + v_a^2 + (w_a + V_{\text{imr}})^2}$$

$$X_{\text{fus}} = -0.5 \rho S_x^{\text{fus}} u_a V_{\infty}$$

$$Y_{\text{fus}} = -0.5\rho S_y^{\text{fus}} v_a V_\infty$$

$$Z_{\text{fus}} = -0.5\rho S_z^{\text{fus}} (w_a + V_{\text{imr}}) V_\infty$$

where S_x^{fus} , S_y^{fus} , and S_z^{fus} are effective frontal, side, and vertical drag areas of the fuselage and u_a , v_a , and w_a are fuselage center of pressure velocities with respect to air (i.e., $u_a = u - u_w$, where u_w is the projection of wind velocity vector on the X body axis). One can neglect small moments generated by the fuselage and assume that the fuselage center of pressure coincides with the helicopter center of gravity. Based on the fuselage projection areas, one can assume that $S_y^{\text{fus}} \approx 2.2S_x^{\text{fus}}$, $S_z^{\text{fus}} \approx 1.5S_x^{\text{fus}}$. Effective frontal drag area can be determined from the average pitch angle required to maintain a certain forward speed. This is best done under automatic control in velocity hold mode. In a steady trimmed flight, $\text{mg}\theta \approx -0.5\rho U^2 S_x^{\text{fus}}$. A pitch angle of -10° was required to maintain 14.5 m/s forward speed, which resulted in the estimate $S_x^{\text{fus}} = 0.1 \text{ m}^2$.

15.4.4 Vertical Fin Forces and Moments

The side force generated by the vertical fin can be approximated as follows:

$$Y_{\text{vf}} = -0.5\rho S_{\text{vf}} (C_{L_\alpha}^{\text{vf}} V_\infty^{\text{tr}} + |v_{\text{vf}}|) v_{\text{vf}} \quad (15.35)$$

where S_{vf} is the vertical fin area, $C_{L_\alpha}^{\text{vf}}$ is its lift curve slope, $V_\infty^{\text{tr}} = \sqrt{u_a \cdot u_a + w_{\text{tr}} \cdot w_{\text{tr}}}$ is the axial velocity at the location of the tail rotor hub, v_{vf} is the side velocity relative to air at the location of the vertical fin, and w_{tr} is the vertical velocity (same as for the tail rotor):

$$v_{\text{vf}} = v_a - \epsilon_{\text{vf}}^{\text{tr}} V_{\text{itr}} - l_{\text{tr}} r \quad (15.36)$$

$$w_{\text{tr}} = w_a + l_{\text{tr}} q - K_\lambda \cdot V_{\text{imr}} \quad (15.37)$$

Here V_{itr} is the induced velocity of the tail rotor (see Eq. (15.50)), r is yaw rate, $\epsilon_{\text{vf}}^{\text{tr}}$ is the fraction of the vertical fin area exposed to full induced velocity from the tail rotor, l_{tr} is the vertical distance between the c.g. and tail rotor hub, which is about the same distance to the center of pressure of the vertical fin, V_{imr} is the main rotor induced velocity, and K_λ is the wake intensity factor, calculated in the tail rotor section.

To accommodate for stall of the vertical fin (McConley 1998), the absolute value of the vertical fin side force is limited by

$$|Y_{\text{vf}}| \leq 0.5\rho S_{\text{vf}} \left((V_\infty^{\text{tr}})^2 + v_{\text{vf}}^2 \right) \quad (15.38)$$

The vertical fin side force creates a yawing moment and a small rolling moment due to the offsets from the c.g.:

$$\begin{aligned} N_{vf} &= -Y_{vf}l_{tr} \\ L_{vf} &= Y_{vf}h_{tr} \end{aligned}$$

15.4.5 Horizontal Stabilizer Forces and Moments

The destabilizing effect of the main rotor flapping due to vertical speed is offset by the weathervaning provided by the horizontal tailplane. The horizontal tail produces lift and a stabilizing pitching moment around the center of gravity. An effective vertical speed at the horizontal tail location is determined, assuming that the stabilizer may be fully or partially submerged in the downwash of the main rotor:

$$w_{ht} = w_a + l_{ht}q - K_\lambda \cdot V_{imr} \quad (15.39)$$

The same wake intensity factor is used for the horizontal fin as for the vertical fin and the tail rotor. Next, the Z-force generated by the horizontal stabilizer is determined according to

$$Z_{ht} = 0.5\rho S_{ht} (C_{L_\alpha}^{ht} |u_a| w_{ht} + |w_{ht}| w_{ht}) \quad (15.40)$$

where S_{ht} is the horizontal stabilizer area and $C_{L_\alpha}^{ht} = 3.0$ is its lift curve slope. To accommodate for the stall of the horizontal stabilizer (McConley 1998), the absolute value of the horizontal stabilizer lift is limited by

$$|Z_{ht}| \leq 0.5\rho S_{ht} (u_a^2 + w_{ht}^2) \quad (15.41)$$

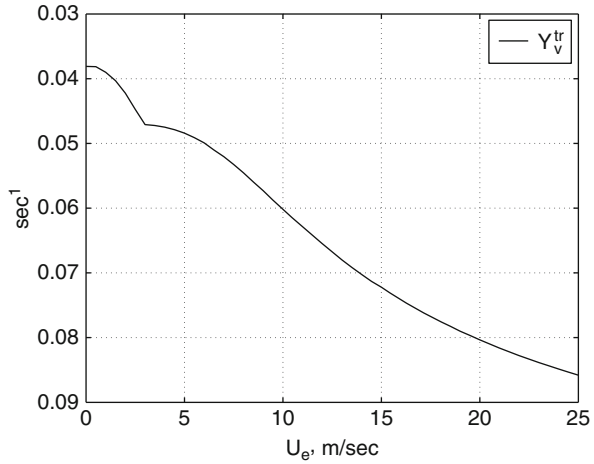
Finally, the pitching moment generated by the horizontal stabilizer is

$$M_{ht} = Z_{ht} \cdot l_{ht} \quad (15.42)$$

15.4.6 Tail Rotor

The tail rotor is subjected to a wide range of flow conditions, including those where the thrust-inflow iteration algorithm (given for the main rotor in Eqs. (15.4) and (15.5)) would fail (e.g., when the tail rotor operates in its own wake at a low in-plane airspeed). The thrust-inflow iteration equations for the tail rotor were linearized around the trim conditions corresponding to a no-sideslip flight at a range of forward speeds. The tail rotor thrust at such trim conditions is always nonzero to compensate for the main rotor torque, and zero sideslip implies that there is no airflow component normal to the rotor disk; therefore, Eqs. (15.4) and (15.5) are applicable:

Fig. 15.8 Linearized tail rotor side force due to side velocity



$$C_{T_{\mu_z}^{tr}} = \frac{\partial C_T^{tr}}{\partial \mu_z^{tr}} (|\mu_{tr}|, \mu_z^{tr} = 0, \delta_r^{trim}) \quad (15.43)$$

$$C_{T_{\delta_r}^{tr}} = \frac{\partial C_T^{tr}}{\partial \delta_r} (|\mu_{tr}|, \mu_z^{tr} = 0, \delta_r^{trim}) \quad (15.44)$$

The partial derivatives in Eqs. (15.43) and (15.44) were computed numerically. Simple approximate analytical expressions for the tail rotor coefficients can be obtained by adapting those used for the main rotor coefficients (Padfield 1996, pp. 219, 229). They fall within 15 % of those computed via numerical or exact analytical linearization. The resulting nondimensional coefficients were used to calculate the corresponding dimensional stability derivatives:

$$Y_v^{tr} = -C_{T_{\mu_z}^{tr}} \frac{f_t \rho \Omega_{tr} R_{tr} \pi R_{tr}^2}{m} \quad (15.45)$$

$$Y_{\delta_r}^{tr} = -C_{T_{\delta_r}^{tr}} \frac{f_t \rho (\Omega_{tr} R_{tr})^2 \pi R_{tr}^2}{m} \quad (15.46)$$

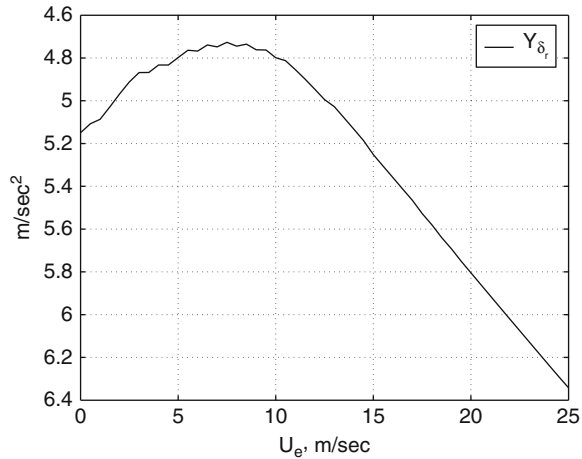
where f_t is the fin blockage factor, as suggested in Padfield (1996, p. 142):

$$f_t = 1.0 - \frac{3}{4} \frac{S_{vf}}{\pi R_{tr}^2}$$

The tail rotor speed is given by $\Omega_{tr} = n_{tr} \Omega_{mr}$, where n_{tr} is the gear ratio given in Table 15.1. For reference, the computed dimensional derivatives are provided in Figs. 15.8 and 15.9. Finally, the side force generated by the tail rotor is given in Eq. (15.47):

$$Y_{tr} = m Y_{\delta_r}^{tr} \delta_r + m Y_v^{tr} \mu_z^{tr} \Omega_{tr} R_{tr} \quad (15.47)$$

Fig. 15.9 Linearized tail rotor side force due to tail rotor pitch



In order to calculate Y_{tr} , one needs to determine the normal (μ_z^{tr}) and the in-plane (μ_{tr}) tail rotor inflow components. The main rotor wake affects the tail rotor thrust in a complex way; to model this influence accurately, an extensive modeling of the wake is required. It was decided to approximate just the increase in an apparent in-plane velocity seen by the tail rotor. For this, determine the main rotor wake intensity factor K_λ . The geometry calculations are equivalent to those given in Leishman (2000), but computationally more efficient since an explicit evaluation of the trigonometric functions is avoided. Calculate the following variables (tangents of the angles determining the geometry):

$$g_i = \frac{l_{tr} - R_{mr} - R_{tr}}{h_{tr}}$$

$$g_f = \frac{l_{tr} - R_{mr} + R_{tr}}{h_{tr}}$$

First, the tail rotor is out of the downwash if $V_{imr} \leq w_a$, in which case there is an effective upwash. Next, at low enough forward speed with respect to air, the tail rotor is out of the wake as well. This can be represented by the condition:

$$\frac{u_a}{V_{imr} - w_a} \leq g_i$$

In both of these cases, $K_\lambda = 0$. The tail rotor is fully in the wake if

$$\frac{u_a}{V_{imr} - w_a} \geq g_f \quad (15.48)$$

In the far wake, the downwash is twice the value at the rotor. It was assumed that $K_\lambda = 1.5$ when the tail rotor is fully immersed. In the remaining case, when the tail rotor is partially immersed, assume a linear growth of the wake intensity factor with the forward speed:

$$K_\lambda = 1.5 \frac{\frac{u_a}{V_{\text{mr}} - w_a} - g_i}{g_f - g_i} \quad (15.49)$$

The derived expression is used to calculate the vertical component of airspeed at the tail rotor location, as shown in Eq. (15.37). Next, determine an advance ratio for the tail rotor:

$$\mu_{\text{tr}} = \frac{u_a^2 + w_{\text{tr}}^2}{\Omega_{\text{tr}} \cdot R_{\text{tr}}}$$

Velocity component normal to the tail rotor is given by

$$v_{\text{tr}} = v_a - l_{\text{tr}} \cdot r + h_{\text{tr}} \cdot p$$

and in nondimensional form

$$\mu_{z_{\text{tr}}} = \frac{v_{\text{tr}}}{\Omega_{\text{tr}} \cdot R_{\text{tr}}}$$

The magnitude of the resulting tail rotor thrust is limited based on the assumed maximum thrust coefficient to model stall of the blades and other viscous losses:

$$Y_{\text{max}}^{\text{tr}} = f_t C_{T_{\text{max}}}^{\text{tr}} \rho (\Omega_{\text{tr}} R_{\text{tr}})^2 \pi R_{\text{tr}}^2$$

$$|Y_{\text{tr}}| \leq Y_{\text{max}}^{\text{tr}}$$

The yawing and small rolling moments due to offsets from the c.g. are computed as follows:

$$N_{\text{tr}} = -Y_{\text{tr}} l_{\text{tr}}$$

$$L_{\text{tr}} = Y_{\text{tr}} h_{\text{tr}}$$

The tail rotor induced velocity, used in the calculation of the vertical fin side force (see Eq. (15.36)), needs to be computed. Using the same derivation as for the main rotor (Padfield 1996, pp. 115–123), the inflow ratio is approximated by Eq. (15.50):

$$\lambda_0^{\text{tr}} = \mu_{z_{\text{tr}}} - 2 \left[\frac{2C_T^{\text{tr}}}{a_{\text{tr}} \sigma_{\text{tr}}} - \delta_r \left(\frac{1}{3} + \frac{\mu_{\text{tr}}^2}{2} \right) \right] \quad (15.50)$$

where C_T^{tr} is the computed tail rotor thrust coefficient, a_{tr} is the tail rotor blade lift curve slope given in Table 15.1, and $\sigma_{\text{tr}} = \frac{2C_{\text{tr}}}{\pi R_{\text{tr}}}$ is the tail rotor solidity ratio.

Finally, the tail rotor torque Q_{tr} is computed similarly to Eqs. (15.11) and (15.12) using the tail rotor parameters in place of the main rotor parameters.

15.5 Actuator Models

The command ranges for the cyclics and collective blade pitch are symmetric around the center point. The tail rotor blade pitch is offset by the trim value given in Table 15.1 such that the tail rotor pitch is computed as

$$\delta_r = \delta_r^{\text{cmd}} + \delta_r^{\text{trim}} \quad (15.51)$$

The following maximum commanded deflections were set, in radians:

$$\delta_{\text{lat}}^{\text{max}} = 0.096$$

$$\delta_{\text{lon}}^{\text{max}} = 0.096$$

$$\delta_{\text{col}}^{\text{max}} = 0.183$$

$$\delta_{r_{\text{cmd}}}^{\text{max}} = 0.38$$

Here $\delta_{\text{lat}}^{\text{max}}$ and $\delta_{\text{lon}}^{\text{max}}$ are actual maximum cyclic pitch angles of the main rotor blades, measured statically. The gearing between servos and pitch angles of control surfaces is close to being linear. Linear functions are used to relate servo pulse-width commands to control surface deflections. Lookup tables can be used for a different gearing. Hobby servos and pulse-width generation electronics used on the helicopter result in significant quantization effects. On average, 150 steps were used to encode rail-to-rail deflection of each control surface. This results, for example, in a tail rotor command quantization of 0.3° . Linear transfer functions are used to model the servo dynamics. Futaba S9402 servos, used for collective and cyclic deflections of the main rotor blades, were subjected to small-amplitude frequency sweeps under 35 oz-in mean load and small inertia, which was assumed to be representative of the actual loads experienced by the servos during the flight. The following transfer function came up as a result:

$$H_{\text{servo}}(s) = \frac{s/T_z + 1}{s/T_p + 1} \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $T_z = 104$ s, $T_p = 33$ s, $\omega_n = 36$ rad/s, and $\zeta = 0.5$. Note that 90° phase lag occurs at roughly 30 rad/s, which also imposes a limitation on the control system bandwidth. A fast digital servo (Futaba S9450) was used for the tail rotor pitch. Since the torque required from the tail rotor servo is much lower than that required from the swashplate actuators, no-load small-signal bandwidth tests provide an

adequate model of the servo. As a result of the tests, the servo transfer function was approximated by a second-order system with the undamped natural frequency of 7 Hz and the damping ratio of 0.6.

Conclusion

A representative nonlinear dynamic model of a miniature aerobatic helicopter was developed with first-principles methods. In addition to the rigid body states, the model includes two states to represent lateral and longitudinal flapping angles of the main rotor, one state for the main rotor speed, and one state for the integral of the main rotor speed tracking error. Flight-test data were used for determining several key parameters. The model is valid up to advance ratios $\mu < 0.15$ for a variety of maneuvering flight conditions, including negative rotor loading and high angular rates. The model can be used for developing and evaluating control design methods for demanding tasks, including aerobatics. This simplified modeling framework is suitable for a class of miniature helicopters with hingeless rotors.

References

- P. Abbeel, A. Coates, A.Y. Ng, Autonomous auto-rotation of an rc helicopter, in *Proceedings of ISER*, Athens, Greece, 2008
- P. Abbeel, A. Coates, A.Y. Ng, Autonomous helicopter aerobatics through apprenticeship learning. *Int. J. Robot. Res.* **29**, 1608–1639 (2010)
- P. Abbeel, V. Ganapathi, A.Y. Ng, Learning vehicular dynamics with application to modeling helicopters, in *Proceedings of NIPS 18*, Vancouver, British Columbia, Canada, 2006
- A.R.S. Bramwell, *Bramwell's Helicopter Dynamics* (AIAA, Reston, 2001)
- R.T. Chen, A simplified rotor system mathematical model for piloted flight dynamics simulation. Technical Memorandum 78575, NASA, 1979
- E. Feron, M. Brenner, J. Paduano, A. Turevskiy, Time-frequency analysis for the transfer function estimation and application to flutter clearance. *AIAA J. Guid. Control Dyn.* **21**(3), 375–382 (1998)
- V. Gavrillets, Autonomous aerobatic maneuvering of miniature helicopters. Ph.D. dissertation, Massachusetts Institute of Technology, 2003
- V. Gavrillets, M. Martinos, B. Mettler, E. Feron, Control logic for automated aerobatic flight of miniature helicopter, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug 2002
- F.D. Harris, Articulated rotor blade flapping motion at low advance ratio. *J. Am. Helicopter Soc.* **17**, 41–48 (1972)
- C. Harris (ed.), *Shock and Vibration Handbook* (McGraw-Hill, New York, 1996)
- A.M. Kuethe, C.Y. Chow, *Foundations of Aerodynamics* (Wiley, New York, 1986)
- M. LaCivita, T. Kanade, G. Papageorgiu, W. Messner, Design and flight testing of a high-bandwidth h-infinity loop shaping controller for a robotic helicopter, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug 2002a
- M. LaCivita, W. Messner, T. Kanade, Modeling of small-scale helicopters with integrated first-principles and integrated system identification techniques, in *Presented at 58th Forum of American Helicopter Society*, Montreal, Canada, June 2002b
- J.G. Leishman, *Principles of Helicopter Aerodynamics* (Cambridge University Press, New York, 2000)
- L. Ljung, *System Identification: Theory for the User* (Prentice Hall, Upper Saddle River, 1999)

- M. McConley, Draper small autonomous aerial vehicle dynamic model. Technical report E41-98-091, Draper Laboratory, Aug 1998
- B. Mettler, *Identification, Modeling and Characteristics of Miniature Rotorcraft* (Kluwer, Boston, 2002)
- B. Mettler, M. Tischler, T. Kanade, W. Messner, Attitude control optimization for a small-scale unmanned helicopter, in *AIAA Guidance, Navigation and Control Conference*, Denver, CO, Aug 2000
- B. Mettler, V. Gavrillets, E. Feron, T. Kanade, Dynamic compensation for high-bandwidth control of small-scale helicopter, in *American Helicopter Society Specialist Meeting*, San Francisco, CA, Jan 2002a
- B. Mettler, M.B. Tischler, T. Kanade, System identification modeling of a small-scale unmanned rotorcraft for control design. *J. Am. Helicopter Soc.* **47**(1), 50–63 (2002b)
- Miniature Aircraft USA, *X-Cell .60 graphite SE Helicopter Kit (Special Edition) Instruction Manual* (Miniature Aircraft USA, Orlando, 1999)
- NASA Ames Research Center, *Comprehensive Identification from Frequency Responses: An Interactive Facility for System Identification and Verification* (NASA Ames Research Center, Moffet Field, 2000)
- G.D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. AIAA Education Series (AIAA, Reston, 1996)
- J.M. Rolfe, K.J. Staples, *Flight Simulation*. (Cambridge University Press, Cambridge, 1986)
- K. Sprague, V. Gavrillets, D. Dugail, B. Mettler, E. Feron, Design and applications of an avionics system for a miniature acrobatic helicopter, in *AIAA Digital Avionics Systems Conference*, Daytona Beach, FL, 2001
- T.D. Talbot, B.E. Tingling, W.A. Decker, R.T. Chen, A mathematical model of a single main rotor helicopter for piloted simulation. Technical Memorandum 84281, NASA, 1982
- M. Tischler (ed.), *Advances in Aircraft Flight Control* (Taylor and Francis, Cornwall, 1996)
- W. von Grunhagen, G. Bouwer, H.-J. Pausder, F. Henchel, J. Kaletka, A high bandwidth control system for the helicopter in-flight simulator at the – modelling, performance and applications, in *Advances in Aircraft Flight Control*, ed. by M. Tischler (Taylor and Francis, Cornwall, 1996)

Caitlin Powers, Daniel Mellinger, and Vijay Kumar

Contents

| | | |
|--------|--|-----|
| 16.1 | Introduction | 308 |
| 16.2 | Modeling | 309 |
| 16.2.1 | Quadrotor Platforms..... | 309 |
| 16.2.2 | Coordinate System and Reference Frames | 310 |
| 16.2.3 | Inertial Properties..... | 311 |
| 16.2.4 | Motor Model..... | 311 |
| 16.2.5 | Rigid Body Dynamics..... | 313 |
| 16.2.6 | Differential Flatness | 314 |
| 16.3 | Robot Controllers | 317 |
| 16.3.1 | Linearized Controller..... | 317 |
| 16.3.2 | Nonlinear Control | 322 |
| 16.4 | Experimental Results | 323 |
| 16.4.1 | Vicon Motion Capture System | 324 |
| 16.4.2 | Software and Integration | 324 |
| 16.4.3 | Results | 325 |
| 16.5 | Conclusion | 327 |
| | References | 327 |

Abstract

This chapter presents an overview of the rigid body dynamics of a quadrotor as well as several controllers for the quadrotor. First, the Newton-Euler equations of motion that govern the quadrotor motion are described, and it is shown that the quadrotor model is differentially flat. Next, two controllers for the quadrotor

C. Powers (✉) • V. Kumar

Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania,
Philadelphia, PA, USA

e-mail: cpow@seas.upenn.edu; kumar@seas.upenn.edu

D. Mellinger

KMel Robotics, Philadelphia, PA, USA

e-mail: daniel@kmelrobotics.com

are presented. The first is a linear controller based on a linearized model of the dynamics. The second is a nonlinear controller derived from the original dynamic model. The architecture of the GRASP quadrotor testbed at the University of Pennsylvania is also presented. Finally, experimental results which illustrate the dynamics and control of small quadrotors are presented.

16.1 Introduction

Aerial robotics is a growing field with tremendous civil and military applications. Potential applications for micro unmanned aerial vehicles include search and rescue tasks, environmental monitoring, aerial transportation and manipulation, and surveillance. Quadrotor designs, rotorcrafts whose propulsive force is provided by four rotors, make flexible and adaptable platforms for aerial robotics. Rotorcraft designs are not limited to quadrotors – vehicle designs range from helicopters with tail rotors as seen in vehicles used for transport, coaxial helicopter designs such as the Skybotix CoaX (Skybotix Technologies 2012) to the Falcon 8, a vehicle with 8 rotors sold by Ascending Technologies, GmbH (Ascending Technologies 2012). However, quadrotors have recently emerged as the platform of choice for research in micro aerial vehicles (Kumar and Michael 2011). They are highly maneuverable and enable safe and low-cost experimentation in mapping, navigation, and control algorithms in three dimensions. They can hover in place and take off and land vertically, unlike their fixed-wing counterparts. They also have sufficient payload and flight endurance to support a number of indoor and outdoor applications. Their ability to move in 3-dimensional space brings new research challenges compared to the wheeled mobile robots that have driven mobile robotics research over the last decade. In recent years, small quadrotors have been demonstrated for mapping three-dimensional environments (Shen et al. 2011), transporting and manipulating payloads (Lindsey et al. 2011), assembling structures (Michael et al. 2011), and for autonomous exploration (Shen et al. 2012).

The work in this chapter deals specifically with quadrotors that are on the order of 0.1–0.5 m in length and 0.1–0.5 kg in mass. As the length is scaled down, mass and inertia scale down much faster resulting in higher angular accelerations and superior maneuverability. As shown in Kushleyev et al. (2012), very small quadrotors (0.1 m in diameter) are remarkably agile. Another advantage is that the quadrotor design is easy to build and is quite robust. The blades have fixed pitch and the propellers rotate in one direction, allowing for simple motors and controllers. There are no hinges or flaps. As a result quadrotors are also easy to model and control.

However, quadrotors are inefficient compared to fixed-wing UAVs. While fixed-wing UAVs are able to exploit the lift generated by the structure of their wings, rotorcraft such as quadrotors must produce all of the energy they need to fly via their rotors, which is a much less efficient process. Because of limits on battery energy density and specific power, the battery represents close to 30 % of total mass, yielding flight times of less than 30 min. Another key challenge to creating autonomous quadrotors is the need to equip them with onboard sensors

and processors to enable 3-D state estimation. Sensors such as lidars are heavy and consume a lot of power, while cameras require significant processing power. The reader is referred to Kumar and Michael (2011) for a discussion of the challenges in developing autonomous quadrotors.

This chapter presents an overview of the dynamics, control, and planning for quadrotors. The next section describes a dynamic model of a quadrotor. The dynamic model is used to develop controllers for the system, which are described in Sect. 16.3. Experimental results with controllers on real platforms are discussed in Sect. 16.4. Two popular quadrotor platforms are used extensively for modeling and experimental results. The first is the Hummingbird quadrotor sold by Ascending Technologies, GmbH (Ascending Technologies 2012). The second platform is the Nano Quadrotor, a micro quadrotor designed by KMeI Robotics (KMeI Robotics 2011; Kushleyev et al. 2012). Both platforms will be described in detail throughout this chapter.

16.2 Modeling

16.2.1 Quadrotor Platforms

This section starts by describing two representative quadrotor platforms. Both platforms have onboard microcontrollers which allow low-level control and estimation to be done in flight by the robot. In addition, both platforms have onboard IMUs which provide feedback of accelerations and angular velocities. The Hummingbird quadrotor is shown in Fig. 16.1. The Hummingbird spans 55 cm from the tip of one rotor to the tip of the opposite rotor, has a height of 8 cm, and has a mass of about 500 g including a battery. The platform is durable enough to survive most crashes, while the blades are soft enough not to cause damage during such event. Furthermore, the 20 min battery life and 200 g payload capacity are also advantageous.



Fig. 16.1 A quadrotor platform with Vicon markers

The Nano Quadrotor spans 15 cm, has a height of 5 cm, and has a mass of about 75 g including battery. The platform is very small and maneuverable and safe due to the small size of the rotors.

16.2.2 Coordinate System and Reference Frames

A convenient description of the kinematics of a quadrotor can be provided using four reference frames. The inertial frame, \mathcal{A} , is defined by the triad \mathbf{a}_1 , \mathbf{a}_2 , and \mathbf{a}_3 with \mathbf{a}_3 pointing upward. Two intermediate frames \mathcal{E} and \mathcal{F} are produced by first rotating through an angle ψ about the \mathbf{a}_3 axis and then rotating about an angle ϕ about the \mathbf{e}_1 axis. Finally, the body frame \mathcal{B} of the quadrotor is defined by rotating about an angle θ about the \mathbf{f}_2 axis. The origin of the body frame is attached to the center of mass of the quadrotor with \mathbf{b}_3 perpendicular to the plane of the rotors pointing vertically up (aligned with \mathbf{a}_3) during perfect hover. The basis vectors of the body frame are parallel to the principal axes of inertia of the quadrotor. The center of mass is denoted as C (Fig. 16.2).

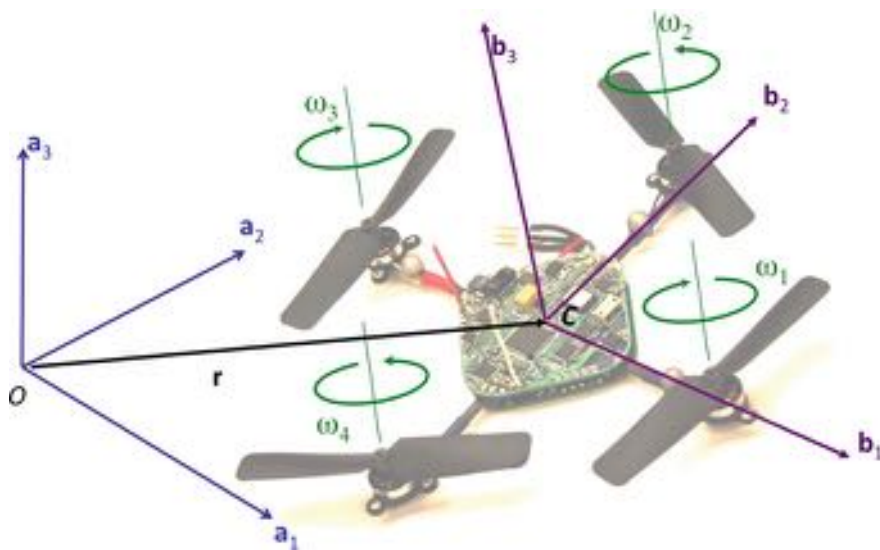


Fig. 16.2 The body-fixed frame and the inertial reference frame. A pair of motors (1 and 3) spins counterclockwise, while the other pair spins clockwise (2 and 4). The pitches on the corresponding propellers are reversed so that the thrust is always pointing in the \mathbf{b}_3 direction for all propellers. However, while the reaction moments on the frame of the robot are also in the vertical direction, the signs are such that they oppose the direction of the angular velocity of the propeller

16.2.3 Inertial Properties

Since \mathbf{b}_i are principal axes, the inertia matrix referenced to the center of mass along the \mathbf{b}_i reference triad, I , is a diagonal matrix. In practice, the three moments of inertia can be estimated by weighting individual components of the quadrotor and building a physically accurate model in SolidWorks or approximated using known inertia tensors of simple shapes (e.g., cylinders for the motors). The key parameters for the rigid body dynamics of the quadrotors used for simulation and experiments in this text are as follows:

- (a) Total mass of the quadrotor m
- (b) The distance from the center of mass to the axis of a motor: L
- (c) The components of the inertia dyadic using \mathbf{b}_i as the basis vectors:

$$[I_C]^{\mathbf{b}_i} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}.$$

The inertial properties of each of the quadrotor platforms are detailed in Table 16.1. Inspection of this table combined with reference to Kumar and Michael (2011) allows the reader to gain some insight into quadrotor scaling. Let L represent the characteristic length. The rotor radius R should scale linearly with L , the mass should as L^3 , and the moments of inertia should scale as L^5 . The force produced by the rotors, F , and the drag force D scale with the cross-sectional area and the square of the blade tip speed, v . Let the angular speed of the blades be denoted ω . Clearly, $\omega = \frac{v}{L}$. Thus $F \sim \omega^2 L^4$ and $D \sim \omega^2 L^4$. Linear acceleration $a = \frac{F}{m}$ scales as $a \sim \omega^2 L$.

16.2.4 Motor Model

Each rotor has an angular speed ω_i and produces a vertical force F_i according to

$$F_i = k_F \omega_i^2. \quad (16.1)$$

Table 16.1 Inertial properties of quadrotor platforms

| Platform | Hummingbird | Nano |
|--|-----------------------|-----------------------|
| $I_{xx}, \frac{\text{kg}}{\text{m}^2}$ | 2.32×10^{-3} | 4.28×10^{-5} |
| $I_{yy}, \frac{\text{kg}}{\text{m}^2}$ | 2.32×10^{-3} | 4.28×10^{-5} |
| $I_{zz}, \frac{\text{kg}}{\text{m}^2}$ | 4.00×10^{-3} | 8.36×10^{-5} |
| m, kg | 0.5 | 0.075 |
| L, m | 0.175 | 0.0635 |

The rotors also produce a moment according to

$$M_i = k_M \omega_i^2. \quad (16.2)$$

The constants k_M and k_F can be determined by experimentation with fixed rotors or by matching the performance of a simulation to the performance of the real system.

Data obtained from system identification experiments suggests that the rotor speed is related to the commanded speed by a first-order differential equation

$$\dot{\omega}_i = k_m(\omega_i^{\text{des}} - \omega_i).$$

This motor gain, k_m , is found from experimental data. The desired angular velocities, ω_i^{des} , are limited to a minimum and maximum value determined through experimentation. The response of a motor from the Nano Quadrotor to a step input is shown in Fig. 16.3.

However, as a first approximation, the motor controllers can be assumed to be perfect and the time constant k_m associated with the motor response to be arbitrarily small. In other words, assume that the actual motor velocities ω_i are equal to the commanded motor velocities, ω_i^{des} . Specific values for the Hummingbird and Nano Quadrotor platforms are given in Table 16.2.

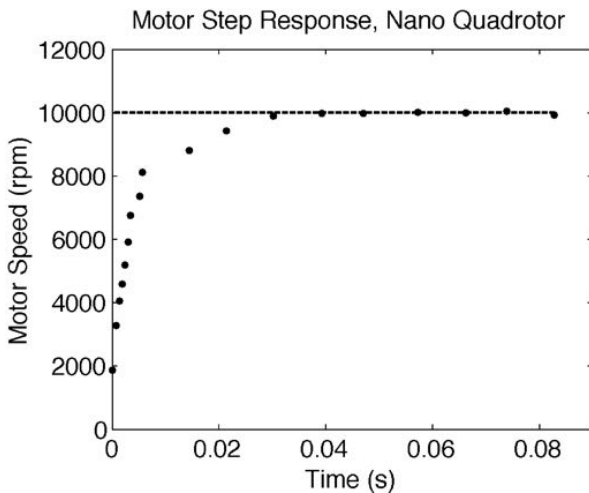


Fig. 16.3 Experimental step response for motor on Nano Quadrotor. The commanded speed is a step from 0 to 10,000 RPM at time 0. Dots represent measured motor speeds. The motor reaches the desired speed in approximately 0.25 s

Table 16.2 Motor characteristics of quadrotor platforms

| Platform | Hummingbird | Nano Quadrotor |
|---------------------------------------|-----------------------|-------------------------|
| $k_M, \frac{\text{Nm}}{\text{rpm}^2}$ | 1.5×10^{-9} | 4.737×10^{-12} |
| $k_F, \frac{\text{N}}{\text{rpm}^2}$ | 6.11×10^{-8} | 1.9985×10^{-9} |
| k_m, s^{-1} | 20 | 200 |

16.2.5 Rigid Body Dynamics

Kinematics The $Z - X - Y$ Euler angle convention is used to model the rotation of the quadrotor in the world frame. As described in Sect. 16.2.2 to get from \mathcal{A} to \mathcal{B} , first rotate about \mathbf{a}_3 through the yaw angle, ψ , to get the triad \mathbf{e}_i . A rotation about the \mathbf{e}_1 through the roll angle, ϕ results in the triad \mathbf{f}_i (not shown in the figure). A third pitch rotation about \mathbf{f}_2 through θ results in the body-fixed triad \mathbf{b}_i . This convention produces the rotation matrix from \mathcal{A} to \mathcal{B} , ${}^{\mathcal{A}}R_{\mathcal{B}}$:

$${}^{\mathcal{A}}[R]_{\mathcal{B}} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}. \quad (16.3)$$

The components of angular velocity of the robot in the body frame are given by p , q , and r :

$${}^{\mathcal{A}}\omega_{\mathcal{B}} = p\mathbf{b}_1 + q\mathbf{b}_2 + r\mathbf{b}_3.$$

These values are related to the derivatives of the roll, pitch, and yaw angles according to

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad (16.4)$$

Newton's Equations of Motion Let \mathbf{r} denote the position vector of C in \mathcal{A} . The forces on the system are gravity, in the $-\mathbf{a}_3$ direction, and the forces from each of the rotors, F_i , in the \mathbf{b}_3 direction. The equations governing the acceleration of the center of mass are

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + {}^{\mathcal{A}}R_{\mathcal{B}} \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}. \quad (16.5)$$

The first input u_1 will be defined to be

$$u_1 = \sum_{i=1}^4 F_i,$$

which is the total force produced by the rotors in the \mathbf{b}_3 direction.

Euler's Equations of Motion In addition to forces, each rotor produces a moment perpendicular to the plane of rotation of the blade, M_i . Rotors 1 and 3 rotate in the $-\mathbf{b}_3$ direction, while 2 and 4 rotate in the $+\mathbf{b}_3$ direction. Since the moment produced on the quadrotor is opposite to the direction of rotation of the blades, M_1 and M_3 act in the \mathbf{b}_3 direction while, M_2 and M_4 act in the $-\mathbf{b}_3$ direction.

The angular acceleration determined by the Euler equations is

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (16.6)$$

This can be rewritten as

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (16.7)$$

where $\gamma = \frac{k_M}{k_F}$ is the relationship between lift and drag given by Eqs. (16.1) and (16.2). Accordingly, the second set of inputs is defined as the vector of moments \mathbf{u}_2 given by

$$\mathbf{u}_2 = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}.$$

Since u_1 and \mathbf{u}_2 are defined using a linear combination of the forces produced by the motors, they become the inputs to the system. Note that a rigid body in cartesian coordinates has six degrees of freedom, so the four inputs result in an underactuated system.

16.2.6 Differential Flatness

The quadrotor has six degrees of freedom in position and orientation but only four actuators. This means that it is an underactuated system. In order to plan dynamically feasible trajectories and feedforward control inputs, it must be shown that the quadrotor system is differentially flat. A system is differentially flat if the states and inputs can be written as algebraic functions of the flat outputs σ and their derivatives (van Nieuwstadt et al. 1994). This means that any trajectory $\sigma(t)$ in the space of flat outputs will be dynamically feasible for the quadrotor. In the case of the quadrotor, the four flat outputs can be shown to be $\sigma = [x, y, z, \psi]$.

The quadrotor is differentially flat if it can be shown that there exists a smooth map

$$(\mathbf{x}, \mathbf{u}) = \Phi(\sigma, \dot{\sigma}, \ddot{\sigma}), \quad (16.8)$$

where \mathbf{x} is the state of the quadrotor, defined as $\mathbf{x} = [x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]$, and \mathbf{u} are the control inputs as defined in Sect. 16.2.5. From the above definition of σ , it is trivial to show that this condition is satisfied for position and velocity, with the acceleration proving that the map is smooth:

$$\begin{aligned} [x, y, z] &= [\sigma_1, \sigma_2, \sigma_3] \\ [\dot{x}, \dot{y}, \dot{z}] &= [\dot{\sigma}_1, \dot{\sigma}_2, \dot{\sigma}_3] \\ [\ddot{x}, \ddot{y}, \ddot{z}] &= [\ddot{\sigma}_1, \ddot{\sigma}_2, \ddot{\sigma}_3]. \end{aligned} \quad (16.9)$$

Next, the condition must be satisfied for rotation and angular velocity as well. Let the rotation matrix ${}^A R_B$ describe the rotation of the quadrotor, with the unit vectors in the body frame denoted by $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$. From the equations of motion,

$$\mathbf{b}_3 = \frac{\frac{1}{u_1}(\ddot{\mathbf{r}} - m\mathbf{g})}{|\frac{1}{u_1}(\ddot{\mathbf{r}} - m\mathbf{g})|} = \frac{(\ddot{\mathbf{r}} - m\mathbf{g})}{|\ddot{\mathbf{r}} - m\mathbf{g}|} \quad (16.10)$$

where the dependence on u_1 can be eliminated since it is a scalar and \mathbf{b}_3 is a unit vector. It has been shown above that $\ddot{\mathbf{r}}$ is a function of the flat outputs alone and the remaining terms are constant, so \mathbf{b}_3 is also a function of the flat outputs. A second unit vector in the body frame can be obtained using the following:

$$\mathbf{e}_1 = \cos \psi \mathbf{a}_1 + \sin \psi \mathbf{a}_2 \quad (16.11)$$

$$\mathbf{b}_3 \times \mathbf{e}_1 = \mathbf{b}_3 \times R' \mathbf{b}_1 = \mathbf{b}_3 \times (\cos \theta \mathbf{b}_1 - \sin \theta \mathbf{b}_3) = \cos \theta \mathbf{b}_2 \quad (16.12)$$

where R^T is a rotation of $-\theta$ about the \mathbf{b}_2 axis. Therefore the result of $\mathbf{b}_3 \times \mathbf{e}_1$ can be normalized to find \mathbf{b}_2 as a function of the flat outputs, provided that $\cos \theta$ is not zero. Finally, the cross product of \mathbf{b}_3 and \mathbf{b}_2 can be used to find \mathbf{b}_1 .

To show that the angular velocity is also a function of the flat outputs, use the derivative of acceleration given by the Newton-Euler equations in (16.5):

$$\mathbf{j} = \frac{1}{m} \dot{u}_1 \mathbf{b}_3 + \frac{1}{m} u_1 {}^A \omega^B \times \mathbf{b}_3. \quad (16.13)$$

Define $\mathbf{j} = \ddot{\mathbf{r}}$ to denote the jerk. Taking the dot product of the above equation with \mathbf{b}_3 shows that

$$\dot{u}_1 = m \mathbf{j} \cdot \mathbf{b}_3. \quad (16.14)$$

This can be substituted into (16.13) to find

$${}^A \omega^B \times \mathbf{b}_3 = \frac{m}{u_1} (\mathbf{j} - (\mathbf{j} \cdot \mathbf{b}_3) \mathbf{b}_3). \quad (16.15)$$

Expressing ${}^A \omega^B$ in terms of components p, q , and r means that the left-hand side of the equations above can be equated to $q \mathbf{b}_1 - p \mathbf{b}_2$ to find

$$p = -\mathbf{b}_2 \cdot \frac{m}{u_1}(\mathbf{j} - (\mathbf{j} \cdot \mathbf{b}_3)\mathbf{b}_3), q = \mathbf{b}_1 \cdot \frac{m}{u_1}(\mathbf{j} - (\mathbf{j} \cdot \mathbf{b}_3)\mathbf{b}_3). \quad (16.16)$$

For the third component of angular velocity, the derivatives of the Euler angles are related to the angular velocity using the following,

$${}^A\omega^B = [\mathbf{e}_1 \ \mathbf{b}_2 \ \mathbf{a}_3] \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = {}^A R_B \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (16.17)$$

This results in a system of three equations and three unknowns ($r, \dot{\phi}, \dot{\theta}$) which can be solved provided that \mathbf{e}_1 , \mathbf{b}_2 , and \mathbf{a}_3 are linearly independent.

The mapping between the flat outputs and the derivatives of angular velocity is found by taking another derivative of Newton's equation, giving

$$m\dot{\mathbf{j}} = \ddot{u}_1\mathbf{b}_3 + 2{}^A\omega^B \times \dot{u}_1\mathbf{b}_3 + {}^A\omega^B \times {}^A\omega^B \times u_1\mathbf{b}_3 + {}^A\alpha^B \times u_1\mathbf{b}_3. \quad (16.18)$$

\ddot{u}_1 can be found by projecting this equation along \mathbf{b}_3 . ${}^A\omega^B$, u_1 and \dot{u}_1 can be used to compute ${}^A\alpha^B \times \mathbf{b}_3$. This produces

$$\dot{p} = -({}^A\alpha^B \times \mathbf{b}_3) \cdot \mathbf{b}_2, \quad \dot{q} = ({}^A\alpha^B \times \mathbf{b}_3) \cdot \mathbf{b}_1. \quad (16.19)$$

For the last component of angular acceleration, the derivative of the relationship between Euler angles and angular velocity (16.17) can be used:

$${}^A R_B \left(\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) = {}^\varepsilon\omega^A \times \dot{\phi}\mathbf{e}_1 + {}^B\omega^A \times \dot{\theta}\mathbf{b}_2 + [\mathbf{e}_1 \ \mathbf{b}_2 \ \mathbf{a}_3] \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}. \quad (16.20)$$

If \mathbf{e}_1 , \mathbf{b}_2 , and \mathbf{a}_3 are linearly independent, then this equation is affine in the variables \dot{p} , \dot{q} and \dot{r} and results in a system of equations which can be solved for \dot{r} .

Finally the control inputs need to be expressed as a function of the flat outputs. This can be done by simply rearranging the Newton-Euler equations to solve for u_1 and \mathbf{u}_2 :

$$u_1 = m || [\ddot{\sigma}_1 \ \ddot{\sigma}_2 \ \ddot{\sigma}_3 + g] || \quad (16.21)$$

$$\mathbf{u}_2 = \mathcal{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathcal{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (16.22)$$

Thus, the existence of the smooth map is shown in (16.9), (16.16), (16.17), and (16.19)–(16.22). However, as presented, the derivation is only valid provided that \mathbf{z}_B is never parallel to \mathbf{x}_C and \mathbf{y}_B is never parallel to \mathbf{z}_W .

16.3 Robot Controllers

16.3.1 Linearized Controller

For control at small angles, the position and attitude control of the quadrotor can be decoupled. This allows the use of nested feedback loops which allows the attitude control to run at a much faster rate than the position control.

The control problem is to determine the four inputs $\{u_1, \mathbf{u}_2\}$ required to follow a desired trajectory \mathbf{z}^{des} . As shown in Fig. 16.4, errors in the robot's position will drive a position controller which directly determines u_1 from (16.26). The model in (16.26) also leads to the desired orientation along the trajectory. The attitude controller for this orientation is derived from the model in (16.7).

The transformation of the forces F_1 to F_4 into $\{u_1, \mathbf{u}_2\}$ is straightforward and is described in Michael et al. (2010). The inner attitude control loop uses onboard accelerometers and gyroscopes to control the roll, pitch, and yaw and runs at approximately 1 kHz (Gurdan et al. 2007), while the outer position control loop uses estimates of position and velocity of the center of mass to control the trajectory in three dimensions at 100–200 Hz.

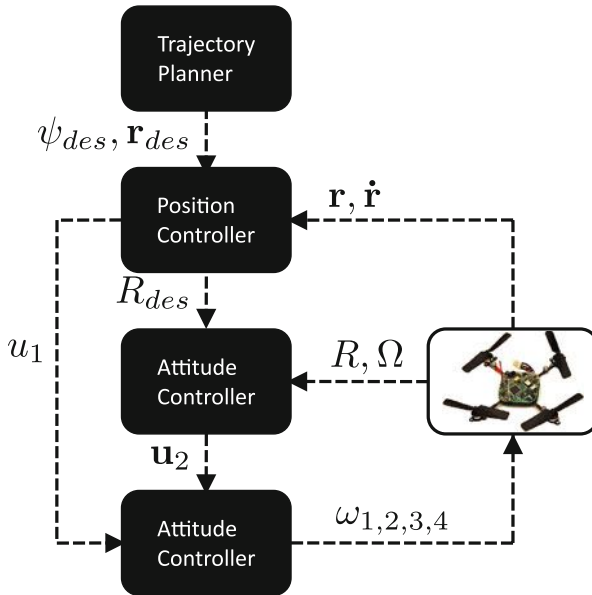


Fig. 16.4 The position and attitude control loops

16.3.1.1 Linearization

The controllers are derived by linearizing the equations of motion and motor model Eqs. (16.2)–(16.5) at an operating point that corresponds to the nominal hover state, \mathbf{x}_0 , with the inputs nominally required for hover, \mathbf{u}_0 . The hover configuration is given by $\mathbf{r} = \mathbf{r}_0$, $\theta = \phi = 0$, $\psi = \psi_0$, $\dot{\mathbf{r}} = 0$, and $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0$. This approximation is valid where the roll and pitch angles are small ($c\phi \approx 1$, $c\theta \approx 1$, $s\phi \approx \phi$, and $s\theta \approx \theta$). Linearizing the equations of motion about the hover state results in an equation of the form

$$\dot{\mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{u})|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x} - \mathbf{x}_0) + \frac{\partial}{\partial \mathbf{u}} f(\mathbf{x}, \mathbf{u})|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{u} - \mathbf{u}_0) = A\mathbf{x} + B\mathbf{u}.$$

At this state the lift from the propellers is given by

$$F_{i,0} = \frac{mg}{4},$$

The nominal values for the inputs at hover are $u_{1,0} = mg$, $\mathbf{u}_{2,0} = \mathbf{0}$. A and B are given by

$$A = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 6} \\ 0_{3 \times 6} & a_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 6} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 6} & 0_{3 \times 3} \end{bmatrix} \quad (16.23)$$

$$a = \begin{bmatrix} g \cos \psi_0 & g \sin \psi_0 & 0 \\ g \sin \psi_0 & -g \cos \psi_0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (16.24)$$

$$B = \begin{bmatrix} 0_{4 \times 5} & & & \\ \frac{1}{m} & 0 & 0 & 0 \\ & 0_{4 \times 3} & & \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}. \quad (16.25)$$

Specifically, the linearized equations for the acceleration are

$$\ddot{r}_1 = g(\Delta\theta \cos \psi_0 + \Delta\phi \sin \psi_0) \quad (16.26)$$

$$\ddot{r}_2 = g(\Delta\theta \sin \psi_0 - \Delta\phi \cos \psi_0)$$

$$\ddot{r}_3 = \Delta u_1 = u_1 - u_{1,0}$$

$$\Delta\theta = \theta - \theta_0$$

$$\Delta\phi = \phi - \phi_0.$$

Assuming that the rotor craft is symmetric so $I_{xx} = I_{yy}$,

$$\begin{aligned}\dot{p} &= \frac{u_{2,x}}{I_{xx}} = \frac{L}{I_{xx}}(F_2 - F_4) \\ \dot{q} &= \frac{u_{2,y}}{I_{yy}} = \frac{L}{I_{yy}}(F_3 - F_1), \\ \dot{r} &= \frac{u_{2,z}}{I_{zz}} = \frac{\gamma}{I_{zz}}(F_1 - F_2 + F_3 - F_4).\end{aligned}$$

In other words, the equations of motion for the angular accelerations are decoupled. Each component of angular acceleration depends only on the appropriate component of \mathbf{u}_2 .

16.3.1.2 Attitude Control

One attitude controller is a proportional plus derivative (PD) attitude controller which tracks a trajectory in $SO(3)$ specified in terms of a desired roll, pitch, and yaw angle. Since the development of the controller will be based on linearized equations of motion, the attitude must be close to the nominal hover state where the roll and pitch angles are small.

Near the nominal hover state the proportional plus derivative control laws take the form:

$$\mathbf{u}_2 = \begin{bmatrix} k_{p,\phi}(\phi^{\text{des}} - \phi) + k_{d,\phi}(p^{\text{des}} - p) \\ k_{p,\theta}(\theta^{\text{des}} - \theta) + k_{d,\theta}(q^{\text{des}} - q) \\ k_{p,\psi}(\psi^{\text{des}} - \psi) + k_{d,\psi}(r^{\text{des}} - r) \end{bmatrix}. \quad (16.27)$$

16.3.1.3 Position Control

In the next subsection, the dynamic model will be used to derive a position control method that uses the roll and pitch angles as inputs to drive the position of the quadrotor. The position control algorithm will determine the desired roll and pitch angles, θ^{des} and ϕ^{des} , which can be used to compute the desired speeds from (16.27). The controller tracks a specified trajectory, $\mathbf{r}_T(t)$, in three dimensions. The yaw angle, ψ , can be specified independently. It can either be constant, ψ_0 , or time varying, $\psi_T(t)$. Since the model is linearized, differential flatness is not necessary for trajectory planning here. Let \mathbf{r}_T denote the closest point on the desired trajectory to the current position \mathbf{r} , and let the desired velocity and acceleration obtained by differentiating the specified trajectory be given by $\dot{\mathbf{r}}_T$ and $\ddot{\mathbf{r}}_T$, respectively. The desired trajectory,

$$\mathbf{z}^{\text{des}} = \begin{bmatrix} \mathbf{r}_T(t) \\ \psi_T(t) \end{bmatrix},$$

will be provided by a trajectory planner as an input to specify the trajectory of the position vector and the yaw angle that the vehicle should track. These are the flat outputs which were discussed in Sect. 16.2.6. Any continuous trajectory in the flat outputs specifies the trajectory for the system and the required inputs.

Thus, appropriate trajectories can be planned in this space. In particular, for hovering, $\mathbf{r}_T(t) = \mathbf{r}_0$ and $\psi_T(t) = \psi_0$.

The command accelerations, $\ddot{\mathbf{r}}_i^{\text{des}}$, are calculated from a proportional plus derivative (PD) controller. Define the position error in terms of components using the standard reference triad \mathbf{a}_i by

$$e_p = (r_T - r).$$

Similarly, define the velocity error by

$$e_v = \dot{r}_T - \dot{r}.$$

In order to guarantee that this error goes exponentially to zero, it is required that

$$(\ddot{r}_{i,T} - \ddot{r}_i^{\text{des}}) + k_{d,i}(\dot{r}_{i,T} - \dot{r}_i) + k_{p,i}(r_{i,T} - r_i) = 0, \quad (16.28)$$

where $\dot{r}_{i,T} = \ddot{r}_{i,T} = 0$ for hover.

Equation (16.26) leads to the relationship between the desired accelerations and roll and pitch angles. Given that $\Delta\theta = \theta - \theta_0 = \theta$ and $\Delta\phi = \phi - \phi_0 = \phi$,

$$\ddot{r}_1^{\text{des}} = g(\theta^{\text{des}} \cos \psi_T + \phi^{\text{des}} \sin \psi_T) \quad (16.29a)$$

$$\ddot{r}_2^{\text{des}} = g(\theta^{\text{des}} \sin \psi_T - \phi^{\text{des}} \cos \psi_T) \quad (16.29b)$$

$$\ddot{r}_3^{\text{des}} = \frac{1}{m}u_1 - g. \quad (16.29c)$$

For hover, (16.29c) yields

$$u_1 = mg + m\ddot{r}_3^{\text{des}} = mg - m(k_{d,3}\dot{r}_3 + k_{p,3}(r_3 - r_{3,0})). \quad (16.30)$$

Equations (16.29a) and (16.29b) can be used to compute the desired roll and pitch angles for the attitude controller:

$$\phi^{\text{des}} = \frac{1}{g}(\ddot{r}_1^{\text{des}} \sin \psi_T - \ddot{r}_2^{\text{des}} \cos \psi_T) \quad (16.31a)$$

$$\theta^{\text{des}} = \frac{1}{g}(\ddot{r}_1^{\text{des}} \cos \psi_T + \ddot{r}_2^{\text{des}} \sin \psi_T). \quad (16.31b)$$

The desired roll and pitch velocities are taken to be zero.

$$p^{\text{des}} = 0 \quad (16.32a)$$

$$q^{\text{des}} = 0. \quad (16.32b)$$

Since the yaw, $\psi_T(t)$, is prescribed independently by the trajectory generator, the controller uses

$$\psi^{\text{des}} = \psi_T(t) \quad (16.33a)$$

$$\dot{\psi}^{\text{des}} = \dot{\psi}_T(t). \quad (16.33b)$$

These equations provide the setpoints for the attitude controller in (16.27).

Note that the attitude controller must run an order of magnitude faster than the position control loop in order for this to work. In practice as discussed in Michael et al. (2010), the position controller runs at 100 Hz, while the inner attitude control loop runs at 1 kHz.

An alternate error metric can be used for more aggressive trajectories which only does not consider error in the tangent direction. Let the unit tangent vector of the trajectory (unit vector along $\dot{\mathbf{r}}_T$) be $\hat{\mathbf{t}}$. The unit normal to the trajectory, $\hat{\mathbf{n}}$, is derived by differentiating the tangent vector with respect to time or arc length, and finally, the unit binormal vector is the cross product $\hat{\mathbf{b}} = \hat{\mathbf{t}} \times \hat{\mathbf{n}}$. The position and velocity errors are defined according to the following equations:

$$\mathbf{e}_p = ((\mathbf{r}_T - \mathbf{r}) \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + ((\mathbf{r}_T - \mathbf{r}) \cdot \hat{\mathbf{b}})\hat{\mathbf{b}}$$

and

$$\mathbf{e}_v = \dot{\mathbf{r}}_T - \dot{\mathbf{r}}.$$

Note that here the position error in the tangent direction is ignored, only considering position error in the plane that is normal to the curve at the closest point. Once again the commanded acceleration, $\ddot{\mathbf{r}}_i^{\text{des}}$, is calculated from the PD feedback as shown in (16.28). In vector notation, this is

$$(\ddot{\mathbf{r}}_T - \ddot{\mathbf{r}}^{\text{des}}) + \mathbf{k}_d \mathbf{e}_v + \mathbf{k}_p \mathbf{e}_p = 0. \quad (16.34)$$

Finally, (16.31)–(16.33) is used to compute the desired roll and pitch angles. Again the reader is referred to Michael et al. (2010) for more details including experimental results obtained from these controllers.

16.3.1.4 Linear Quadratic Regulators

The gains used in a controller have a critical effect on its performance and stability. Since the model is linearized, the controller can be framed as a linear quadratic regulator. This allows the problem of choosing gains to be formatted as an optimization problem. A quadratic functional can be used for the cost:

$$J = \int_0^\infty (\mathbf{x} - \mathbf{x}_0)^T \mathbf{Q} (\mathbf{x} - \mathbf{x}_0) + (\mathbf{u} - \mathbf{u}_0)^T \mathbf{R} (\mathbf{u} - \mathbf{u}_0) dt \quad (16.35)$$

which is subject to the dynamic constraints that $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$. \mathbf{Q} and \mathbf{R} are chosen to weight the position errors and control efforts depending on the task at hand.

The analytic solution can be found using calculus of variations, which leads to a controller of the form

$$(\mathbf{u} - \mathbf{u}_0) = -\mathbf{K}(\mathbf{x} - \mathbf{x}_0). \quad (16.36)$$

K can be solved for using the Riccati equation, which can be found in many texts on optimal control, such as Kirk (2004).

16.3.2 Nonlinear Control

In order to control the quadrotor along trajectories that require large rotations, a different control law is needed. A nonlinear controller can track trajectories in the differentially flat output space defined by $\sigma(t) = [\mathbf{r}_{\text{des}}(t)^T, \psi_{\text{des}}(t)^T]^T$. The error metrics on position and velocity are defined as

$$\mathbf{e}_p = \mathbf{r} - \mathbf{r}_{\text{des}}, \mathbf{e}_v = \dot{\mathbf{r}} - \dot{\mathbf{r}}_{\text{des}}. \quad (16.37)$$

The vector describing the total force to apply to the quadrotor is computed using

$$\mathbf{F}_{\text{des}} = -K_p \mathbf{e}_p - K_v \mathbf{e}_v - m\mathbf{g}\mathbf{a}_3 + m\ddot{\mathbf{r}}_{\text{des}}. \quad (16.38)$$

With K_p and K_v being positive definite matrices of gains. The first input is chosen to be the desired force vector projected onto the body frame z axis of the quadrotor as such:

$$u_1 = -\mathbf{F}_{\text{des}} \cdot \mathbf{b}_3. \quad (16.39)$$

For the attitude control, the z axis of the quadrotor body frame should align with the desired force vector, as the quadrotor can only produce force along this axis. Using this fact and the specified differentially flat outputs, the desired rotation matrix R_{des} can be defined. Next, the error metric for rotation is defined as

$$\mathbf{e}_R = \frac{1}{2}(R_{\text{des}}^T R - R^T R_{\text{des}})^\vee \quad (16.40)$$

where \vee represents the “unhat” operator which transforms a skew-symmetric matrix into a vector. The vector is arranged such that $Ab = A^\vee \times b$ for any skew-symmetric matrix A and vector b . The derivative error metric for attitude control will be

$$\mathbf{e}_\omega = {}^A\omega^B - {}^A\omega_{\text{des}}^B. \quad (16.41)$$

${}^A\omega_{\text{des}}^B$ can be found from the differential flatness derivation. So the attitude controller will simply be a PD controller using these two error metrics with some gains:

$$\mathbf{u}_2 = \mathcal{I}(-K_R \mathbf{e}_R - K_\omega \mathbf{e}_\omega) + {}^A\omega^B \times \mathcal{I}({}^A\omega^B) - \mathcal{I}(\hat{\omega} R^T R_{\text{des}} \omega_{\text{des}} - R^T R_{\text{des}} \dot{\omega}_{\text{des}}). \quad (16.42)$$

An advantage to using this controller design is that under a few assumptions, statements can be made about the stability of the controller. According to Lee et al. (2010), if actuator saturation is ignored and K_R and K_ω are chosen to be positive, the attitude error dynamics are exponentially stable provided the initial configuration of the quadrotor satisfies the following:

$$\text{tr}(I - R_{\text{des}}^T(0)R(0)) < 2 \quad (16.43)$$

$$\|e_\omega(0)\|^2 < \frac{2}{\lambda_{\max}(\mathcal{I})} k_R (2 - \frac{1}{2} \text{tr}(I - R_{\text{des}}^T(0)R(0))). \quad (16.44)$$

If the initial conditions of the vehicle are in a smaller sublevel set satisfying

$$\text{tr}(I - R_{\text{des}}^T(0)R(0)) < 1 \quad (16.45)$$

then the complete dynamics are exponentially stable. Both of these propositions are proved in Lee et al. (2010). This controller has the advantage that it functions directly on $SE(3)$, and avoids the singularities and complications associated with Euler angles or quaternion representations.

Another property of note is that the basin of attraction of the attitude control subsystem of this controller increases as the vehicle scale decreases, which can be seen from the condition on the angular rate, since the maximum angular velocity error permissible is proportional to the inverse of the maximum eigenvalue of the inertia. Thus, smaller inertia tensors lead to stabilization starting with higher angular velocities. This is further explained in Kumar and Michael (2011).

In practice, it is possible to neglect the nonlinear terms (${}^A\omega^B \times \mathcal{I}^A\omega^B$ and $\mathcal{I}(\hat{\omega} R^T R_{\text{des}} \omega_{\text{des}} - R^T R_{\text{des}} \dot{\omega}_{\text{des}})$) and still achieve good results (Mellinger and Kumar 2011). This neglect can be justified by inspecting the individual terms of the attitude controller. The first two terms, $-\mathcal{I}K_R \mathbf{e}_R$ and $-\mathcal{I}K_\omega \mathbf{e}_\omega$, are on the order of $I_{zz}\omega_z$ for both the Nano Quadrotor and the Hummingbird. The nonlinear terms are both on the order of $I_{zz}\omega_z^2$. Since ω_z is usually much smaller than 1, this means that the nonlinear terms are much smaller than the linear terms.

16.4 Experimental Results

This section presents experimental results for the linearized control of the Nano Quadrotor platform. The reader is referred to Mellinger and Kumar (2011) for results for the Hummingbird platform. A diagram of the system architecture used with the Nano Quadrotor is presented in Fig. 16.5. The system consists of the Vicon motion capture system (Vicon MX Systems 2012) for perception, which tracks the position and orientation of the quadrotor using an array of cameras and reflective markers. The Vicon system sends data to a desktop computer where the high-level decisions and position control are performed. The desktop communicates via wireless module to the Nano Quadrotor, which perform attitude control onboard.

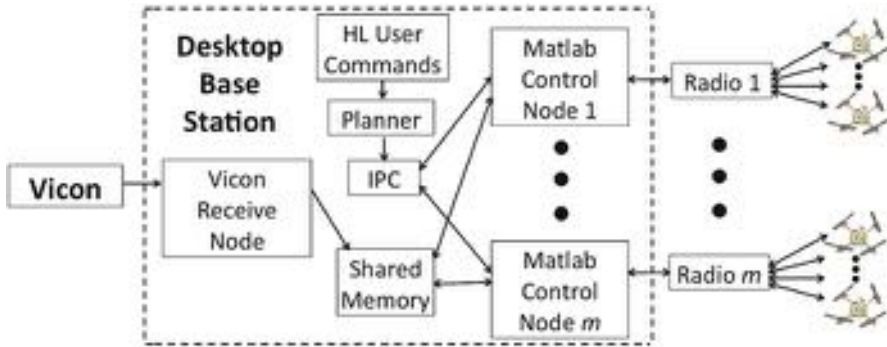


Fig. 16.5 System architecture for Nano Quadrotor. There are three software modules: (a) the Vicon interface; (b) the user commands or trajectory planner; and (c) the control software for each vehicle

The diagram shows how several control nodes may be used to control a swarm of Nano Quadrotors. A similar architecture is used for the Hummingbirds. Each component of the system is described below.

16.4.1 Vicon Motion Capture System

The Vicon motion capture system provides a state estimate for the quadrotor, which is nearly ground truth. The Vicon system offers three important features. First, *it is fast*: the system can be run at or below 375 Hz. Second, *it is precise*: experimental tests show that the deviations of position estimates for single static markers are on the order of $50\text{ }\mu\text{m}$ which is well beyond the requirements for flight. Lastly, *it is robust*: the Vicon *Tracker* software assumes that markers in models are rigidly attached which enables the system to maintain tracking even if all but one camera are occluded. Using this software, tracking of quadrotors is rarely lost, even during extreme situations such as fast maneuvers (speeds of $3.5\text{ }\frac{\text{m}}{\text{s}}$, accelerations of $15\text{ }\frac{\text{m}}{\text{s}^2}$, and angular speeds of $1,000\text{ }\frac{\circ}{\text{s}}$).

16.4.2 Software and Integration

The control software for the quadrotors is integrated with the motion capture using a custom interprocess communication (IPC) framework as shown in Fig. 16.5. The high-level control such as trajectory generation and position control as in Sect. 16.3.1.3 is done using MATLAB (2011), which then sends commands to the robot via XBee radios using the ZigBee communication protocol at 100 Hz. Low-level processing such as attitude and motor control (16.27) is performed onboard.

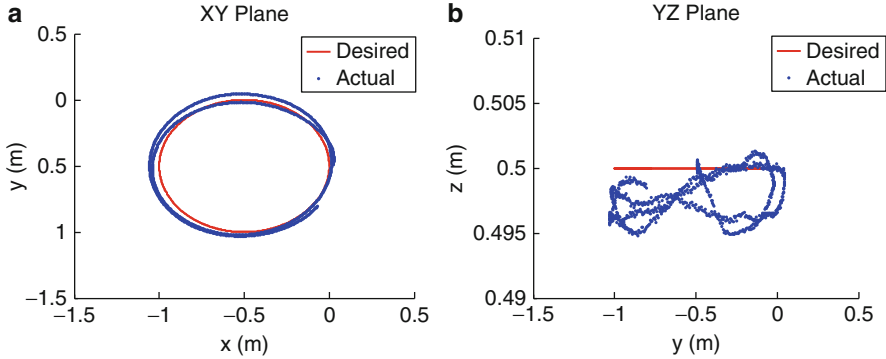


Fig. 16.6 The desired circular trajectory and the actual trajectory, run at $1.2 \frac{\text{rad}}{\text{s}}$, projected along the x - y horizontal plane and a vertical y - z plane. Note the small scale on the z axis

16.4.3 Results

As discussed above in Sects. 16.2.6 and 16.3, trajectories for the quadrotor can be defined by $r(t)$ and $\psi(t)$. This subsection presents results for the linearized controller running on the Nano platform for two different trajectories.

First, Figs. 16.6 and 16.7 show the trajectory and the errors for a Nano Quadrotor commanded to fly in a circular trajectory of radius 0.5 m. The maximum accelerations encountered are on the order of $0.008 \frac{\text{m}}{\text{s}^2}$. The errors are on the order of 10 cm in the horizontal plane and less than 1 cm in the vertical plane. While the errors in the horizontal plane are significantly higher than the errors in the vertical direction, the errors never exceed 22 cm.

Similarly, Figs. 16.8 and 16.9 present the trajectory and errors for a Nano Quadrotor flying in a helical trajectory of radius 0.35 m and pitch $0.05 \frac{\text{m}}{\text{rad}}$. The errors are on the order of 10 cm in the horizontal plane and less than 1 cm in the vertical plane. Although there is a vertical motion in this experiment, the horizontal errors remain significantly larger than the vertical errors. The errors in this experiment never exceed 26 cm.

These are two representative experiments. The maximum roll angles for the circle and helix trajectories were 0.11 and 0.27 rad, respectively. Similarly, the maximum pitch angles for the circle and helix trajectories were 0.19 and 0.51 rad. As the requested speed of the quadrotor around the circle increases, so does the required excursion from hover. For a circle trajectory run at $2 \frac{\text{rad}}{\text{s}}$, the maximum roll and pitch angles were 0.82 and 0.41 rad, much larger than the circle run at $1.2 \frac{\text{rad}}{\text{s}}$. In general, as the commanded trajectory becomes more aggressive in terms of changes in speed and centripetal acceleration, the roll and pitch angles required increase, and the errors in trajectory following increase. In this work the experiments were conducted far away from walls and ceilings. As shown in Powers et al. (2012), ground and ceiling effects can cause significant disturbances which are not easily characterized using the models in this chapter.

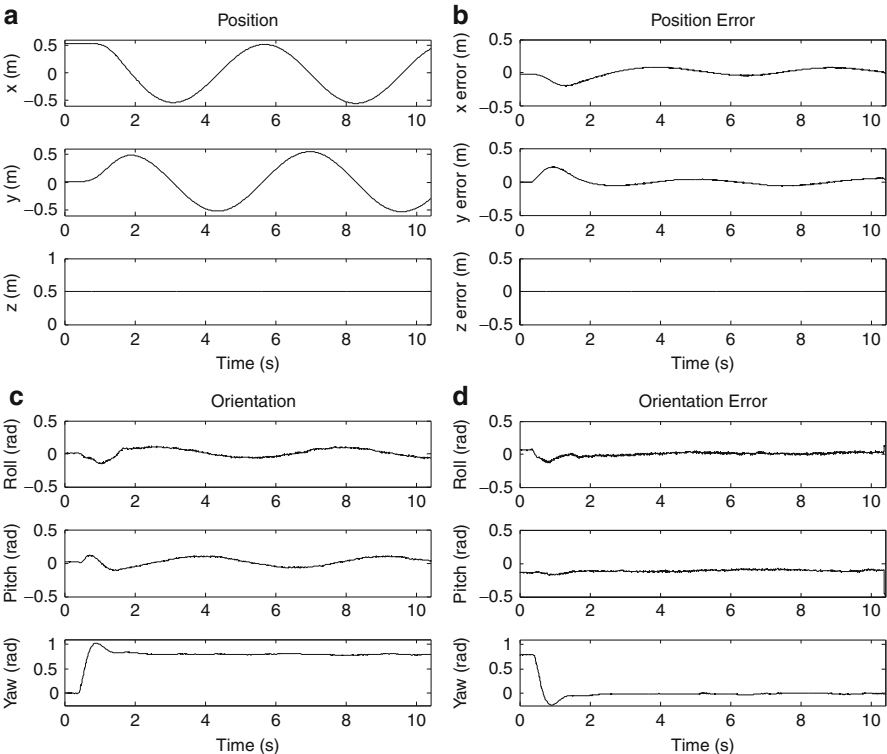


Fig. 16.7 The position and orientation history and the errors in trajectory control for the experiment with the circular trajectory

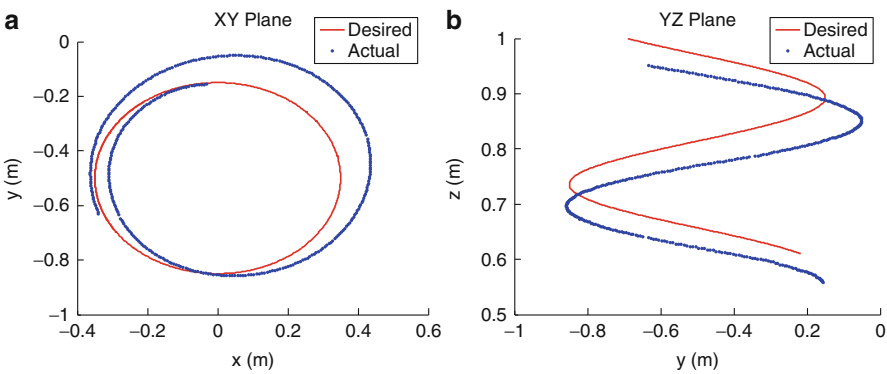


Fig. 16.8 The desired helical trajectory and the actual trajectory, run at $2 \frac{\text{rad}}{\text{s}}$ and $0.05 \frac{\text{m}}{\text{rev}}$, projected along the x - y horizontal plane and a vertical y - z plane

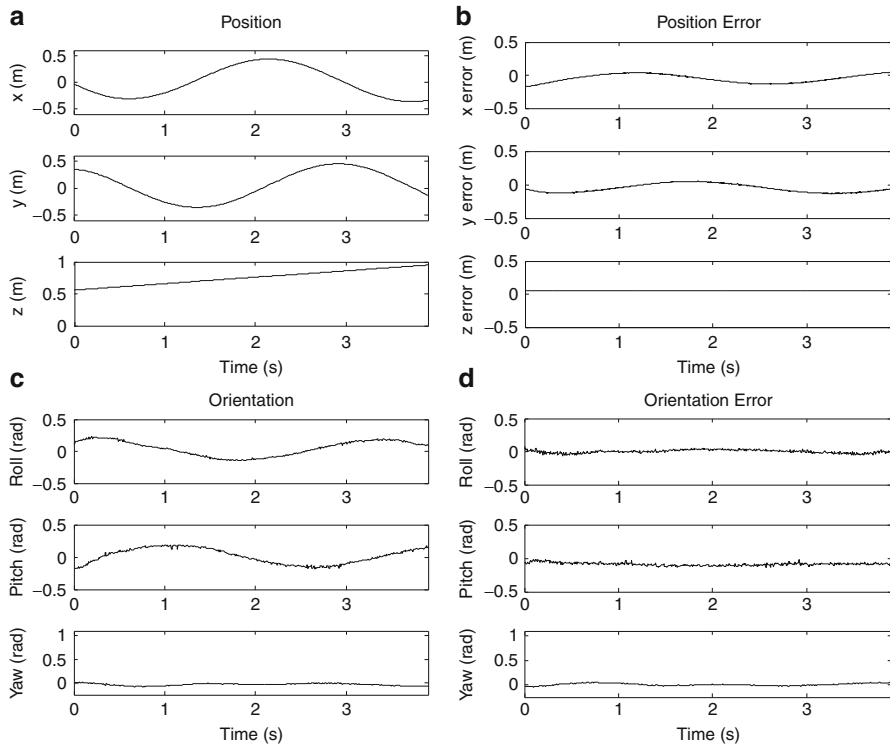


Fig. 16.9 The position and orientation history and the errors in trajectory control for the experiment with the helical trajectory

16.5 Conclusion

This chapter presents a model of the rigid body dynamics for a quadrotor. The quadrotor is underactuated with a 12-dimensional state space and four actuators. The system is shown to be differentially flat. Two controllers are discussed. The first is a linear controller based on a linearized model of the dynamics. The second is a nonlinear controller that is effective for operation in conditions where the roll and pitch angles are larger. Finally, experimental results with quadrotors are shown to illustrate the basic approach to modeling and control.

References

- Ascending Technologies (2012), <http://www.asctec.de>
D. Gurdan, J. Stumpf, M. Achtelik, K. Doth, G. Hirzinger, D. Rus, Energy-efficient autonomous four-rotor flying robot controlled at 1 khz, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, April 2007

- D.E. Kirk, *Optimal Control Theory* (Dover, Mineola, 2004)
- KMel Robotics (2011), <http://www.kmelrobotics.com>
- V. Kumar, N. Michael, Opportunities and challenges with autonomous micro aerial vehicles, in *International Symposium of Robotics Research*, Flagstaff, Aug 2011
- A. Kushleyev, D. Mellinger, V. Kumar, Towards a swarm of agile micro quadrotors, in *Robotics: Science and Systems (RSS)*, ed. by H.F. Durrant-Whyte, N. Roy, P. Abbeel (MIT, Cambridge, 2012)
- T. Lee, M. Leok, N. McClamroch, Geometric tracking control of a quadrotor uav on SE(3), in *Proceedings of the IEEE Conference on Decision and Control*, Atlanta, 2010
- Q.J. Lindsey, D. Mellinger, V. Kumar, Construction of cubic structures with quadrotor teams, in *Robotics: Science and Systems*, ed. by Y. Matsuoka, H.F. Durrant-Whyte, J. Neira (MIT, Cambridge, 2011)
- MATLAB, R2011b, (The MathWorks, Natick, 2011)
- D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, May 2011
- N. Michael, D. Mellinger, Q. Lindsey, V. Kumar, The grasp multiple micro uav testbed. *IEEE Robot. Autom. Mag.* **17**, 56–65 (2010)
- N. Michael, J. Fink, V. Kumar, Cooperative manipulation and transportation with aerial robots. *Auton. Robots* **30**, 73–86 (2011). ISSN: 0929-5593. <http://dx.doi.org/10.1007/s10514-010-9205-0> doi:10.1007/s10514-010-9205-0
- C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, V. Kumar, Influence of aerodynamics and proximity effects in quadrotor flight, in *International Symposium on Experimental Robotics (ISER)*, Québec, June 2012
- S. Shen, N. Michael, V. Kumar, Autonomous multi-floor indoor navigation with a computationally constrained mav, in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, Shanghai, pp. 20–25. doi:10.1109/ICRA.2011.5980357
- S. Shen, N. Michael, V. Kumar, Autonomous indoor 3d exploration with a micro-aerial vehicle, in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, May 2012, pp. 9–15. doi:10.1109/ICRA.2012.6225146
- Skybotix Technologies (2012), www.skybotix.com
- M. van Nieuwstadt, M. Rathinam, R.M. Murray, Differential flatness and absolute equivalence, in *Proceedings of the 33rd IEEE Conference on Decision and Control*, 1994, Lake Buena Vista, Dec 1994, vol. 1, pp. 326–332. doi:10.1109/CDC.1994.410908
- Vicon MX Systems (2012), <http://www.vicon.com/products/viconmx.html>

David B. Doman, Michael W. Oppenheimer, and
David O. Sigthorsson

Contents

| | | |
|------|---|-----|
| 17.1 | Introduction | 330 |
| 17.2 | Idealized Wing Motion | 331 |
| 17.3 | Instantaneous Aerodynamic Forces and Moments | 334 |
| 17.4 | Quasi-equilibrium Hover and Vehicle Design | 337 |
| 17.5 | Flapping Wing Aerodynamic Control Derivatives | 338 |
| 17.6 | Influence of Control Mechanisms on Wing Motion and Cycle-Averaged Generalized Forces | 338 |
| 17.7 | Flapping Wing Aerodynamic Stability Derivatives | 341 |
| 17.8 | Simulation Methods | 344 |
| 17.9 | Conclusion | 345 |
| | References | 345 |

Abstract

The methods presented in this chapter are for use with hover-capable flapping wing vehicles in the conceptual design phase. The class of flapping wing vehicles considered are those that maneuver by making changes to the motion of flapping wings, rather than making use of conventional aerodynamic surfaces.

D.B. Doman (✉)

Control Design and Analysis Branch, Air Force Research Laboratory AFRL/RBCA,
Wright-Patterson AFB, OH, USA
e-mail: david.doman@wpafb.af.mil

M.W. Oppenheimer

Control Design and Analysis Branch, Air Force Research Laboratory, Wright-Patterson AFB,
OH, USA
e-mail: michael.oppenheimer@wpafb.af.mil

D.O. Sigthorsson

General Dynamics Information Technology, Air Force Research Laboratory, Wright-Patterson
AFB, OH, USA
e-mail: sigthorsson@gmail.com

Methods are presented that allow one to iterate between vehicle design, trim, and controllability analysis until closure is achieved. The motion of the flapping wings is characterized by seven wing motion parameters: wing stroke amplitude, symmetric and asymmetric frequency, bias, upstroke and downstroke angle-of-attack limits, and wing stroke plane inclination. This chapter presents a table of cycle-averaged control derivatives that can be used with the multivariable chain rule to produce control derivative estimates for a diverse set of wing beat generation and control mechanisms that might be encountered in flapping wing vehicle designs. A method is presented that will enable the designer to estimate how the seven wing motion parameters change in response to changes in the control mechanism position; a step that is necessary to make use of the chain rule. A method for estimating the combined effect of fuselage motion and flapping wing motion is presented as well. Control design considerations and simulation methods are also discussed.

17.1 Introduction

The flight dynamics and control of flapping wing micro air vehicles (MAVs) are significantly more complex than those of fixed wing and even rotary-winged aircraft. The class of MAVs considered in this chapter make use of variations in the motion of flapping wings to affect changes to the aircraft trajectory. This class of aircraft does not make use of conventional steady aerodynamic surfaces such as elevators, rudders, and ailerons for control. Part of the complexity associated with such aircraft arises because the forces and moments produced by flapping wings are periodic in nature. Additionally, for flight control and simulation purposes, these forces are transformed into a body-fixed reference frame that results in the signs of these forces becoming functions of the wing position and attitude relative to the vehicle center-of-mass. It is well recognized that unsteady aerodynamic effects (Sane and Dickinson 2001), fluid-structure interactions (Stanford and Beran 2010), and wing-load interactions with drive mechanisms (Doman et al. 2011) play a role in the instantaneous behavior of the forces and moments. Nevertheless, a number of researchers have reported (Anderson 2011; Ol and Granlund 2012; Doman et al. 2010a; Deng et al. 2006a, b) that adequate controllers for such vehicles can be designed by making use of cycle-averaged models of the aerodynamic forces and moments. The success of this idea is due in large part to the differences between the frequency of the wingbeats and the time constants associated with the motion of the fuselage. These models map changes in the generalized aerodynamic forces to wing kinematic properties that vary slowly when compared to the frequency of the flapping wings. When the control problem is viewed from the cycle-averaged perspective, conventional control approaches have been shown to provide adequate tracking performance when applied to simulations of vehicles that model periodic and unsteady aerodynamic forces and moments.

This book chapter provides a synopsis of analytical results that facilitate vehicle simulation, design for trim, and the computation of cycle-averaged aerodynamic

stability and control derivatives for flapping winged aircraft about a quasi-equilibrium hover condition. Vehicle design parameters appear in the analytical expressions and provide the designer with a tool that connects vehicle design to controllability. The results presented in this chapter were derived using blade element theory and the assumption that the wing stroke position can be characterized as piecewise cosinusoidal. Seven variations in wing motion are considered, namely, symmetric and asymmetric wingbeat frequency, wing stroke amplitude and bias, upstroke and downstroke angle-of-attack limits, and wing stroke plane inclination. Control mechanisms on flapping wing aircraft influence one or more of these seven parameters and the control derivatives associated with a specific mechanism. One can compute the changes in the wingbeat kinematic parameters with respect to changes in the position of the control mechanism available on a specific aircraft design. Using the multivariable chain rule, this result can then be combined with cycle-averaged aerodynamic control derivatives to provide estimates of the cycle-averaged control derivatives for a wide range of flapping wing vehicle control mechanisms. It is important to remember that these estimates will not include the effects of unsteady aerodynamics, fluid-structure interactions, wing inertial effects, or wing-load interactions with the drive mechanism. However, these estimates do provide a point of departure for a control synthesis problem for a very complex system that may provide adequate performance when applied to the vehicle itself or high-fidelity simulation models and can, if necessary, be subsequently tuned to account for more complex phenomena.

17.2 Idealized Wing Motion

Seven parameters are considered that, when taken together, can approximate a wide range of wing movements. Figure 17.1 illustrates a generic hover-capable MAV or animal that has the ability to change these seven parameters. The illustrations show the body-axes coordinate system (x_B, y_B, z_B) , the wing root coordinate system, (x_r, y_r, z_r) , and the geometry of the wing. The center-of-gravity of the vehicle is offset from the wing root hinge point in the x_B, z_B plane by $\Delta x, \Delta z$, while the distance between the wing root hinge points is defined as w . The wing root y-axis, y_r , is fixed, while the wing planform axis, y_{wp} moves with the wing. The y_B axis points in the direction of the right wing when it passes through its neutral position. The center-of-pressure of the wing is defined by the location x_{cp}, y_{cp} , written in a wing-carried planform frame (x_{wp}, y_{wp}) .

The mathematical model that is used to describe the motion of the wing root or spar is given by

$$\phi_U(t) = A \cos [(\omega - \delta)t] + \eta, \quad 0 \leq t < \frac{\pi}{(\omega - \delta)} \quad (17.1)$$

$$\phi_D(t) = A \cos [(\omega + \sigma)t + \xi] + \eta, \quad \frac{\pi}{(\omega - \delta)} \leq t < \frac{2\pi}{\omega} \quad (17.2)$$

where $\phi_U(t)$ and $\phi_D(t)$ denote the angular position associated with a wing on the upstroke and downstroke, respectively, A denotes the stroke amplitude, η defines the wing bias or the wing oscillation offset from the zero position, ω denotes the symmetric frequency, δ is the asymmetric frequency that allows the upstroke and downstroke velocity profiles to differ,

$$\sigma \triangleq \frac{\delta\omega}{\omega - 2\delta} \quad (17.3)$$

and the phase shift, which enforces a matching condition at stroke reversal, is given by

$$\xi \triangleq \frac{-2\pi\delta}{\omega - 2\delta} \quad (17.4)$$

The bottom four diagrams in Fig. 17.1 show the effects of control variables on the wingbeat motion. Shown in the first subplot in the second row of Fig. 17.1 are the effects of a positive asymmetric frequency, δ , and a positive bias, η . When $\delta > 0$, the term $(\omega - \delta)$ in Eq. 17.1 is decreased, resulting in a smaller frequency, as shown by more lines in the strobed animation. In turn, for $\delta > 0$, the term $(\omega + \sigma)$ is increased, resulting in a larger frequency and fewer lines in the strobed animation. The bias simply shifts all points in the wingbeat profile fore or aft of the nominal neutral position of the wing spar, as can be seen with the right wing in the first subplot of the second row of Fig. 17.1. The remaining subplots show the effects of upstroke and downstroke angle-of-attack and stroke plane variation.

The subsequent information presented allows for the independent actuation of wings. In other words, the right and left wings are taken to be driven by independent motion sources; however, the results are also applicable to designs where the left and right wings are coupled. Figure 17.2 graphically illustrates the effects of differences in asymmetric frequency, bias, and amplitude. The top two plots show the effect of asymmetric frequency for both positive and negative values of δ . The bottom figures shows the wing bias, η , and the amplitude adjustment, A .

The wingbeat cycle is split into two parts, the first part is the upstroke, while the second part is the downstroke. Considering Fig. 17.2, the upstroke is defined as moving from a wing position of 1 rad to a wing position of -1 rad, while the downstroke moves from a wing position of -1 rad to a wing position of 1 rad. A positive asymmetric frequency, δ , impedes or slows down the upstroke while the downstroke advances or speeds up, as shown in the upper left plot in Fig. 17.2. A negative asymmetric frequency has the opposite effect in that the upstroke is advanced, while the downstroke is impeded. The wing bias shifts the midstroke wingbeat angular position fore or aft by the bias amount, while the amplitude adjusts the limits of wing stroke.

Both the bias and amplitude variation, as presented in Fig. 17.2, yield discontinuous wingbeat positions. The physical mechanism used to produce the wingbeat motion would not allow such discontinuities and would smooth the discontinuous functions but would result in an actual wingbeat motion that is different from that which was commanded. To avoid this issue, techniques exist that modify the

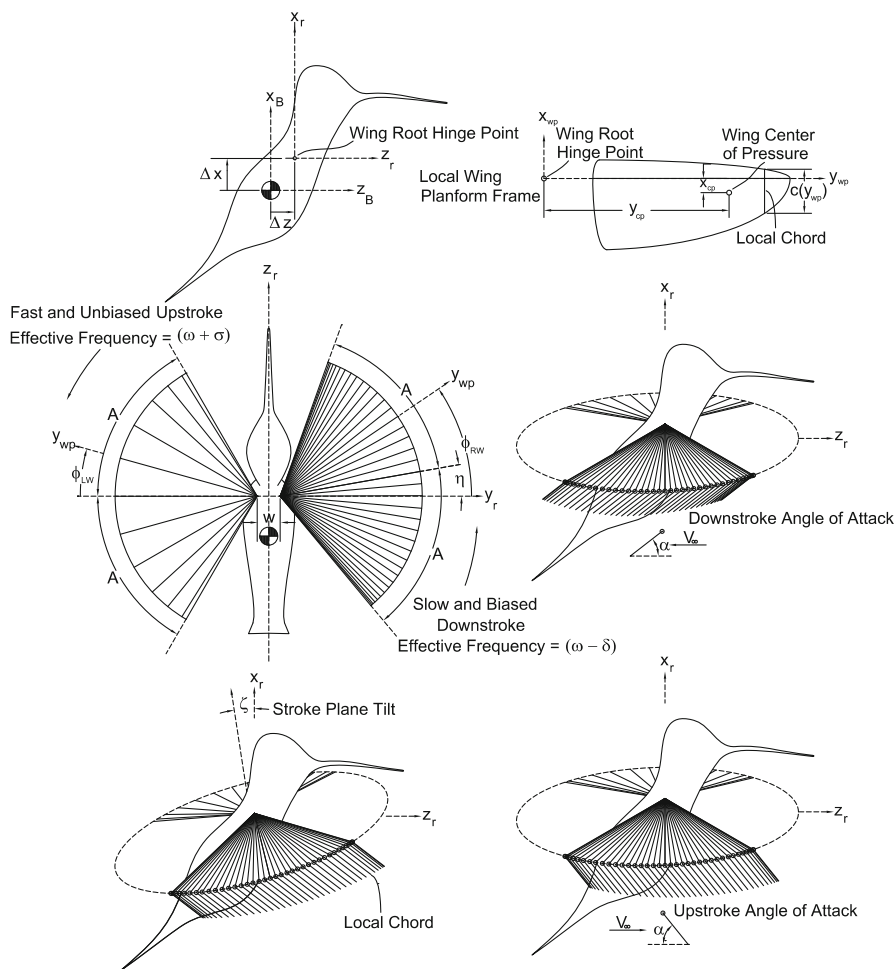


Fig. 17.1 Strobed illustration of effect of variations in wing kinematics and definition of parameters

wingbeat forcing functions to yield continuous functions (Doman et al. 2010b; Oppenheimer et al. 2011a) even when the bias and/or amplitude change from one wingbeat cycle to another.

The aerodynamic model employed to generate the results in this work is applicable to vehicle designs that employ passive angle-of-attack regulation in the form of limiters (Wood 2008; Doman and Regisford 2010). The subsequent results assume that the wing angle-of-attack changes instantaneously at the end of an upstroke or downstroke and remains on a limit, α_u or α_d , over the course of an upstroke or downstroke, respectively. Finally, the model allows for the variation in stroke plane inclination, ζ , which allows the resultant aerodynamic force vector to be inclined with respect to the MAV fuselage.

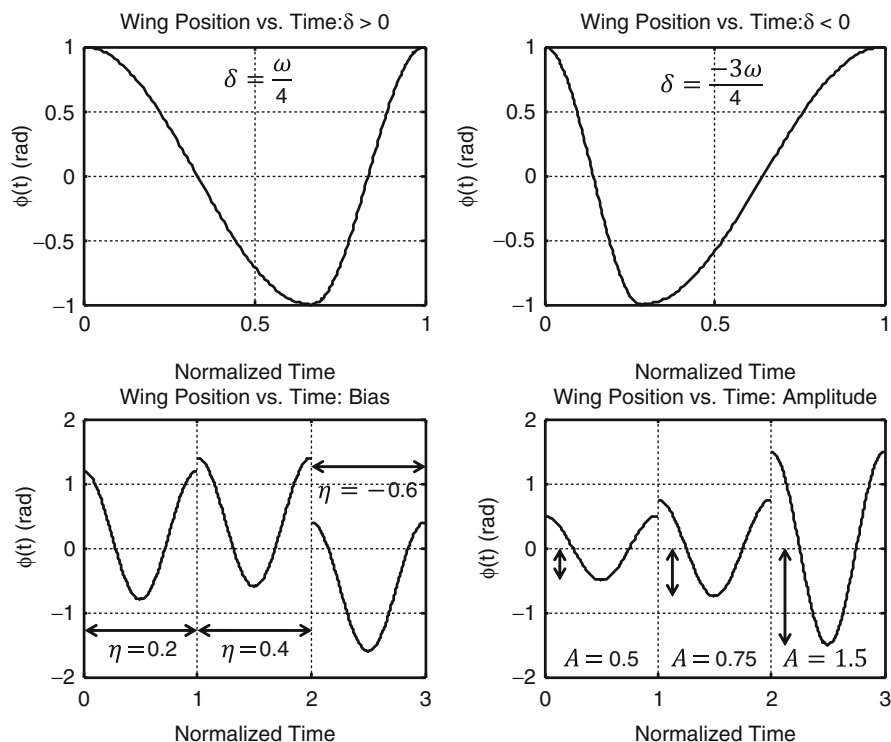


Fig. 17.2 Examples of split cycle, bias, and amplitude wingbeat variations

17.3 Instantaneous Aerodynamic Forces and Moments

The seven parameters used to describe the motion of the wings can be used to approximate a wide range of wing motions; however, in practice, the actual motion is produced by drive elements and mechanisms and will therefore vary from the idealized model of wingbeat motion presented in Fig. 17.2. Nonlinear linkage-based mechanisms, load interactions with drive elements, and wing flexibility are just a few phenomena that can cause such departures from the idealized model of wing motion. The equations below provide a blade element-based estimate of the instantaneous forces and moments that act on a flapping wing vehicle. The equations are useful for simulations that are used to test the performance and robustness of model-based control laws designed using cycle-averaged models of such aircraft. They are also useful for conceptual design studies and computing quasi-equilibrium trim conditions for such vehicles. Note that the angular displacement of the wing in the stroke plane, $\phi(t)$; the wing angle-of-attack, $\alpha(t)$; and the stroke plane

Table 17.1 Aerodynamic forces expressed in body frame

| Phase | Body frame force vector |
|---------------|--|
| RW upstroke | $\mathbf{F}_{\text{RWU}}^B(t) = \begin{bmatrix} L(t) \cos \zeta(t) + D(t) \cos \phi(t) \sin \zeta(t) \\ -D(t) \sin \phi(t) \\ -L(t) \sin \zeta(t) + D(t) \cos \phi(t) \cos \zeta(t) \end{bmatrix}$ |
| RW downstroke | $\mathbf{F}_{\text{RWD}}^B(t) = \begin{bmatrix} L(t) \cos \zeta(t) - D(t) \cos \phi(t) \sin \zeta(t) \\ D(t) \sin \phi(t) \\ -L(t) \sin \zeta(t) - D(t) \cos \phi(t) \cos \zeta(t) \end{bmatrix}$ |
| LW upstroke | $\mathbf{F}_{\text{LWU}}^B(t) = \begin{bmatrix} L(t) \cos \zeta(t) + D(t) \cos \phi(t) \sin \zeta(t) \\ D(t) \sin \phi(t) \\ -L(t) \sin \zeta(t) + D(t) \cos \phi(t) \cos \zeta(t) \end{bmatrix}$ |
| LW downstroke | $\mathbf{F}_{\text{LWD}}^B(t) = \begin{bmatrix} L(t) \cos \zeta(t) - D(t) \cos \phi(t) \sin \zeta(t) \\ -D(t) \sin \phi(t) \\ -L(t) \sin \zeta(t) - D(t) \cos \phi(t) \cos \zeta(t) \end{bmatrix}$ |

inclination, $\zeta(t)$, are written as general functions of time in the instantaneous model, in order to apply to a wide class of vehicles. Table 17.1 provides expressions for the instantaneous aerodynamic force vectors.

F_{RWU}^B and F_{RWD}^B denote forces associated with the right wing on the upstroke and downstroke written in the body-axes coordinate frame, respectively. F_{LWU}^B and F_{LWD}^B denote the same for the left wing. The lift, $L(t)$, and drag, $D(t)$, are expressed in wing-carried frames and are given by

$$L(t) = k_L \dot{\phi}(t)^2 \quad (17.5)$$

$$D(t) = k_D \dot{\phi}(t)^2 \quad (17.6)$$

where,

$$k_L = \frac{\rho}{2} I_A C_L(\alpha) \quad (17.7)$$

$$k_D = \frac{\rho}{2} I_A C_D(\alpha) \quad (17.8)$$

and I_A is the wing planform area moment of inertia about the spar root hinge point that defines the stroke plane. The lift and drag coefficients are given by the empirical formulae (Sane and Dickinson 2001)

$$C_L(\alpha) = 0.225 + 1.58 \sin(2.13\alpha - 7.2) \quad (17.9)$$

$$C_D(\alpha) = 1.92 - 1.55 \cos(2.04\alpha - 9.82) \quad (17.10)$$

where α is expressed in degrees. The center-of-pressure of each planform must be obtained in order to compute moments on the fuselage. The center-of-pressure location for each wing can be computed as

$$x_{cp} = \frac{\int_0^R y_{wp}^2 \frac{1}{2} c(y_{wp})^2 dy_{wp}}{\int_0^R y_{wp}^2 c(y_{wp}) dy_{wp}} \quad (17.11)$$

$$y_{cp} = \frac{\int_0^R y_{wp}^3 c(y_{wp}) dy_{wp}}{\int_0^R y_{wp}^2 c(y_{wp}) dy_{wp}} \quad (17.12)$$

where the subscript wp indicates a coordinate axis within a planform frame and $c(y_{wp})$ is the wing chord as a function of span. Let $\Delta \mathbf{r}_R^B$ and $\Delta \mathbf{r}_L^B$ denote the position vector from the vehicle center-of-gravity to the right and left wing root hinge points, i.e.,

$$\Delta \mathbf{r}_R^B \triangleq [\Delta x \ w/2 \ \Delta z]^T \quad (17.13)$$

$$\Delta \mathbf{r}_L^B \triangleq [\Delta x \ -w/2 \ \Delta z]^T \quad (17.14)$$

where $\Delta x, w, \Delta z$ are defined in Fig. 17.1. It is assumed that the vehicle center-of-gravity is fixed due to the wing mass being much lower than the mass of the fuselage and drive mechanisms. The centers of pressure associated with each wing and stroke, expressed in the body frame, are summarized in Table 17.2.

The expressions for the aerodynamic moments associated with each wing and stroke are given by

$$\begin{aligned} \mathbf{M}_{RWU}^B &= \mathbf{r}_{cpRWU}^B \times \mathbf{F}_{RWU}^B \\ \mathbf{M}_{RWD}^B &= \mathbf{r}_{cpRWD}^B \times \mathbf{F}_{RWD}^B \\ \mathbf{M}_{LWU}^B &= \mathbf{r}_{cpLWU}^B \times \mathbf{F}_{LWU}^B \\ \mathbf{M}_{LWD}^B &= \mathbf{r}_{cpLWD}^B \times \mathbf{F}_{LWD}^B \end{aligned} \quad (17.15)$$

Table 17.2 Centers of pressure for each wing expressed in the body frame

| CP location | Body frame expression |
|---------------|--|
| RW upstroke | $\mathbf{r}_{cpRWU}^B(t) = \begin{bmatrix} x_{cp} \sin \alpha(t) \cos \zeta(t) + \sin \zeta(t)(-x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta x \\ x_{cp} \sin \phi(t) \cos \alpha(t) + y_{cp} \cos \phi(t) + \frac{w}{2} \\ -x_{cp} \sin \alpha(t) \sin \zeta(t) + \cos \zeta(t)(-x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta z \end{bmatrix}$ |
| RW downstroke | $\mathbf{r}_{cpRWD}^B(t) = \begin{bmatrix} x_{cp} \sin \alpha(t) \cos \zeta(t) + \sin \zeta(t)(x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta x \\ -x_{cp} \sin \phi(t) \cos \alpha(t) + y_{cp} \cos \phi(t) + \frac{w}{2} \\ -x_{cp} \sin \alpha(t) \sin \zeta(t) + \cos \zeta(t)(x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta z \end{bmatrix}$ |
| LW upstroke | $\mathbf{r}_{cpLWU}^B(t) = \begin{bmatrix} x_{cp} \sin \alpha(t) \cos \zeta(t) + \sin \zeta(t)(-x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta x \\ -x_{cp} \sin \phi(t) \cos \alpha(t) - y_{cp} \cos \phi(t) - \frac{w}{2} \\ -x_{cp} \sin \alpha(t) \sin \zeta(t) + \cos \zeta(t)(-x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta z \end{bmatrix}$ |
| LW downstroke | $\mathbf{r}_{cpLWD}^B(t) = \begin{bmatrix} x_{cp} \sin \alpha(t) \cos \zeta(t) + \sin \zeta(t)(x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta x \\ x_{cp} \sin \phi(t) \cos \alpha(t) - y_{cp} \cos \phi(t) - \frac{w}{2} \\ -x_{cp} \sin \alpha(t) \sin \zeta(t) + \cos \zeta(t)(x_{cp} \cos \phi(t) \cos \alpha(t) + y_{cp} \sin \phi(t)) + \Delta z \end{bmatrix}$ |

17.4 Quasi-equilibrium Hover and Vehicle Design

For insight into how to design a tailless flapping wing aircraft that can be trimmed, it is useful to consider the expression for a cycle-averaged generalized force:

$$\bar{G} = \frac{\omega}{2\pi} \int_0^{\frac{2\pi}{\omega}} G(\phi(t)) dt \quad (17.16)$$

The necessary conditions for a flapping wing aircraft to achieve a quasi-equilibrium hover condition are that the cycle-averaged x-body force must be equal to the weight, W , of the aircraft and the cycle-averaged y- and z-body forces and body moment vector must be equal to zero, namely,

$$\bar{X}^B \triangleq \bar{X}_{LW}^B + \bar{X}_{RW}^B = W \quad (17.17)$$

$$\bar{Y}^B \triangleq \bar{Y}_{LW}^B + \bar{Y}_{RW}^B = 0 \quad (17.18)$$

$$\bar{Z}^B \triangleq \bar{Z}_{LW}^B + \bar{Z}_{RW}^B = 0 \quad (17.19)$$

$$\bar{M}_x^B \triangleq \bar{M}_{xLW}^B + \bar{M}_{xRW}^B = 0 \quad (17.20)$$

$$\bar{M}_y^B \triangleq \bar{M}_{yLW}^B + \bar{M}_{yRW}^B = 0 \quad (17.21)$$

$$\bar{M}_z^B \triangleq \bar{M}_{zLW}^B + \bar{M}_{zRW}^B = 0 \quad (17.22)$$

These conditions are necessary for a flapping wing aircraft to orbit a fixed point in space. These conditions are not sufficient for conventional steady equilibrium because of the periodic forces and moments produced by flapping wings. The periodicity of the forces causes limit-cycle oscillations to occur about a point, thus the use of the term quasi-equilibrium hover or trim condition. Furthermore, certain initial conditions must be met in order to avoid drifting away from the quasi-equilibrium point (Oppenheimer et al. 2010a). By making use of the instantaneous equations for the forces and moments from the previous section and computing cycle-averages from Eq. 17.16, one can solve for values of the seven wing motion parameters and center-of-gravity locations that satisfy the necessary conditions. In general, this calculation will have to be performed numerically. A typical method (Doman and Regisford 2010) for solving the trim problem follows:

1. Determine the stroke plane inclination, ζ , or upstroke and downstroke angle-of-attack combination, α_U , α_D , that causes $\bar{Z}^B = 0$ for a given drive mechanism that produces wing spar motion $\phi(t)$.
2. Adjust the symmetric frequency ω and/or amplitude A such that $\bar{X}^B = W$.
3. Determine the center-of-gravity location, Δx and Δz , and/or wing bias, η , such that $\bar{M}_y^B = 0$.
4. If the left and right wings are synchronized, then $\bar{Y}^B = \bar{M}_x^B = \bar{M}_z^B = 0$.

17.5 Flapping Wing Aerodynamic Control Derivatives

Cycle-averaged control derivatives can be physically interpreted as the sensitivities of the cycle-averaged aerodynamic forces and moments to changes in the parameters that characterize the flapping motion of the wings. The method by which such control derivatives can be obtained is outlined below. Let $G(t)$ denote a generalized periodic aerodynamic force of period $2\pi/\omega$ that is produced by a flapping wing. The generalized forces are written in body-axis coordinates where the origin of the coordinate frame is fixed to the fuselage center-of-gravity. A cycle-averaged generalized aerodynamic force is given by

$$\bar{G} = \frac{\omega}{2\pi} \left[\int_0^{\frac{\pi}{\omega-\delta}} G(t)dt + \int_{\frac{\pi}{\omega-\delta}}^{\frac{2\pi}{\omega}} G(t)dt \right] \quad (17.23)$$

The cycle-averaged control derivatives for a given generalized force are given by

$$\left[\partial \bar{G} / \partial \omega \quad \partial \bar{G} / \partial \delta \quad \partial \bar{G} / \partial A \quad \partial \bar{G} / \partial \eta \quad \partial \bar{G} / \partial \zeta \quad \partial \bar{G} / \partial \alpha_u \quad \partial \bar{G} / \partial \alpha_d \right] \Big|_{\text{hover}} \quad (17.24)$$

Table 17.3 was created using the procedure described above, and it presents all nonzero control derivatives associated with the seven wingbeat kinematic parameters for the specific case where $\zeta = \delta = \eta = 0$ at hover. More general expressions for the control derivatives may be found in Oppenheimer et al. (2010b, 2011b). In Table 17.3, $J_1(A)$ is a Bessel function of the first kind. Similarly $J_0(A)$ and $J_2(A)$ are Bessel functions of the zeroth and second kind. Finally the derivatives of k_L and k_D with respect to angle of attack are denoted by $d_L = \partial k_L / \partial \alpha$ and $d_D = \partial k_D / \partial \alpha$, respectively.

17.6 Influence of Control Mechanisms on Wing Motion and Cycle-Averaged Generalized Forces

Wing drive mechanisms for flapping wing aircraft must produce periodic wingbeat motion. Control mechanisms will cause variations in one or more of the seven parameters described above. A general procedure for estimating the influence of changes in the position of control mechanisms on flapping wing motion is to first compute the wing spar position, $\phi(t)$, over one complete wingbeat cycle at the quasi-equilibrium hover condition and then adjust the parameters of Eqs. 17.1 and 17.2 to fit the actual wing motion profile. By adjusting the state of the control mechanism, such that one or more of the seven parameters is altered, and recomputing the fit parameters, an approximation for the partial derivatives of the seven parameters with respect to changes in the control mechanism can be estimated. A simple fitting procedure for the periodic wing root position follows:

1. Determine the period, T , of the wingbeat and compute the symmetric frequency:

Table 17.3 Nonzero cycle-averaged control derivatives evaluated at quasi-equilibrium hover

| Sensitivity | Left wing | Right wing |
|---|--|--|
| $\partial \overline{F_x^B} / \partial \omega$ | $A^2 \omega k_L$ | $A^2 \omega k_L$ |
| $\partial \overline{F_x^B} / \partial A$ | $A \omega^2 k_L$ | $A \omega^2 k_L$ |
| $\partial \overline{F_x^B} / \partial \alpha_u$ | $1/4 A^2 \omega^2 d_L$ | $1/4 A^2 \omega^2 d_L$ |
| $\partial \overline{F_x^B} / \partial \alpha_d$ | $1/4 A^2 \omega^2 d_L$ | $1/4 A^2 \omega^2 d_L$ |
| $\partial \overline{F_z^B} / \partial \delta$ | $-A \omega k_D J_1(A)$ | $-A \omega k_D J_1(A)$ |
| $\partial \overline{F_z^B} / \partial \alpha_u$ | $1/2 A \omega^2 J_1(A) d_D$ | $1/2 A \omega^2 J_1(A) d_D$ |
| $\partial \overline{F_z^B} / \partial \alpha_d$ | $-1/2 A \omega^2 J_1(A) d_D$ | $-1/2 A \omega^2 J_1(A) d_D$ |
| $\partial \overline{F_z^B} / \partial \zeta$ | $-1/2 A^2 \omega^2 k_L$ | $-1/2 A^2 \omega^2 k_L$ |
| $\partial \overline{M_x^B} / \partial \delta$ | $1/2 A \omega k_D (A y_{cp} + J_1(A) w)$ | $-1/2 A \omega k_D (A y_{cp} + J_1(A) w)$ |
| $\partial \overline{M_x^B} / \partial \alpha_u$ | $1/4 A \omega^2 d_D (A y_{cp} + J_1(A) w)$ | $-1/4 A \omega^2 d_D (A y_{cp} + J_1(A) w)$ |
| $\partial \overline{M_x^B} / \partial \alpha_d$ | $-1/4 A \omega^2 d_D (A y_{cp} + J_1(A) w)$ | $1/4 A \omega^2 d_D (A y_{cp} + J_1(A) w)$ |
| $\partial \overline{M_x^B} / \partial \zeta$ | $A \omega^2 k_L (J_1(A) y_{cp} + A w/4)$ | $-A \omega^2 k_L (J_1(A) y_{cp} + A w/4)$ |
| $\partial \overline{M_y^B} / \partial \delta$ | $A \omega J_1(A) (k_D (\Delta x_R^B - x_{cp} \sin(\alpha)) - k_L x_{cp} \cos(\alpha))$ | $A \omega J_1(A) (k_D (\Delta x_R^B - x_{cp} \sin(\alpha)) - k_L x_{cp} \cos(\alpha))$ |
| $\partial \overline{M_y^B} / \partial \eta$ | $A \omega^2 J_1(A) y_{cp} k_L$ | $A \omega^2 J_1(A) y_{cp} k_L$ |
| $\partial \overline{M_y^B} / \partial \alpha_u$ | $-A \omega^2 J_1(A) (\Delta x_R^B d_D + x_{cp} ((d_L + k_D) \cos(\alpha) + (d_D - k_L) \sin(\alpha)))$ | $-A \omega^2 J_1(A) (\Delta x_R^B d_D + x_{cp} ((d_L + k_D) \cos(\alpha) + (d_D - k_L) \sin(\alpha)))$ |
| $\partial \overline{M_y^B} / \partial \alpha_d$ | $A \omega^2 J_1(A) (\Delta x_R^B d_D + x_{cp} ((d_L + k_D) \cos(\alpha) + (d_D - k_L) \sin(\alpha)))$ | $A \omega^2 J_1(A) (\Delta x_R^B d_D + x_{cp} ((d_L + k_D) \cos(\alpha) + (d_D - k_L) \sin(\alpha)))$ |
| $\partial \overline{M_y^B} / \partial \zeta$ | $1/2 A^2 \omega^2 \Delta x_R^B k_L$ | $1/2 A^2 \omega^2 \Delta x_R^B k_L$ |
| $\partial \overline{M_z^B} / \partial \omega$ | $2 A \omega k_L (J_1(A) y_{cp} + A w/4)$ | $-2 A \omega k_L (J_1(A) y_{cp} + A w/4)$ |
| $\partial \overline{M_z^B} / \partial A$ | $A \omega^2 k_L ((J_1(A)/A + (J_0(A) - J_2(A))/2) y_{cp} + w/2)$ | $-A \omega^2 k_L ((J_1(A)/A + (J_0(A) - J_2(A))/2) y_{cp} + w/2)$ |
| $\partial \overline{M_z^B} / \partial \alpha_u$ | $-1/2 A \omega^2 d_L (J_1(A) y_{cp} + A w/4)$ | $1/2 A \omega^2 d_L (J_1(A) y_{cp} + A w/4)$ |
| $\partial \overline{M_z^B} / \partial \alpha_d$ | $-1/2 A \omega^2 d_L (J_1(A) y_{cp} + A w/4)$ | $1/2 A \omega^2 d_L (J_1(A) y_{cp} + A w/4)$ |

$$\omega = \frac{2\pi}{T} \quad (17.25)$$

2. Determine the time at which wing stroke reversal occurs, T_R , and compute the asymmetric frequency:

$$\delta = \omega - \frac{\pi}{T_R} \quad (17.26)$$

3. Compute the stroke amplitude:

$$A = \frac{1}{2} [\phi_{\max} - \phi_{\min}] \quad (17.27)$$

4. Compute the wing stroke bias:

$$\eta = \phi_{\max} - A \quad (17.28)$$

5. Compute the best fit stroke plane angle:

$$\hat{\zeta} = \arg \min_z \int_0^{\frac{2\pi}{\omega}} (\zeta(t) - z)^2 dt \quad (17.29)$$

Note that this computation is trivial if the control mechanism simply tilts the stroke plane at a constant angle.

6. Compute the best fit upstroke angle-of-attack

$$\hat{\alpha}_u = \arg \min_{a_u} \int_0^{\frac{\pi}{\omega-\delta}} (\alpha(t) - a_u)^2 dt \quad (17.30)$$

7. Compute the best fit downstroke angle-of-attack

$$\hat{\alpha}_d = \arg \min_{a_d} \int_{\frac{\pi}{\omega-\delta}}^{\frac{2\pi}{\omega}} (\alpha(t) - a_d)^2 dt \quad (17.31)$$

Approximations to the partial derivatives of the seven parameters with respect to the control inputs can be obtained by finite difference methods. These partial derivatives can be used in conjunction with the cycle-averaged aerodynamic control derivatives presented in Table 17.3 to compute estimates of a control effectiveness matrix for a wide range of control inputs that might be found on flapping wing aircraft. The multivariable chain rule can be used to compute the sensitivities of a cycle-averaged generalized force to changes in a general control input u . Application of the multivariable chain rule yields

$$\frac{\partial \bar{G}}{\partial u} = \frac{\partial \bar{G}}{\partial \omega} \frac{\partial \omega}{\partial u} + \frac{\partial \bar{G}}{\partial \delta} \frac{\partial \delta}{\partial u} + \frac{\partial \bar{G}}{\partial A} \frac{\partial A}{\partial u} + \frac{\partial \bar{G}}{\partial \eta} \frac{\partial \eta}{\partial u} + \frac{\partial \bar{G}}{\partial \zeta} \frac{\partial \zeta}{\partial u} + \frac{\partial \bar{G}}{\partial \alpha_u} \frac{\partial \alpha_u}{\partial u} + \frac{\partial \bar{G}}{\partial \alpha_d} \frac{\partial \alpha_d}{\partial u} \quad (17.32)$$

Equation 17.32 serves as a basis for generating the linear control effectiveness matrix, \mathbf{B} , for a suite of control effectors that change the cycle-averaged forces and moments acting on a vehicle through changes in the wingbeat kinematics. The control effectiveness matrix is

$$\mathbf{B} \triangleq \begin{bmatrix} \partial \bar{F}_x^B / \partial u_1 & \partial \bar{F}_x^B / \partial u_2 & \dots & \partial \bar{F}_x^B / \partial u_m \\ \partial \bar{F}_y^B / \partial u_1 & \partial \bar{F}_y^B / \partial u_2 & \dots & \partial \bar{F}_y^B / \partial u_m \\ \partial \bar{F}_z^B / \partial u_1 & \partial \bar{F}_z^B / \partial u_2 & \dots & \partial \bar{F}_z^B / \partial u_m \\ \partial \bar{M}_x^B / \partial u_1 & \partial \bar{M}_x^B / \partial u_2 & \dots & \partial \bar{M}_x^B / \partial u_m \\ \partial \bar{M}_y^B / \partial u_1 & \partial \bar{M}_y^B / \partial u_2 & \dots & \partial \bar{M}_y^B / \partial u_m \\ \partial \bar{M}_z^B / \partial u_1 & \partial \bar{M}_z^B / \partial u_2 & \dots & \partial \bar{M}_z^B / \partial u_m \end{bmatrix} \quad (17.33)$$

Given a desired cycle-averaged force and moment vector, $[\bar{F}_{x_{des}} \ \bar{F}_{y_{des}} \ \bar{F}_{z_{des}} \ \bar{M}_{x_{des}} \ \bar{M}_{y_{des}} \ \bar{M}_{z_{des}}]^T$, the objective is to find the control vector, $[u_1 \ u_2 \ \dots \ u_m]^T$, such that

$$\begin{bmatrix} \bar{F}_{x_{des}} \\ \bar{F}_{y_{des}} \\ \bar{F}_{z_{des}} \\ \bar{M}_{x_{des}} \\ \bar{M}_{y_{des}} \\ \bar{M}_{z_{des}} \end{bmatrix} = \mathbf{B} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \quad (17.34)$$

Control allocation methods (Oppenheimer et al. 2010c) can be used to solve Eq. 17.34. These methods blend the available effectors to provide the cycle-averaged forces and moments commanded by a standard control law. The control designer must bear in mind that the dynamics of the fuselage must evolve on a much slower timescale than the wingbeats and that one can only influence the cycle-averaged forces and moments rather than the instantaneous periodic forces and moments. Thus, special provisions must be made to ensure that the controller and control allocator do not attempt to compensate for fuselage motion errors occurring within the timescale of the wingbeat. A simple method of addressing this problem is to implement a “cycle zero-order hold” that only allows new control input commands to be passed to the control effectors once per wingbeat cycle (Doman et al. 2010a). The advantage of this method is that it allows one to use any standard method to control a flapping wing aircraft that operates under the influence of periodic rather than steady forces and moments. The steady force and moment commands generated by the control law are interpreted as cycle-averaged force and moment commands, and the control allocation algorithm effectively finds the inverse mapping between those commands and a set of control effector positions that are required to produce them.

17.7 Flapping Wing Aerodynamic Stability Derivatives

Extracting stability derivatives requires that the effects of the fuselage velocity be added to the force and moment computations. Previous works have extended the quasi-steady model presented here to include such effects without great computational cost (Sigthorsson et al. 2012a; Faruque and Humbert 2010). There, the expression for the velocity field, used to compute the dynamic pressure over the surfaces of the wings, includes the rigid body velocities. The instantaneous forces and moments for a particular vehicle configuration can be computed by gridding the surfaces of the wings and applying numerical integration over the surface. The cycle-averaged forces and moments are obtained by numerical integration with respect to time over a full wingbeat cycle. Numerical estimation of partial derivatives with respect to fuselage velocity changes over the wingbeat cycle can then be computed to obtain cycle-averaged stability derivatives, which can be used to form the elements of a system matrix (Sigthorsson et al. 2012b).

In order to account for the vehicle velocity in the aerodynamic force expressions, the computation of the dynamic pressure on the wing and the angle-of-attack is required. These terms are functions of the total velocity of the wing with respect to quiescent air. The total velocity of the wing at a point

$$\mathbf{r}^{wp} \triangleq [-x_{wp} \ y_{wp} \ 0]^T \quad (17.35)$$

on the wing planform is composed of the velocity due to the wing flapping, \mathbf{v}_f ; fuselage translational velocity, \mathbf{v}_b ; and the fuselage rotational velocity, Ω

$$\mathbf{v}^B(\mathbf{r}) = \begin{bmatrix} v_x^B(\mathbf{r}) & v_y^B(\mathbf{r}) & v_z^B(\mathbf{r}) \end{bmatrix}^T \triangleq \mathbf{v}_f^B(\mathbf{r}) + \mathbf{v}_b^B + (\Omega^B \times \mathbf{r}^B) \quad (17.36)$$

Furthermore, the small angle between the local freestream velocity due to flapping and fuselage motion at a point on the wing planform is found to be Sigthorsson et al. (2012a)

$$\alpha_v(\mathbf{r}) = \text{atan} \left(v_z^{wp}(\mathbf{r}), \sqrt{(v_x^{wp}(\mathbf{r}))^2 + (v_y^{wp}(\mathbf{r}))^2} \right) \quad (17.37)$$

The local angle-of-attack, defined as $\alpha(\mathbf{r}) \triangleq |\alpha_v(\mathbf{r})|$, is used to compute the lift and drag coefficients. By definition, the drag is along the local freestream velocity vector and the lift is perpendicular to the drag. Define a frame V with its origin at \mathbf{r} , the drag is along the positive x -axis, the lift lies on the z -axis, and the y -axis lies in the wing planform plane. The infinitesimal aerodynamic force is expressed as

$$d\mathbf{F}^V = \begin{bmatrix} dD \\ 0 \\ \text{sgn}(\alpha_v(\mathbf{r}))dL \end{bmatrix} \quad (17.38)$$

where the infinitesimal lift and drag forces are, respectively,

$$dL = \frac{\rho}{2} \|\mathbf{v}(\mathbf{r})\|^2 C_L(\alpha(\mathbf{r})) dx_{wp} dy_{wp} \quad (17.39)$$

$$dD = \frac{\rho}{2} \|\mathbf{v}(\mathbf{r})\|^2 C_D(\alpha(\mathbf{r})) dx_{wp} dy_{wp} \quad (17.40)$$

and the infinitesimal aerodynamic moment is

$$d\mathbf{M}^B = \mathbf{r}^B \times d\mathbf{F}^B \quad (17.41)$$

where the moment arm associated with a point on the planform, \mathbf{r}^B , can be obtained directly from Table 17.2 by substituting the subscripts cp with wp . Many flapping wing aircraft make use of passive angle-of-attack regulation, where dynamic pressure acting on the wing causes it to rotate to a wing stop, or twist due

to the load. When flapping in still air at hover, the wing rotation is well defined as it takes place at stroke reversal. As the fuselage velocity increases, the wing rotation points become more difficult to compute. To illustrate the difficulty, consider an extreme but impractical case where the fuselage forward velocity is equal to the maximum blade tip velocity of the wing, measured relative to the fuselage. In this case, the dynamic pressure at the blade tip would be zero on the upstroke; thus, the wing would experience little to no passive rotation at midstroke. It is therefore intractable to write a general expression for the total aerodynamic force and center-of-pressure location using a fixed wing rotation stop over each upstroke and downstroke. However, at small velocities the wing rotation can be approximated as constant over the upstroke, α_u , and downstroke, α_d . Using this approximation and, for brevity, dropping the explicit notation of which variables depend on \mathbf{r} and t , the infinitesimal aerodynamic force for the right wing on the upstroke and downstroke, respectively, is expressed as follows:

$$d\mathbf{F}_{\text{RWU}}^B = \frac{\rho \|\mathbf{v}\|}{2} \begin{bmatrix} \cos(\zeta) \sin(\alpha_u) - \cos(\alpha_u) \cos(\phi) \sin(\zeta) & \sin(\zeta) \sin(\phi) & \cos(\alpha_u) \cos(\zeta) + \cos(\phi) \sin(\alpha_u) \sin(\zeta) \\ \cos(\alpha_u) \sin(\phi) & \cos(\phi) & -\sin(\alpha_u) \sin(\phi) \\ -\cos(\alpha_u) \cos(\zeta) \cos(\phi) - \sin(\alpha_u) \sin(\zeta) & \cos(\zeta) \sin(\phi) & \cos(\zeta) \cos(\phi) \sin(\alpha_u) - \cos(\alpha_u) \sin(\zeta) \end{bmatrix} \\ * \left(\mathbf{v}^{wp} C_D(\alpha) + \begin{bmatrix} -v_x^{wp} v_z^{wp} / \sqrt{(v_x^{wp})^2 + (v_y^{wp})^2} \\ -v_y^{wp} v_z^{wp} / \sqrt{(v_x^{wp})^2 + (v_y^{wp})^2} \\ \sqrt{(v_x^{wp})^2 + (v_y^{wp})^2} \end{bmatrix} \text{sgn}(v_z^{wp}) C_L(\alpha) \right) dx_{wp} dy_{wp} \quad (17.42)$$

$$d\mathbf{F}_{\text{RWD}}^B = \frac{\rho \|\mathbf{v}\|}{2} \begin{bmatrix} \cos(\zeta) \sin(\alpha_d) + \cos(\alpha_d) \cos(\phi) \sin(\zeta) & \sin(\zeta) \sin(\phi) & -\cos(\alpha_d) \cos(\zeta) + \cos(\phi) \sin(\alpha_d) \sin(\zeta) \\ -\cos(\alpha_d) \sin(\phi) & \cos(\phi) & -\sin(\alpha_d) \sin(\phi) \\ \cos(\alpha_d) \cos(\zeta) \cos(\phi) - \sin(\alpha_d) \sin(\zeta) & \cos(\zeta) \sin(\phi) & \cos(\zeta) \cos(\phi) \sin(\alpha_d) + \cos(\alpha_d) \sin(\zeta) \end{bmatrix} \\ * \left(\mathbf{v}^{wp} C_D(\alpha) + \begin{bmatrix} -v_x^{wp} v_z^{wp} / \sqrt{(v_x^{wp})^2 + (v_y^{wp})^2} \\ -v_y^{wp} v_z^{wp} / \sqrt{(v_x^{wp})^2 + (v_y^{wp})^2} \\ \sqrt{(v_x^{wp})^2 + (v_y^{wp})^2} \end{bmatrix} \text{sgn}(v_z^{wp}) C_L(\alpha) \right) dx_{wp} dy_{wp} \quad (17.43)$$

The force for the left wing is the same for a symmetric stroke but with a sign change on the second component of the force vector.

After gridding the wing planforms, the instantaneous aerodynamic forces and moments can be computed by numerically integrating the differential forces and moments over the planform area. In order to compute the cycle-averaged forces and moments, the total instantaneous forces and moments can be numerically integrated with respect to time over the period of the wingbeat. Cycle-averaged stability derivatives can be estimated through numerical difference computations applied to the vehicle linear and angular velocities.

17.8 Simulation Methods

Rigid body equations of motion for aircraft can be used as a point of departure for simulation analysis, provided that the mass of the wings is small when compared to the mass of the fuselage (Doman et al. 2010a). The reader is referred to the multibody equations of motion for flapping wing aircraft for cases where this condition is violated (Bolender 2009). It is important to note that the forces and moments that drive these equations are periodic functions of time. The equations of motion for a rigid body aircraft are

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}^{-1} (\mathbf{M}^B(\mathbf{t}) - \boldsymbol{\Omega} \times \mathbf{I} \boldsymbol{\Omega}) \quad (17.44)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} + \frac{1}{m} [\mathbf{F}^B(t)] - \mathbf{R}_I^B \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (17.45)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{R}_B^I \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (17.46)$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (17.47)$$

where \mathbf{I} is the mass moment of inertia matrix; $\boldsymbol{\Omega} = [p \ q \ r]^T$ is the body-axis angular rate vector; $\mathbf{M}^B(t)$ is the sum of the left and right wing instantaneous aerodynamic body-axis moment vectors consisting of the rolling, pitching, and yawing moments; u, v, w are the body-axis translational velocities; $\mathbf{F}^B(t)$ is sum of the left and right wing instantaneous aerodynamic body-axis force vectors; m is the vehicle mass; g is the acceleration due to gravity; x, y, z are the vehicle positions with respect to an inertial frame; and $\mathbf{q} = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$ is a quaternion used to perform the 3-2-1 standard Euler transformation. Also, $\mathbf{R}_B^I = \mathbf{R}_I^{B^T}$ are rotation matrices that transform from body to inertial axes and back.

A simple blade element-based simulation can be constructed using the instantaneous forces and moments that can be computed from Tables 17.1 and 17.2 and Eq. 17.15. Such models account for periodic forces and moments due to fluctuations in dynamic pressure; however, they do not account for unsteady aerodynamic effects. Such simulations can be used to assess the efficacy of a cycle-averaged model-based control law when applied to a vehicle model that includes time periodic variations in the forces and moments due to the flapping wings without unsteady aerodynamic or structural flexibility effects. Directly modeling such effects can consume significant computational resources, which can produce a simulation that

is impractical for the study of the behavior of control systems for flapping wing aircraft. It is recommended that such effects be captured in tabular form based upon the results of either high-fidelity computational methods or experiments, e.g., the low Reynolds number experiments of Sane and Dickinson (2001). Such results can be used to produce instantaneous lift and drag coefficients as a function of stroke angle, angle-of-attack, amplitude, frequency, and asymmetric frequency. This more complete representation may include translational and rotational aerodynamic contributions, unsteady aerodynamic effects, and fluid structure interactions. For each set of parameters considered in a table, the lift and drag coefficients should be estimated for a given stroke angle and angle-of-attack. Instantaneous estimates of the aerodynamic forces produced in this way should be representative of the steady-state periodic forces produced by the wings. In other words, transients due to changes in parameters other than stroke angle and angle-of-attack would not appear in such a model. Whether a simple blade element model or a higher-fidelity table-lookup is used, both simulations will provide the designer with a model that can be used to assess the performance of cycle-averaged model-based control laws on aircraft models that are driven by periodic forces and moments.

17.9 Conclusion

The methods presented in this chapter are intended for use with hover-capable flapping wing vehicles in the conceptual design phase. The methods allow engineers to iterate between vehicle design, trim, and controllability analysis until closure is achieved. Prototype model-based controllers can be designed by using the control derivative tables, the multivariable chain rule, and conventional control design methods. The resulting control laws provide a point of departure design that can be tested on the prototype aircraft or in higher-fidelity simulations. Tuning and optimization of the control law will likely be required to compensate for the effects of unmodeled unsteady aerodynamic effects, fluid-structure interactions, as well as interactions between wing loads, drive mechanism loads, and prime movers.

References

- M.L. Anderson, Design and control of flapping wing micro air vehicles. Ph.D. thesis, Air Force Institute of Technology, (2011)
- M.A. Bolender, Rigid multi-body equations of motion for flapping wing micro air vehicles using kane's equations, in *AIAA Guidance, Navigation, and Control Conference*, Chicago, (2009). AIAA Paper 2010-6158
- X. Deng, L. Schenato, W.C. Wu, S.S. Sastry, Flapping flight for biomimetic robotic insects: part I – system modeling. *IEEE Trans. Robot.* **22**(4), 776–788 (2006a)
- X. Deng, L. Schenato, S.S. Sastry, Flapping flight for biomimetic robotic insects: part II – flight control design. *IEEE Trans. Robot.* **22**(4), 789–803 (2006b)
- D.B. Doman, S. Regisford, Wing sizing, trim, and control consideration in the design of hover-capable flapping wing micro air vehicles, in *AIAA Atmospheric Flight Mechanics Conference*, Toronto, (2010). AIAA Paper 2010-7629

- D.B. Doman, M.W. Oppenheimer, D.O. Sigthorsson, Wingbeat shape modulation for flapping-wing micro-air-vehicle control during hover. *J. Guid. Control Dyn.* **33**(3), 724–739 (2010a)
- D.B. Doman, M.W. Oppenheimer, D.O. Sigthorsson, Dynamics and control of a biomimetic vehicle using biased wingbeat forcing functions: part i – aerodynamic model, in *48th AIAA Aerospace Sciences Meeting*, Orlando, (2010b). AIAA Paper 2010–1023
- D.B. Doman, C.P. Tang, S. Regisford, Modeling interactions between flexible flapping-wing spars, mechanisms, and drive motors. *J. Guid. Control Dyn.* **34**(5), 1457–1473 (2011)
- I. Faruque, J.S. Humbert, Dipterian insect flight dynamics. Part 1 longitudinal motion about hover. *J. Theor. Biol.* **264**(2), 538–552 (2010)
- M.V. Ol, K. Granlund, Abstraction of aerodynamics of flapping-wings: is it quasi-steady? in *50th AIAA Aerospace Sciences Meeting*, Nashville, (2012). AIAA Paper 2012–0587
- M.W. Oppenheimer, M.A. Bolender, D.O. Sigthorsson, D.B. Doman, Analysis of the translation motion of a flapping wing micro air vehicle, in *AIAA Guidance, Navigation and Control Conference*, Toronto, (2010a). AIAA Paper 2010–7708
- M.W. Oppenheimer, D.B. Doman, D.O. Sigthorsson, Dynamics and control of a biomimetic vehicle using biased wingbeat forcing functions: part II – controller, in *48th AIAA Aerospace Sciences Meeting*, Orlando, (2010b). AIAA Paper 2010–1024
- M.W. Oppenheimer, D.B. Doman, M.A. Bolender, Control allocation, in *Control System Application*, ed. by W.S. Levine. Control Handbook, vol. II, 2nd edn. (CRC, Boca Raton, 2010c)
- M.W. Oppenheimer, D.B. Doman, D.O. Sigthorsson, Dynamics and control of a hovering biomimetic vehicle using biased wingbeat forcing functions. *J. Guid. Control Dyn.* **34**(4), 647–662 (2011a)
- M.W. Oppenheimer, D.O. Sigthorsson, D.B. Doman, Body torque generation for a flapping wing micro air vehicle by angle of attack change, in *49th AIAA Aerospace Sciences Meeting*, Orlando, (2011b). AIAA Paper 2011–1281
- S.P. Sane, M.H. Dickinson, The control of flight force by a flapping wing: lift and drag force production. *J. Exp. Biol.* **204**, 2607–2626 (2001)
- D.O. Sigthorsson, M. Oppenheimer, D. Doman, Flapping wing micro air vehicle aerodynamic modeling including flapping and rigid body velocity, in *50th AIAA Aerospace Sciences Meeting*, Nashville, (2012a). AIAA-Paper-2012–0026
- D.O. Sigthorsson, M. Oppenheimer, D. Doman, Insect sized flapping wing vehicles versus rotorcrafts, a comparative study, in *50th AIAA Aerospace Sciences Meeting*, Nashville, TN.: (2012b). AIAA-Paper-2012–0028
- B.K. Stanford, P.S. Beran, Analytical sensitivity analysis of an unsteady vortex method for flapping-wing optimization. *J. Aircr.* **47**(3), 647–662 (2010)
- R.J. Wood, The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Trans. Robot.* **24**(2), 341–347 (2008)

Gabriel Hugh Elkaim, Fidelis Adhika Pradipta Lie, and
Demos Gebre-Egziabher

Contents

| | | |
|--------|---|-----|
| 18.1 | Introduction | 348 |
| 18.2 | Attitude and Position Estimation | 351 |
| 18.2.1 | INS/GPS Integrated Navigation System | 352 |
| 18.2.2 | Implementation and Practical Challenges | 357 |
| 18.2.3 | Flight Test Result | 359 |
| 18.3 | Inner Loop Control | 359 |
| 18.3.1 | PID Control | 361 |
| 18.3.2 | Lateral Control | 363 |
| 18.3.3 | Longitudinal Control | 364 |
| 18.3.4 | Trim Conditions | 367 |
| 18.3.5 | Simulation and Flight Test Results | 367 |
| 18.4 | Guidance | 369 |
| 18.4.1 | General Tracking | 370 |
| 18.4.2 | Straight Line Tracking | 371 |
| 18.4.3 | L_2^+ Control | 372 |
| 18.4.4 | Waypoint Switching | 374 |
| 18.4.5 | Point Acquisition and RTB | 376 |
| 18.4.6 | Simulation and Flight Test Results | 377 |
| 18.5 | Conclusion | 378 |
| | References | 379 |

G.H. Elkaim (✉)

Computer Engineering Department, University of California Santa Cruz, Santa Cruz, CA, USA
e-mail: elkaim@soe.ucsc.edu

F.A. Pradipta Lie • D. Gebre-Egziabher

Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA
e-mail: adhika@aem.umn.edu; gebre@aem.umn.edu

Abstract

Two complete system architectures for a guidance, navigation, and control solution of small UAVs are presented. These systems (developed at the University of California Santa Cruz and the University of Minnesota) are easily reconfigurable and are intended to support test beds used in navigation, guidance, and control research. The systems described both integrate a low-cost inertial measurement unit, a GPS receiver, and a triad of magnetometers to generate a navigation solution (position, velocity, and attitude estimation) which, in turn, is used in the guidance and control algorithms. The navigation solution described is a 15-state extended Kalman filter which integrates the inertial sensor and GPS measurement to generate a high-bandwidth estimate of a UAV's state. Guidance algorithms for generating a flight trajectory based on waypoint definitions are also described. A PID controller which uses the navigation filter estimate and guidance algorithm to track a flight trajectory is detailed. The full system architecture – the hardware, software, and algorithms – is included for completeness. Hardware in the loop simulation and flight test results documenting the performance of these two systems is given.

18.1 Introduction

In current usage, the term uninhabited aerial vehicles or UAVs refers to aircraft which fly without a human operator onboard. They span a wide range in size and complexity. The largest UAVs such as Predator or Global Hawk can weigh several thousand pounds and have wingspans on the order of 10 to 100 ft. At the other end of the size spectrum are UAVs whose maximum dimensions and mass are on the order of centimeters and grams, respectively. In general, these vehicles are being used or envisioned for use in operations where they serve primarily as a platform for a sensor payload. The sensor payload can be as simple as an electro-optical camera or as complex as synthetic aperture radar used for remote sensing.

Regardless of their size and mission, all UAVs share the need for sensor or sensor systems which provide an estimate of the vehicle's full state vector. The state vector normally consists of three position coordinates, three components of the velocity vector, and anywhere between three and nine parameters which describe the vehicle's attitude. In addition to state sensing and estimation, UAVs (especially one that operates autonomously) need control and guidance systems which allow them to maneuver in a way consistent with their mission. In simple terms, the guidance system generates instruction on what state trajectory the UAV should follow in accomplishing its mission. The control system in turn operates the aircraft controls (aileron, elevators, thrust, etc.) to follow the trajectory generated by the guidance systems. A high-level depiction of a GNC system is shown in Fig. 18.1 and consists of both airborne and ground components.

Off-the-shelf guidance, navigation, and control (GNC) solutions for small UAVs (i.e., UAVs which fall in the so-called "Class I" category of vehicles as defined in UAS CoE 2010) exist. Some of these off-the-shelf solutions consist of the

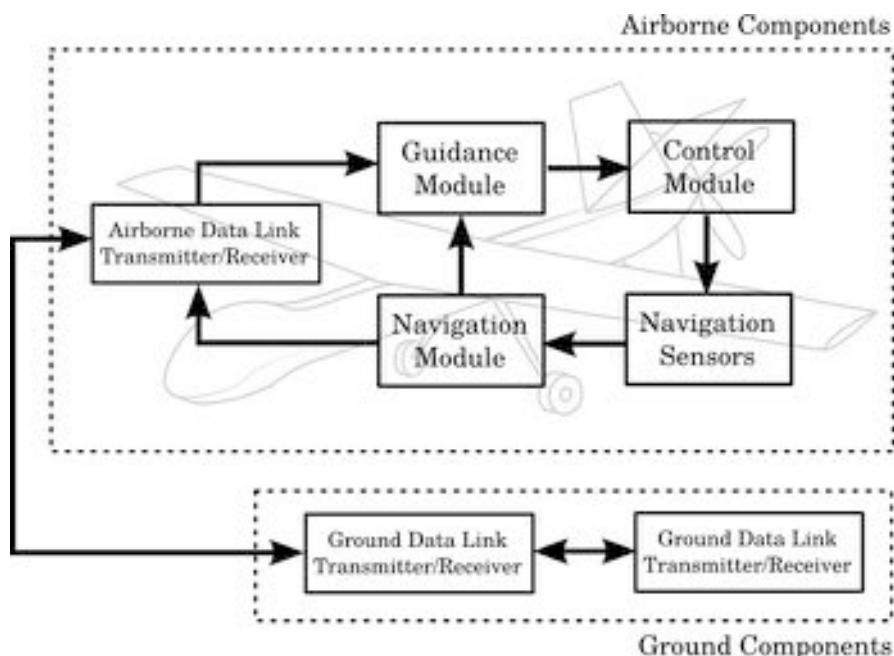


Fig. 18.1 Unmanned aerial system (UAS) concept of operation

vehicle itself, avionics, and associated ground support systems. While most of these “turnkey” solutions allow some level of customization by the user, they can be very restrictive for use in research environments. For example, in research on the robust control problem, the effect of uncertainties and potential fault modes in the guidance and navigation algorithms must be known. Very few off-the-shelf solutions allow access to the internals of the GNC solution. Thus, a GNC solution that is inexpensive and easily reconfigurable is desirable for the research community.

This chapter describes two such GNC solutions for small UAVs. This chapter discusses the implementation aspect of the GNC modules on University of California Santa Cruz’s (UCSC) and University of Minnesota’s (UMN) UAVs. The UAV laboratory at UMN is equipped with two fixed-wing aircraft models: Ultrastick 25e and Ultrastick 120. Shown in, Fig. 18.2 is the Ultrastick 120, also known as FASER (Free-flying for Subscale Experimental Research). FASER is equipped with conventional flight control surfaces and a pair of wingtip vanes on each side of the aircraft to measure aerodynamic angles (angle-of-attack and sideslip angle). Prior to joining the UAV Laboratory, FASER’s airframe was formerly used by NASA for their research program (Owens et al. 2006). Extensive wind tunnel testing during its time at NASAs has been used to develop a comprehensive aerodynamic model of the aircraft. At University of Minnesota, FASER is the workhorse for the multi-sensor navigation research such as GPS attitude and heading determination system, synthetic airdata estimator (Lie and Gebre-Egziabher 2012), and vision-aided navigation (Chu et al. 2011). Thor, an Ultrastick 25e, is an approximately



Fig. 18.2 Free-flying for Subscale Experimental Research (FASER) UAV (a modified Ultra-stick 120)



Fig. 18.3 The UCSC SLUGS autopilot (a) and Multex Mentor UAV (b)

65 % scaled model of FASER with the same basic configuration. As an addition to the system identification work (Dorobantu et al. 2011), this UAV is used for fault-tolerant control research at the University. The UAV Laboratory Web site (Murch 2012b) provides a detailed information on the airframe, avionics, and software architecture. The flight software is available as an open source software package available upon request.

The UCSC autopilot, Santa Cruz Low-Cost UAV GNC Subsystem (SLUGS), has been developed over the past 5 years in order to implement a rapidly reconfigurable autopilot for UAV guidance, navigation, and control research. The SLUGS, pictured in Fig. 18.3a, consists of two fast dsPIC33 microcontrollers (DSC's) and a suite of sensors; the SLUGS-based electric UAV is pictured in Fig. 18.3b. Details of the design, development, and deployment of the SLUGS can be found in Lizarraga (2009) and Lizarraga et al. (2011a,b, 2009a,b).

Section 18.2 describes the UMN's navigation module for attitude and position estimation. Sections 18.3 and 18.4 describe the inner loop control and guidance algorithm implemented on the UCSC SLUGS platform, respectively, including simulation and experimental results.

18.2 Attitude and Position Estimation

In order to be able to guide and control the UAV, the state of the UAV must be available to the controller at high fidelity and high bandwidth. Accurate position is required to perform automatic control for precision applications (e.g., landing). The navigation module in the GNC system implements aircraft state estimation. The advent of powerful computers made available at relatively low cost has allowed sensor fusion for navigation; sensor fusion is a technique of optimally blending information from multiple different (flawed) sensors. Multi-sensor navigation framework aims to obtain the most information about the aircraft states by using minimum combination of sensors. This is key to UAV operations where size, weight, and power are all critical. This section will describe the navigation module implemented on University of Minnesota's UAV.

The complete state of the UAV comprises its position, velocity, attitude, airspeed, angle-of-attack, sideslip angle, and rotation (pitch, roll, and yaw) rates. Position, velocity, and attitude are also known as the navigation state (Gleason and Gebre-Egziabher 2009). The sensors used to measure these quantities are called navigation sensors: an inertial measurement unit (IMU) and global positioning system (GPS) receiver. Figure 18.4a shows the Analog Devices IMU ADIS16405 installed on the UAVs at University of Minnesota. It is a 6 degree-of-freedom temperature calibrated inertial measurement unit with 3-axis accelerometers, 3-axis gyros, and 3-axis magnetometer. With a flight computer that computes the navigation state only from IMU measurements, it is known as an inertial navigation system (INS). Figure 18.4b shows Crescent, an OEM GPS receiver from Hemisphere GPS. In addition to the differentially corrected position and velocity estimates, Crescent also outputs raw pseudorange and carrier phase measurement. These features enable a great deal of navigation research at the UMN UAV Laboratory. Crescent's small form-factor and low power consumption also makes it a very suitable choice for UAV applications.

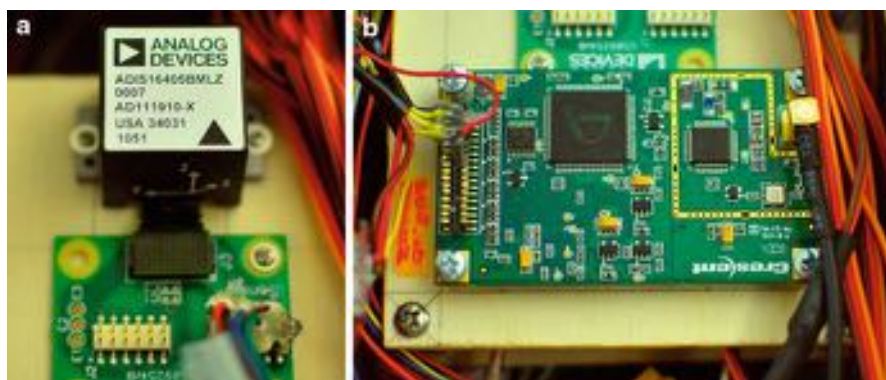


Fig. 18.4 Navigation sensor of UMN UAV: ADIS16405 (a) and Hemisphere Crescent OEM GPS receiver (b)

Fig. 18.5 UMN's Thor instrumented with five-hole pitot tube from Goodrich to measure airspeed, α , and β



Airspeed, angle-of-attack (α), and sideslip angle (β) are known as the airdata quantities, which are traditionally measured using airdata sensors such as pitot tube and wind vane. Another alternative for measuring these angles is to use a multiple hole pitot tube such as the 5-hole pitot tube as shown in Fig. 18.5. Recent work (Lie and Gebre-Egziabher 2012) has shown that this airdata can also be estimated by using sensor fusion from the IMU and GPS.

The navigation state is of particular interest. INS/GPS integration has been studied extensively in the past decades to estimate aircraft's navigation state. This section will focus on the implementation aspects of an integrated navigation systems using inertial sensors and GPS (INS/GPS system). A brief description of INS/GPS is provided; a more in-depth discussion can be found in Gleason and Gebre-Egziabher (2009), Groves (2008), Farrell and Barth (1999), and Titterton and Weston (2004). System architecture and implementation challenges are discussed next. Following these descriptions, FASER's flight test result is presented.

18.2.1 INS/GPS Integrated Navigation System

An inertial navigation system (INS) uses the output of inertial sensors to estimate the vehicle's position, velocity, and attitude. A complete six degree-of-freedom inertial sensor consists of 3-axis accelerometers and 3-axis gyros. The accelerometer measures the specific force acting on the platform and the gyros measures its rotation. Inertial sensors can be categorized according to its resulting navigational accuracy (Gleason and Gebre-Egziabher 2009; Gebre-Egziabher 2001). Automotive-grade micro electro-mechanical system (MEMS) inertial sensors are most suitable for low-cost UAV applications; however, when operating as a stand-alone navigator, these sensors produce positioning errors on the order of several hundreds of meter per minute. These large errors in the position, velocity, and attitude estimates are mainly due to sensor bias and noise that corrupt the measurements. They are also a

function of the initial error on the state estimates. The position error grows linearly with the initial velocity error estimate, quadratically with uncorrected bias, and at a cubic rate with attitude error (Titterton and Weston 2004). Despite the unbounded growth of the error with time, one of the important advantages of inertial navigators is that they are self-contained; that is, they require no external signal to provide a navigation solution. In terms of the data rate, the navigation states can be computed at a rate limited only by the processing power of the flight computer.

Contrast GPS to the advantages and disadvantages of an inertial navigator: GPS requires an external signal to operate, and although the navigation solution accuracy depends on the signal quality and the geometry of the satellite in view, this error is not a function of time (Misra and Enge 2001). Although there are specialized GPS receivers that can generate data up to 100 Hz, most low-cost GPS receivers provide output data at 1–10 Hz. In other words, although GPS navigation solution has long-term stability, the bandwidth of the solution is much lower than that of an inertial navigator.

Integration of INS with GPS allows a navigation solution that has the high bandwidth of the inertial sensors and the drift-free long-term stability of the GPS solution. Attitude determination is an integral part of strapdown INS. This is because the specific force measured by the accelerometer needs to be transformed into the navigation frame in which the position and velocity are calculated. This transformation calls for the knowledge of the orientation of the platform (defined as the attitude).

Attitude can be equivalently described by a set of three angles known as the Euler angle sequence or the four-parameter attitude quaternion. An Euler angle sequence consists of the yaw (ψ), pitch (θ), and roll (ϕ) angles that describe three successive rotations about the body z , y , and x axes, respectively. Although this representation carries physical interpretation, it is singular at $\theta = \pm 90^\circ$. The attitude quaternion is a set of four numbers that can be related to the roll, pitch, and yaw angles using the following relationship:

$$\begin{aligned}\phi &= \arctan\left(\frac{2q_2q_3 + 2q_0q_1}{2q_0^2 + 2q_3^2 - 1}\right) \\ \theta &= \arcsin(-2q_1q_3 + 2q_0q_2) \\ \psi &= \arctan\left(\frac{2q_2q_3 + 2q_0q_1}{2q_0^2 + 2q_3^2 - 1}\right).\end{aligned}\tag{18.1}$$

The quaternion $\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]$ represents the transformation from the local North-East-Down coordinate to the body-fixed frame. Although it has no singularity, directly visualizing the aircraft's orientation can be challenging. Hence, for remote pilot display purposes and most control algorithms, the quaternion attitude needs to be transformed into its corresponding roll, pitch, and yaw angles. For applications where only benign maneuvers are expected, the Euler angle representation can instead be used.

Attitude determination using gyro's output calls for integrating the angular velocity to propagate the attitude forward in time. Since gyros measure inertial rotation, they must be compensated to account for both the earth's rotation rate and the transport rate (Groves 2008) due to the Earth's curvature. For most low-cost UAV applications, these terms are small ($\sim 10^{-5}$ rad/sec) compared to the noise level in the sensors. Hence, they can be neglected for all practical purposes. When calculating attitude at the IMU sample rate, the following equations can be used: for an Euler angle attitude representation:

$$\begin{bmatrix} \phi[k+1] \\ \theta[k+1] \\ \psi[k+1] \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi[k]) \tan(\theta[k]) & \cos(\phi[k]) \tan(\theta[k]) \\ 0 & \cos(\phi[k]) & -\sin(\phi[k]) \\ 0 & \sin(\phi[k]) \sec(\theta[k]) & \cos(\phi[k]) \sec(\theta[k]) \end{bmatrix} \begin{bmatrix} \omega_x^B[k] \Delta t \\ \omega_y^B[k] \Delta t \\ \omega_z^B[k] \Delta t \end{bmatrix} \quad (18.2)$$

and for a quaternion attitude representation:

$$\mathbf{q}[k+1] = \mathbf{q}[k] \otimes \left[1 \quad \frac{1}{2} \omega_x[k]^B \Delta t \quad \frac{1}{2} \omega_y[k]^B \Delta t \quad \frac{1}{2} \omega_z[k]^B \Delta t \right] \quad (18.3)$$

where \otimes operator is the quaternion multiplication operator. Details on quaternion algebra can be found on Lefferts et al. (1982).

In order to improve the navigation solution between subsequent GPS solutions, the filter makes frequent corrections to compensate for the inertial sensor errors. Although more sophisticated sensor error models exist, a simplified model presented in Gebre-Egziabher (2001) is used in this implementation. In this model, the sensor output (e.g., gyro) as a function of time can be written as

$$\omega(t) = \tilde{\omega}(t) + b_{gs} + b_{gd}(t) + w_g \quad (18.4)$$

where $\tilde{\omega}(t)$ represents the true angular velocity, b_{gs} is the turn-on bias, $b_{gd}(t)$ is a time-varying bias, and w_g is measurement noise that can be regarded as white noise. $b_{gd}(t)$ is modeled as first-order Gauss-Markov process, written as

$$\dot{b}_{gd}(t) = -\frac{1}{\tau} b_{gd}(t) + w_b. \quad (18.5)$$

This model is robust to parameters that are unobservable when the UAV is not accelerating (Gleason and Gebre-Egziabher 2009). Using this model, the estimated sensor bias is not simply the true bias corrupting the measurement; it also accounts for all unmodeled errors that corrupt the sensor measurement.

The architecture of the INS/GPS filter using an extended Kalman filter (EKF) (Simon 2006) is shown on Fig. 18.6. Included in the state vector (Euler angle representation) are:

$$\mathbf{x} = \left[\underbrace{L \ \Lambda \ h}_{\text{Position}} \quad \underbrace{V_{\text{North}} \ V_{\text{East}} \ V_{\text{Down}}}_{\text{Groundspeed}} \quad \underbrace{\phi \ \theta \ \psi}_{\text{Attitude}} \quad \underbrace{b_{ax} \ b_{ay} \ b_{az}}_{\text{AccelBias}} \quad \underbrace{b_{gx} \ b_{gy} \ b_{gz}}_{\text{GyroBias}} \right]. \quad (18.6)$$

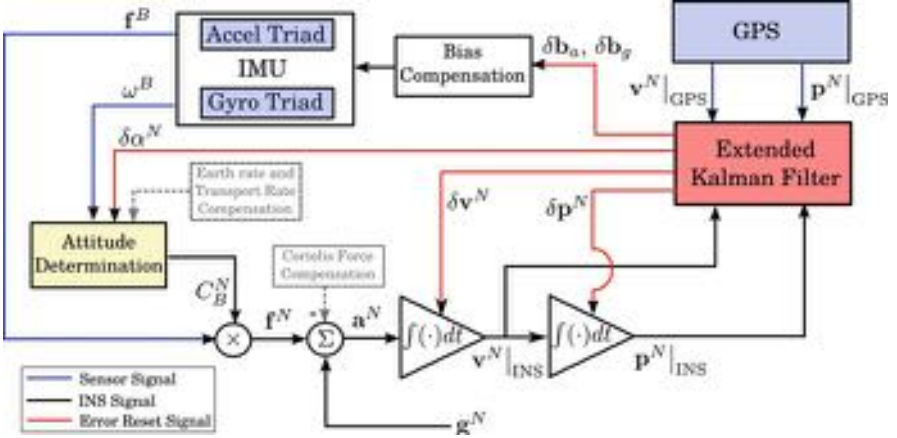


Fig. 18.6 Block diagram of INS/GPS integration

The time update process, executed at the internal IMU sample rate, is:

$$[\phi[k+1] \ \theta[k+1] \ \psi[k+1]]^T = f(\phi[k], \theta[k], \psi[k]) \cdot \omega^B[k-1] \cdot \Delta t \quad (18.7)$$

$$\mathbf{V}^N[k+1] = \mathbf{V}^N[k] + C_B^N[k] \mathbf{f}^B[k] \cdot \Delta t \quad (18.8)$$

$$h[k+1] = h[k] - V_{\text{Down}[k]} \cdot \Delta t$$

$$\Lambda[k+1] = \Lambda[k] + \frac{V_{\text{East}[k]}}{R_E + h[k]} \cdot \Delta t \quad (18.9)$$

$$L[k+1] = L[k] + \frac{V_{\text{North}[k]}}{R_E + h[k]} \cdot \Delta t$$

The covariance is updated according to

$$\mathbf{P}^{(-)}[k+1] = \Phi[k] \mathbf{P}^{(+)}[k] \Phi[k]^T + \mathbf{Q}[k]. \quad (18.10)$$

The state transition matrix can be approximated as $\Phi[k] = \mathbf{I} + \mathbf{F}[k] \cdot \Delta t$. \mathbf{F} is the Jacobian of the time-update equations and \mathbf{Q} is the process noise covariance matrix. Details on the elements of \mathbf{F} and \mathbf{Q} can be found in Gleason and Gebre-Egziabher (2009).

When GPS position and velocity becomes available at time-step k , the measurement update process is outlined as follows. The measurement, \mathbf{y} , is defined as follows:

$$\mathbf{y}|_{\text{GPS}} = [p_N \ p_E \ p_D \ V_{\text{North}} \ V_{\text{East}} \ V_{\text{Down}}]|_{\text{GPS}} \quad (18.11)$$

where p_N , p_E , and p_D are the reported North-East-Down position with respect to a specific location, e.g., the initial position.

Define the state error as:

$$\delta \mathbf{y} = \mathbf{y}|_{\text{GPS}} - [p_N \ p_E \ p_D \ V_{\text{North}} \ V_{\text{East}} \ V_{\text{Down}}] \Big|_{\text{INS}[k]}. \quad (18.12)$$

The covariance is updated using

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 9} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 9} \end{bmatrix} \\ \mathbf{K}[k] &= \mathbf{P}^{(-)}[k] \mathbf{H}^T (\mathbf{R} + \mathbf{H} \mathbf{P}^{(-)}[k] \mathbf{H}^T)^{-1} \\ \mathbf{P}^{(+)}[k] &= (\mathbf{I} - \mathbf{K}[k] \mathbf{H}) \mathbf{P}^{(+)}[k] (\mathbf{I} - \mathbf{K}[k] \mathbf{H})^T + \mathbf{K}[k] \mathbf{R}[k] \mathbf{K}[k]^T. \end{aligned} \quad (18.13)$$

The state is updated according to:

$$\delta \mathbf{x} = \mathbf{K}[k] \delta \mathbf{y} \quad (18.14)$$

Position:

$$\begin{aligned} h^{(+)}[k] &= h^{(-)}[k] - \delta \mathbf{x}(3) \\ \Lambda^{(+)}[k] &= \Lambda^{(-)}[k] + \frac{\delta \mathbf{x}(2)}{R_E + h^{(+)}[k]} \\ L^{(+)}[k] &= L^{(-)}[k] + \frac{\delta \mathbf{x}(1)}{R_E + h^{(+)}[k]}. \end{aligned} \quad (18.15)$$

Velocity:

$$\begin{aligned} V_{\text{North}}^{(+)}[k] &= V_{\text{North}}^{(-)}[k] + \delta \mathbf{x}(4) \\ V_{\text{East}}^{(+)}[k] &= V_{\text{East}}^{(-)}[k] + \delta \mathbf{x}(5) \\ V_{\text{Down}}^{(+)}[k] &= V_{\text{Down}}^{(-)}[k] + \delta \mathbf{x}(6). \end{aligned} \quad (18.16)$$

When using Euler angle representation, the attitude is updated using the transformation matrix C_B^N .

$$C_B^N(+) = (\mathbf{I} - [\delta \mathbf{x}(7 : 9) \times]) \cdot C_B^N(-) \quad (18.17)$$

where $[\cdot \times]$ indicates the skew-symmetric matrix operator. Euler angles can be derived from the updated transformation matrix, $C_B^N(+)$. Accelerometer and gyros bias are updated as follows:

$$\mathbf{b}_a^{(+)}[k] = \mathbf{b}_a^{(-)}[k] + \delta \mathbf{x}(10 : 12). \quad (18.18)$$

$$\mathbf{b}_g^{(+)}[k] = \mathbf{b}_g^{(-)}[k] + \delta \mathbf{x}(13 : 15). \quad (18.19)$$

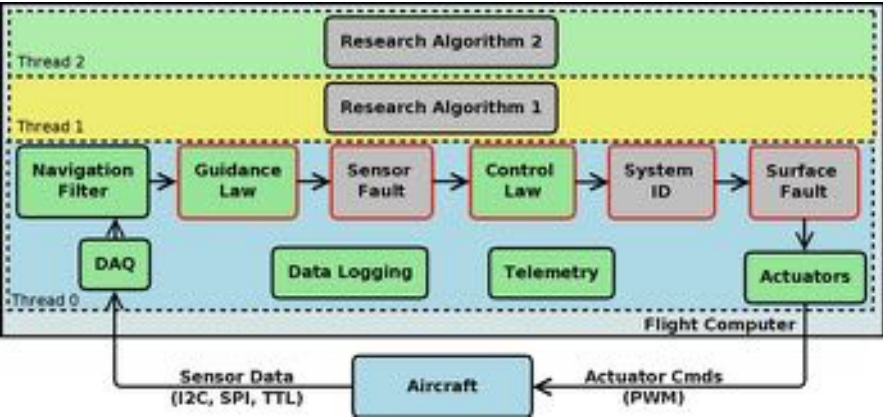


Fig. 18.7 Flight software architecture

18.2.2 Implementation and Practical Challenges

The avionics suite in the UMN’s UAV is implemented on a 32-bit PowerPC Phytec MPC5200B-tiny SoM. It utilizes a real-time operating system (RTOS) and flight software written in C. This computer handles data acquisition; performs guidance, navigation, and control tasks; stores relevant data; and sends information to the ground control station via the telemetry modem. The flight software uses a multi-threaded architecture in which all of the flight critical tasks execute in the highest priority thread at 50 Hz, while additional tasks (i.e., those not required to control the aircraft, such as a fault detection filter) are executed in separate, lower-priority threads (Murch 2012a). As shown in Fig. 18.7, there are ten modules executed in the highest-priority thread (thread 0). Navigation tasks are executed immediately after data acquisition (DAQ). It starts as valid GPS measurements are available. The position and velocity are initialized at the reported position and velocity. Since this initialization occurs on the ground, attitude is initialized to a known attitude of the aircraft on the ground.

One of the key aspects to the correct fusion of inertial sensors and GPS information is aligning data acquisition from non-timing sensors, such as an IMU, with sensors that have inherent timing capability such as GPS. Data acquisition to an IMU is achieved by polling the IMU at the desired sample rate. GPS receivers, however, stream out their navigation solutions (position and velocity) at a preset rate (e.g., 1 Hz). This rate is usually driven by a receiver clock that has been steered to the GPS time. Despite being precise, the flight computer clock might be running at a different rate from that of the GPS clock. When the IMU data stream is not aligned with the GPS output, incorrect measurements are fed to the EKF with resulting errors in the vehicle state estimates. The magnitude of the error depends on the acceleration experienced by the aircraft and the clock rate difference.

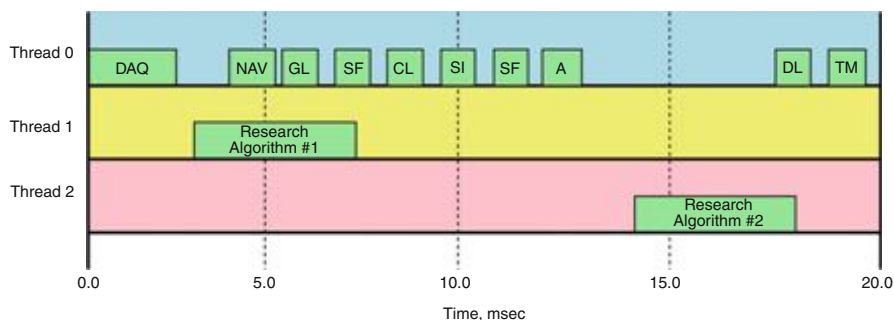


Fig. 18.8 Flight software timing diagram

There are many ways to synchronize IMU and GPS outputs. One way to do this is by performing data acquisition at a deterministic sample rate. By checking the availability of GPS solution at the highest rate available in the flight computer, IMU and GPS data are always synchronized within the resolution of one sampling interval. In this architecture, this is achieved by using the alarm functions provided by the RTOS kernel which are used to trigger module execution. The timing diagram that shows the execution time of each module is shown in Fig. 18.8.

Tuning the extended Kalman filter is known to improve filter convergence time and short-term accuracy (Gleason and Gebre-Egziabher 2009). However, in order to tune the filter, a truth reference is needed so that a comparison can be made to the resulting filter estimate. Although tuning the filter is an art, experience can help smooth the process. Having a truth reference system might be prohibitive for most small scale UAVs; it is recommended that the data set used for tuning is from a known trajectory. This helps to develop a feeling for how the errors grow and how effective is the bias compensations. Since the states' observability depends highly on the acceleration experienced by the vehicle, it is important that a trajectory with sufficient acceleration is used: acceleration and deceleration on a straight line path and frequent turns should be included. It might also be worth testing the filter performance both during powered and gliding operation since structural vibration might adversely affect the filter performance. Adding damping to the system or simply moving the IMU placement on the UAV can often alleviate vibration problems.

Lastly, when the IMU is not collocated with the GPS antenna, the distance between the IMU and the GPS antenna results in different velocity and position measured by both sensors. This is known as the lever arm problem, and the Kalman filter must account for it to make correct state estimates. There are generally two ways to account for this error. The first way is to transfer the GPS measurement to the location of the IMU by using the information of the aircraft attitude and the known lever arm vector in the body frame. The dependency on estimated attitude to transfer this measurement might lead to inaccurate GPS position and velocity estimates. The second way is by inflating the GPS measurement covariance. This method fails when the inflation factor is so large that it renders the GPS position

and velocity useless. For most UAV applications, however, the distance between the GPS antenna and the IMU is not large, thus only a very small inflation required to compensate for this error. In this work, the latter method is employed.

18.2.3 Flight Test Result

Figure 18.9 shows the trajectory flown during one of FASER's flight test. The navigation data are extracted from the flight computer's data recorder and are plotted on Fig. 18.10. During the flight, there were several GPS outages, and the navigation state estimates during these period of free inertials are plotted in green. When bias corrections are continuously fed to the IMU, this inflight calibration improves the performance of the inertial navigation and slows down error growth during these GPS outages.

In Fig. 18.10a, b, the position and velocity estimates from INS/GPS integration system are differenced with the velocity from the GPS measurement. Due to lack of a reference (truth) system, it is not possible to compare the attitude and bias estimates. Hence, the attitude and bias estimates from the INS/GPS filter are plotted on Fig. 18.10c, d. Qualitative evaluation of the filter performance indicates good performance. This can be seen from the smoothness of the INS/GPS solution and good convergence of the inertial sensor bias estimates. As such, the INS/GPS integration provides a high-bandwidth solution with long-term stability suitable for UAV control.

18.3 Inner Loop Control

In order to fly an aircraft, a low-level control system must stabilize the airframe using available sensor inputs and actuators. A higher-level outer loop control will implement path following (see Sect. 18.4) while the inner loop keeps the aircraft flying.

There are myriad ways to implement an inner loop control on a UAV. In this chapter the inner loop controller is based on the SLUGS autopilot (details at Lizarraga (2009), ASL, and SLU), which uses relatively simple PID controllers in the inner loop. The PID-based inner loop control that has flown on the SLUGS platform is presented, and actual flight data of the implementation is included.

The UCSC SLUGS autopilot is divided into two hardware sections: a control processor and a sensor processor. The sensor processor is tasked with taking the raw sensor measurements and fusing them into a high-quality position and attitude estimate (similar to Sect. 18.2). The control processor is tasked with both the inner loop (stabilization) and the outer loop (guidance). The SLUGS implementation is illustrated in Fig. 18.11.

While there are many different ways to implement a low-level control system, the one presented is based on decoupling the longitudinal and lateral flight dynamics into two separate control systems (Stewart 2001). The lateral control channel

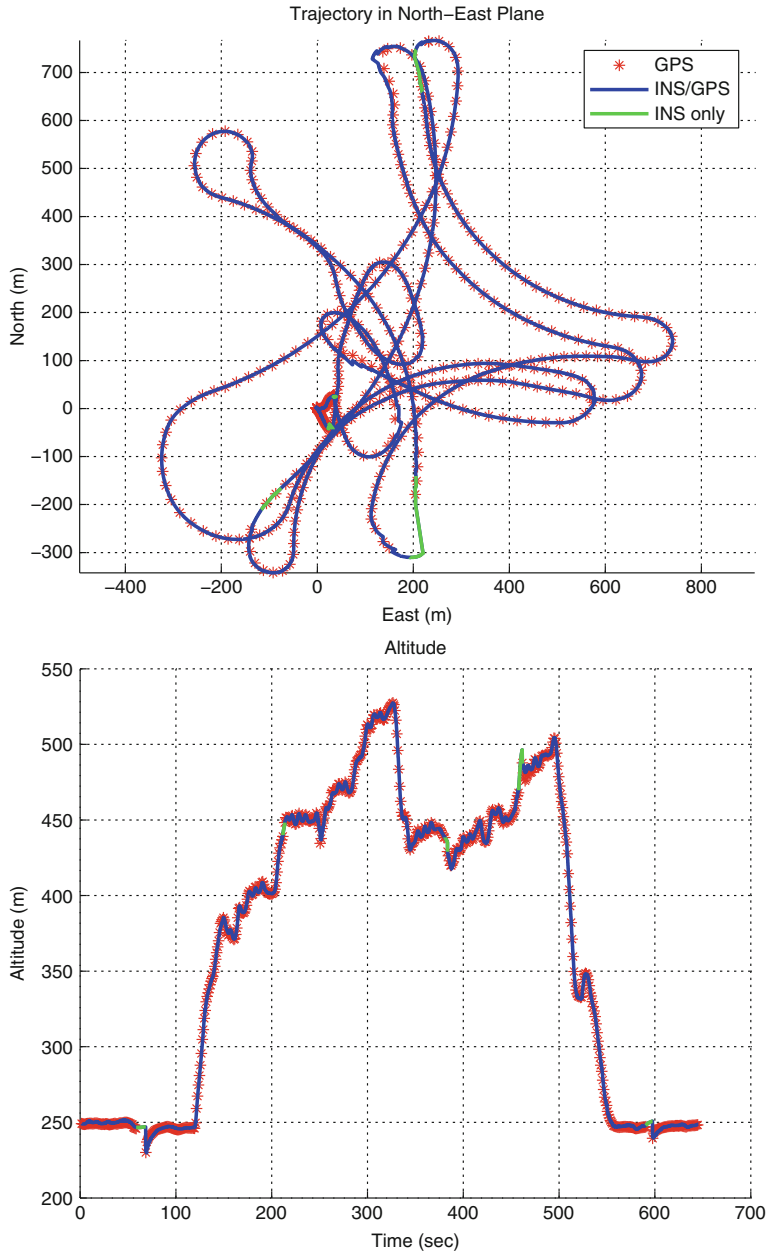


Fig. 18.9 Flight test trajectory

controls rudder and ailerons, while the longitudinal control handles throttle and elevator. While this is reasonable for traditional aircraft-like UAV’s flying gentle maneuvers, it is not appropriate for aircraft with high degrees of lateral-longitudinal cross coupling, nor for aircraft performing aggressive aerobatic maneuvers.

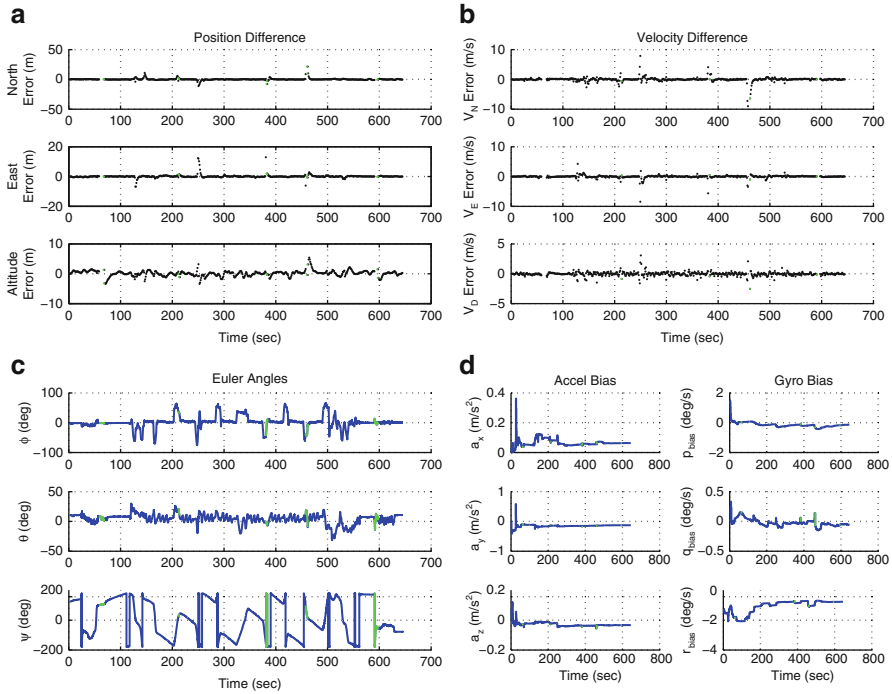


Fig. 18.10 INS/GPS solution: position error (a), velocity error (b), attitude estimate (c), and sensor bias estimate (d)

Furthermore, each of the two main (lateral and longitudinal) controllers is divided into successive loop closure using proportional-integral-derivative (PID) controllers. PID controllers are simple to tune experimentally, and while they can theoretically perfectly stabilize a second-order plant, in practice they work quite well with higher-order plants. Some attention must be paid to integrator windup, and the use of a numerical derivatives in the PID loop, or the resulting controller will have poor performance. Lastly, these controllers are implemented digitally, in order to accommodate the actual autopilot hardware.

18.3.1 PID Control

As stated above, the proportional-integral-derivative (PID) control can perfectly stabilize a second-order plant, given the right gains. However, in practice, it can usually perform well in a variety of settings without the need for a precise model of the underlying plant (a large advantage when system identification is either difficult or imprecise). The classical PID controller consists of an input derived from the measurement of the system and the desired reference for it to track and an actuator signal on the output. The difference between the measurement and the reference,

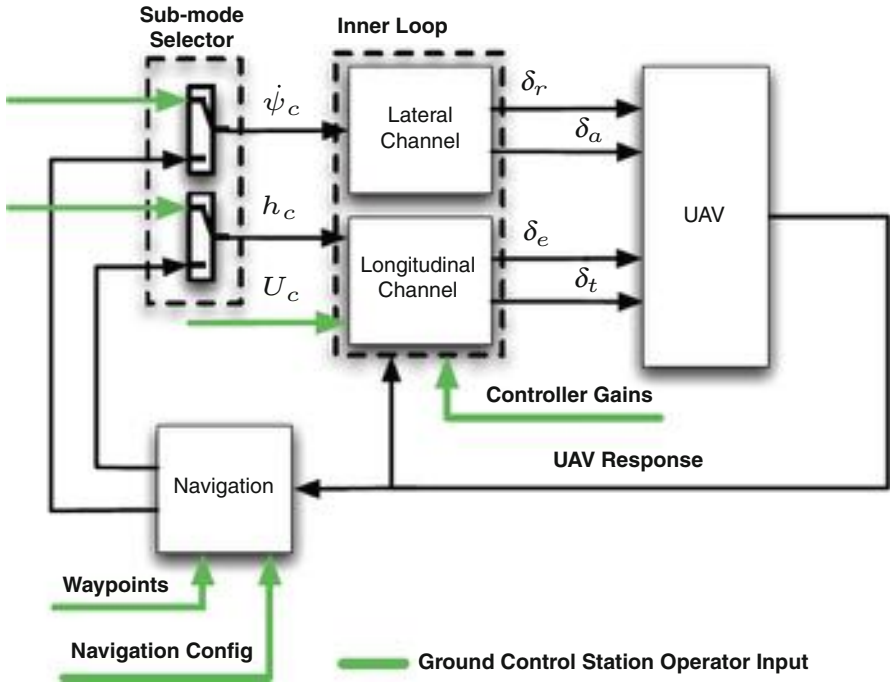


Fig. 18.11 SLUGS block diagram of inner loop and guidance

referred to as the error, is fed into the PID block, and an output is generated from the three gains: K_p , the proportional gain is multiplied directly by the error; K_d , the derivative gain is multiplied by the time rate of change of the error; and lastly, K_i , which multiplies the integral of the error.

At its most basic form, the PID control simply consists of three gains, with a first-order difference for the derivative and a running sum for the integral. Several embellishments are used to make the PID controller behave better at the corner cases. Firstly, the derivative is changed from the time rate of change of the error to the time rate of change of the measurement; this is done so that the PID controller does not throw the output due to a reference change. If a direct derivative signal is available (for instance, pitch rate gyro), then that is used in preference to the first-order difference. Also, the first-order difference can be changed to longer time steps if noise sensitivity is an issue.

Secondly, saturation limits are included to clip the output if it would saturate the actuator. The integral is changed from a running sum to trapezoidal integration, and anti-windup logic is included. Integral state windup is caused by the integrator continuing to integrate even after the control is saturated. This creates a memory effect within the controller that causes overshoot and degrades the controller performance when coming out of saturation. In order to prevent this, an anti-windup scheme is implemented which checks if the actuator would saturate on the current time step and does not perform the integration if this is the case.

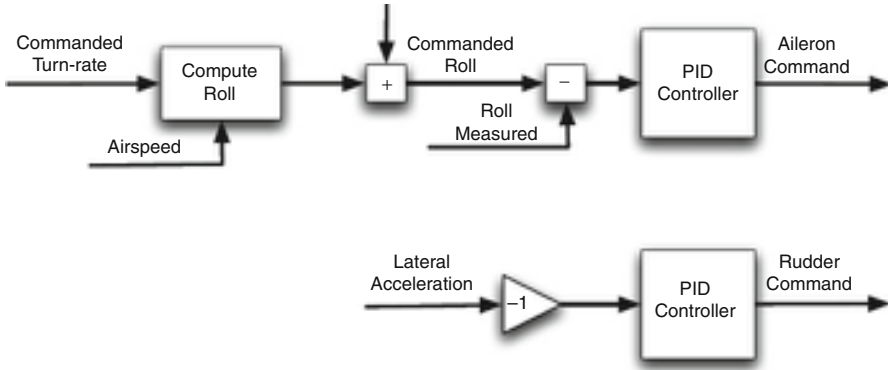


Fig. 18.12 SLUGS lateral inner loop control

The integrator state can be reset under software control, such that when the control is engaged, a bumpless transfer is achieved. Furthermore, the integral state can be manipulated when changing the control gains, such that the output remains stable as the new gains are latched into the controller. Again, all of this is done to maintain smooth control for flight.

18.3.2 Lateral Control

The lateral controller (Fig. 18.12) uses the rudder and ailerons to keep the aircraft flying in a coordinated turn and following a commanded turn rate (including a zero turn rate for straight flight). The lateral dynamics of an aircraft include the roll-rate damping mode, a spiral mode, and the dutch roll mode (yaw-roll coupling). In this formulation of the inner loop control, yaw rate, $\dot{\psi}_c$, is commanded and is converted to a commanded roll angle, ϕ_c , using the formulation

$$\phi_c = \arctan \left(\frac{\dot{\psi}_c U_m}{g} \right) \quad (18.20)$$

where U_m is the measured airspeed and g is gravity. Equation 18.20 assumes that the aircraft is flying in a coordinated turn, that is, where the turn rate is constant and the body-fixed lateral acceleration, a_y , is zero.

The commanded roll angle, ϕ_c , is used as the reference to the PID control, with the plant output being the actual roll angle, ϕ , that comes from the attitude estimation algorithm. The commanded bank angle is limited with a saturation block to keep the roll angle from becoming too extreme. In the case of the SLUGS small UAV, this is limited to $\pm 40^\circ$.

The output generated by the PID block is to the ailerons, which are used to drive the roll error (commanded – actual) to zero. In this case, the derivative of the roll

error is taken directly from the body-fixed gyros (with the bias again taken care of by the attitude estimation algorithm). That is, while the full derivative of roll rate

$$\dot{\phi} = p + (q \sin \phi + r \cos \phi) \tan \theta \quad (18.21)$$

where $[p, q, r]$ are the body-fixed roll, pitch, and yaw rates and ϕ, θ are the roll and pitch euler angles, respectively. Note that with a small pitch angle and yaw rate, this can be approximated as

$$\dot{\phi} \simeq p. \quad (18.22)$$

Lastly, when the roll angle command, ϕ_c , is the opposite polarity from the previous command, the integral state is reset to improve roll tracking performance.

The above describes the turn rate command to aileron control system, which will drive the roll error to zero. However, Eq. 18.20 depends on coordinated flight to be effective. In order to ensure coordinated flight, a second PID loop is closed around to rudder. The commanded input to the turn coordinator PID block is the negative of the lateral acceleration (this accounts for the fact that the rudder is behind the center of mass of the aircraft and has a negative sign in the transfer function), and the output is the rudder actuator command.

The PID loop will actuate the rudder in order to drive the body-fixed lateral acceleration, a_y , to zero. Here, again, the fact that the output of the aircraft affected by the rudder has a direct measurement of its derivative from the body-fixed gyros can be used for the D term. That is,

$$\dot{\psi} = \frac{1}{\cos \theta} (q \sin \phi + r \cos \phi) \quad (18.23)$$

which for small angles of pitch and roll, can be reliably approximated as

$$\dot{\psi} \simeq r. \quad (18.24)$$

Note that the turn coordinator, with the derivative feedback from the body-fixed yaw rate gyro, r , has the effect of also acting as a yaw damper for the aircraft. More traditional autopilots often use the rudder solely as a yaw damper, without the turn coordination function, and rely on the directional stability of the aircraft to keep the turns coordinated. In that case, the rudder feedback command would be based on a high-passed version of the yaw rate gyro. Experience with small UAVs indicates that the former approach works better for model scale aircraft, giving both coordinated turns and decent yaw damping.

18.3.3 Longitudinal Control

The longitudinal low-level control loops (Fig. 18.13) operate similarly to the lateral ones, with the other two flight controls being throttle and elevator. There are two

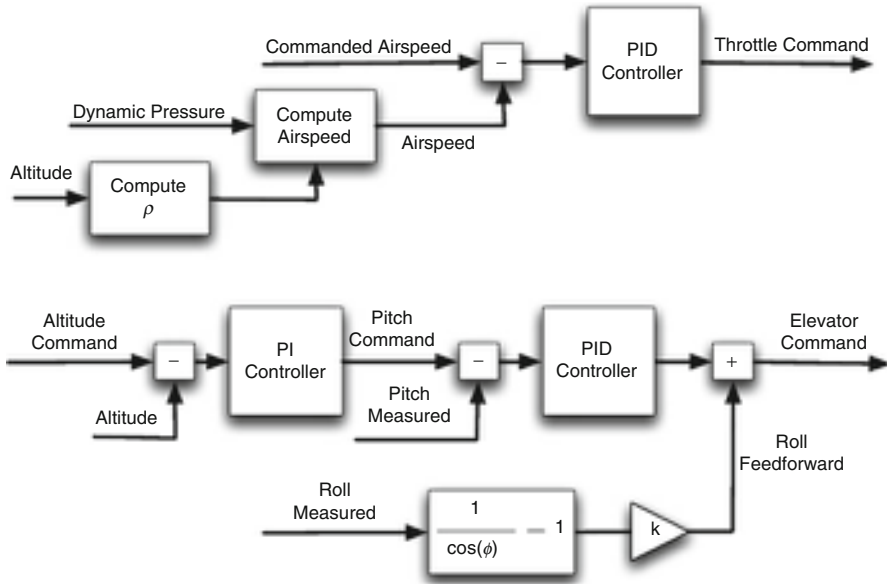


Fig. 18.13 SLUGS longitudinal inner loop control

modes of operation for a longitudinal low-level controller: *climb/descent* and *level flight*. In *climb/descent* mode, the throttle is set to a fixed value (high for climb, low for descent), and the airspeed and climb rate are controlled through the aircraft pitch via the elevator.

In *level flight* mode, airspeed is controlled directly by the throttle and altitude by aircraft pitch via the elevator. Since airspeed is being kept constant, the altitude responds rapidly to even small changes in aircraft pitch. As a design choice, only the *level flight* mode is implemented, given that the UAV will spend most of its mission in level flight at constant altitude following waypoints.

The airspeed hold control loop takes in a commanded airspeed, U_c , and compares this to the measured airspeed. Note that the airspeed control uses airspeed, and not ground speed (as available from GPS). Compensation for wind is handled within the higher-level guidance controller rather than in the low-level stabilizing controllers.

The aircraft does not, however, have an onboard measurement of airspeed. Rather, it has an onboard measurement of dynamic pressure, q , and altitude, from which the airspeed can be extracted. That is,

$$U_m = \sqrt{\frac{2q}{\rho}} \quad (18.25)$$

where q is the dynamic pressure and ρ is the atmospheric density. However, ρ is a function of altitude, such that

$$\rho = \rho_0 \left(1 - \frac{h_m}{44331}\right)^{4.255876} \quad (18.26)$$

where ρ_0 is the sea-level air density (1.025 Kg/m^3) and h_m is the measured altitude in meters. This approximation is good for most altitudes reachable by UAVs and is sufficient for control. The output of the PID block is the throttle actuator. In the case of the airspeed, there is no direct derivative to be used in the throttle loop (though the body-fixed longitudinal acceleration could be used). As such, some care must be used in both the derivatives and in tuning the airspeed hold loop in general.

Too aggressive gains on the airspeed hold loop causes the throttle to surge and cut in flight, with unpleasant results. This is due both to the noise on measured airspeed and also due to the lag in response to the throttle input. Here tuning the gains for an acceptable error and relying on the integral term to close the error works well.

The altitude control loop consists of two chained PID controllers. The first is a straight PI controller (the derivative gain is set to 0) that takes in commanded altitude, h_c , and measured altitude, h_m , as its error and outputs a pitch command. The pitch command is saturated at $\pm 15^\circ$ with the integral windup stopped if the saturation is reached. The limited pitch command is used as a reference input for the second PID loop and is compared to the aircraft pitch, θ , from the attitude estimation to generate the error. The output of the second PID loop drives the elevator, which works to match the aircraft pitch attitude to the commanded one. Here, the derivative term comes straight from the body-fixed gyros, with the full equation

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (18.27)$$

which for small bank angles can be reduced to

$$\dot{\theta} \simeq q. \quad (18.28)$$

The direct derivative term allows the pitch to elevator PID control to be aggressive, yet still retain good damping characteristics. This PID controller will drive the aircraft pitch to the commanded pitch until the desired altitude is reached.

While the aircraft is flying straight and level, this control works very well. However, while the aircraft is in a turn, the aircraft will descend. This is because the lift from the wings must match the aircraft weight, but in a turn part of the lift is directed inwards to cause the turn itself. More precisely, in order not to lose altitude in a turn, the lift is related to the weight and the bank angle as

$$L = \frac{W}{\cos \phi} \quad (18.29)$$

where L is lift, W is aircraft weight, and ϕ is roll angle. This is a simplified model assuming lift only from the wings and not accounting for sideslip angles of the fuselage for knife-edge type flight. Using this model, the change in lift is:

$$\Delta L = W \left(\frac{1}{\cos \phi} - 1 \right) \quad (18.30)$$

which indicates how much the lift must be increased to maintain altitude in a coordinated circular turn. Pilots are trained to increase elevator (back stick) when entering a turn to maintain altitude. In order to ensure that the UAV holds altitude during turns, a feedforward gain proportional to the increase in lift is included:

$$\delta e_{ff} = K_{ff} \left(\frac{1}{\cos \phi} - 1 \right) \quad (18.31)$$

where δe_{ff} is the additional elevator command due to the feedforward term. Note that in the implementation, the roll angle, ϕ , is low pass filtered to reduce unwanted pitch oscillations resulting from attitude estimation noise and is also limited to $\pm 60^\circ$. At angles beyond this amount, the UAV will simply enter into an accelerated stall trying to hold altitude, and the low-level control system will no longer be able to stabilize the aircraft.

18.3.4 Trim Conditions

Before leaving the subject of the low-level inner loop controllers, it is necessary to discuss aircraft trim conditions. In the case of the SLUGS small-scale UAV, trim is established by the safety pilot before the autopilot is engaged. At the moment the autopilot is switched on, the inner control loops assume the aircraft is currently trimmed for straight and level flight.

The inner PID control loops above each output a *change* in actuator output summed with the original trim condition. Note again that for the SLUGS autopilot, this trim is established by a human safety pilot. The trim could just as easily be established using a conventional 6DOF model of the airframe and computing trim for various flight conditions. That is, for every combination of airspeed, climb rate, and altitude, there is a throttle and elevator setting that will keep the UAV flying in steady state at those conditions. The ailerons and rudder surfaces are assumed to be set at zero for straight flight.

18.3.5 Simulation and Flight Test Results

In order to validate the inner loop control, data from the hardware-in-the-loop (HIL) simulation is presented showing the results of the inner loop PID controllers. [Figure 18.14a](#) shows the lateral controller. Note that at every change in commanded roll angle, the lateral acceleration receives a large disturbance, which must be washed out. In panel (b) is the longitudinal control, with both pitch and airspeed control. Again, note that the spikes in airspeed are due to large climb or descent inputs to the controller.

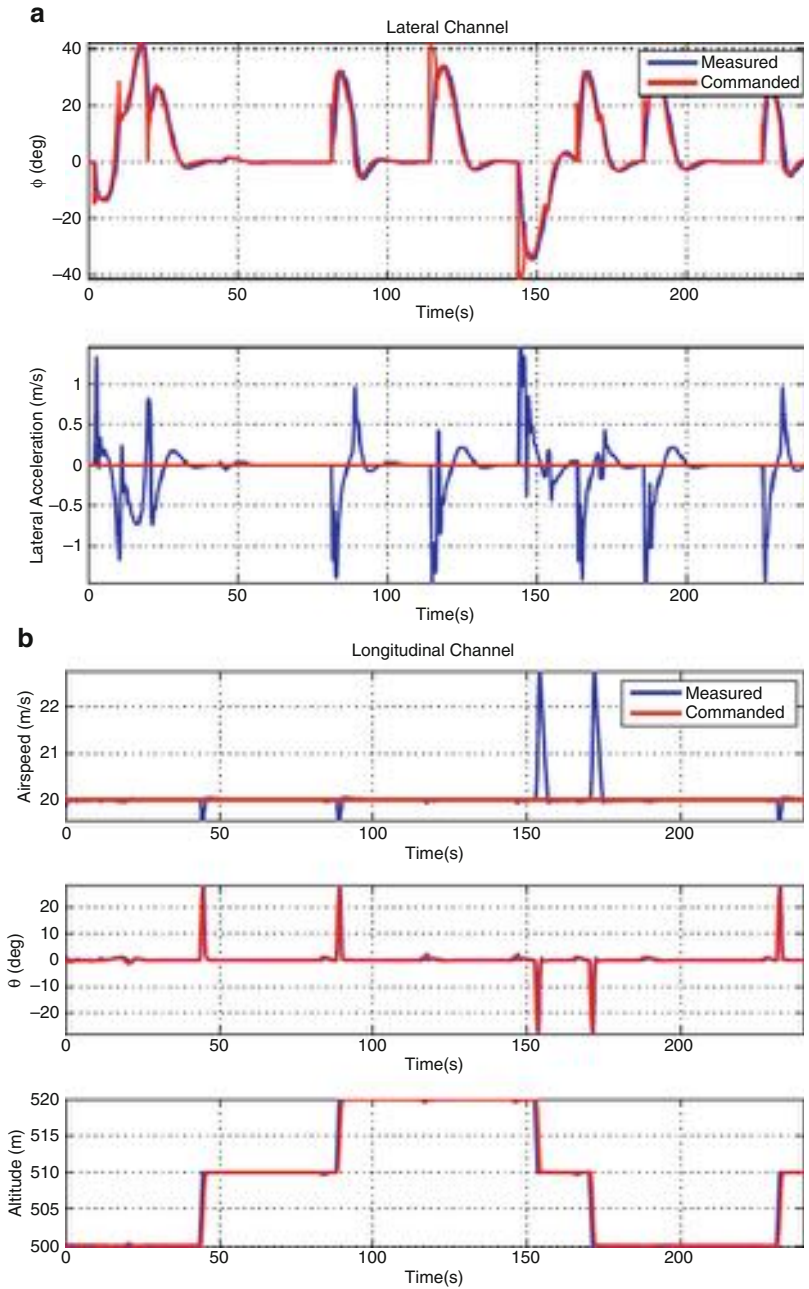


Fig. 18.14 Simulation of the inner loop control using the HIL simulator: (a) the lateral channel, (b) the longitudinal channel

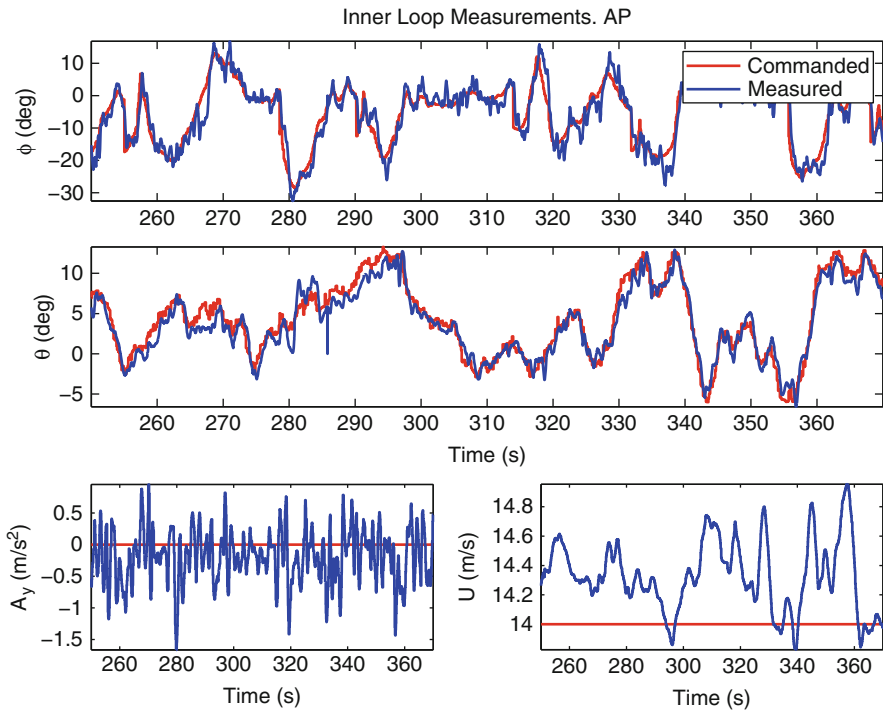


Fig. 18.15 SLUGS inner loop flight test data

Flight test data from the inner loop control is presented in Fig. 18.15, which shows excellent performance on pitch and roll commands, a reasonable attenuation of lateral acceleration (with a limit to less than 1 g for most of the flight time) and a fairly good airspeed hold. This is in the presence of wind, gusts, and other disturbances. Note that the bias on the lateral acceleration is most likely from the accelerometer being mounted with a slight tilt on the airframe.

18.4 Guidance

UAV guidance, navigation, and control (GNC) ultimately signifies the ability to follow a desired trajectory through the sky. With attitude estimation established (Sect. 18.2), and inner loop stabilizing the aircraft (Sect. 18.3), what is left is to guide the UAV along the desired trajectory rejecting disturbances such as wind.

Again, while there are several ways to implement this guidance control, this section discusses the guidance algorithms implemented on the SLUGS autopilot (Lizarraga 2009). In order to deliver a mission to the UAV from the ground station, a simple set of GPS-based waypoints, along with an altitude and speed for each leg of the mission, is transmitted to the UAV via telemetry link. Furthermore, the interest

is in flying the aircraft on the trajectory, and thus the legs are blended into each other using circular arcs rather than forcing the UAV to overfly each waypoint. Note that this is a design choice, and both variants have a utility depending on the specific mission.

The SLUGS guidance algorithm is based on an extension of a simple line-of-sight guidance law originally developed for ground robotics. Much of the art of waypoint guidance consists of determining which leg of the trajectory the UAV is on and when to switch to the next leg of the trajectory.

The basic approach is to define a Serret-Frenet frame which points from the current waypoint to the next (\vec{T} , along the current leg), with one axis down and the third defined by the cross product between the along track, \vec{T} , and down, \vec{B} , unit vectors. The projection of the UAV position onto the \vec{T} direction is used to determine when to switch to the next leg. In practice this is better than using a distance from the next waypoint to switch (especially in very windy conditions).

18.4.1 General Tracking

The outer loop guidance law follows closely the work developed in Amidi and Thorpe (1991), Park et al. (2004), and Park et al. (2007). The original derivation is presented here for completeness. The guidance algorithm steers the velocity vector toward the line of sight; this is one form of pursuit guidance. Making the commanded acceleration proportional to $\sin \eta$ is only one of number of possible guidance laws. The basic guidance algorithm is to determine an aim point and steer the velocity vector toward it.

Referring to Fig. 18.16, V_g is the UAV's ground speed, and C is a circular arc of radius R that originates at the UAV and intercepts the desired path. L_1 is a constant look-ahead distance from the UAV to the path in the desired direction of travel. From elementary trigonometry:

$$\frac{|L_1|}{2} = R \sin \eta. \quad (18.32)$$

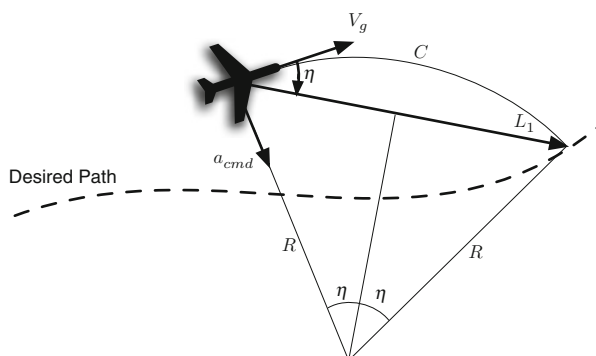


Fig. 18.16 Navigation control law geometry (Reproduction of Fig. 1 in Park et al. 2004)

Additionally, from elementary kinematics, it is known that the centripetal acceleration, a_c , required for a point mass to follow the circular arc C is given by

$$a_c = \frac{|V_g|^2}{R}. \quad (18.33)$$

Thus the UAV must command a lateral acceleration of a_c . Solving Eq. 18.32 for R and substituting it into Eq. 18.33 produces the following control law for commanded acceleration:

$$a_{cmd} = 2 \frac{|V_g|^2}{|L_1|} \sin \eta. \quad (18.34)$$

The only requirements for the implementation of this control law are to select the lookahead distance $|L_1|$ and determine $\sin \eta$, the sine of the angle from the velocity vector to L_1 . η is sometimes called the line of sight angle. Choice of $|L_1|$ is analogous to feedback gain, with a larger L_1 corresponding to smaller gains; $\sin \eta$ is found from the vector cross product of V_g and L_1 .

$$\sin \eta = \frac{|V_g \times L_1|}{|V_g||L_1|}. \quad (18.35)$$

For the UAV to actually track the desired trajectory, the lateral acceleration command must be converted to an appropriate bank angle command using the steady-state turn equation:

$$\phi_{cmd} = \tan^{-1} \frac{a_{cmd}}{g}. \quad (18.36)$$

18.4.2 Straight Line Tracking

The guidance algorithm is most easily described using a straight line (though it is certainly not limited to such). In Fig. 18.17, the UAV is following a straight line segment from waypoint P_0 to P_1 . A Serret-Frenet coordinate frame (Etkin 2005) is attached to the initial waypoint, P_0 , such that the calculations are always in local path coordinates.

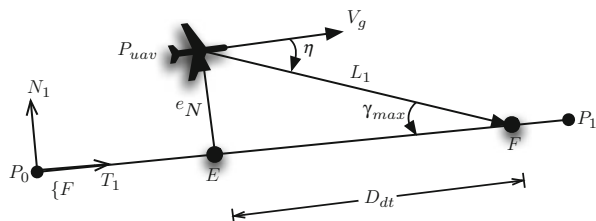


Fig. 18.17 Geometry description of the angle η computation

$$\begin{aligned}
T &= [x_T, y_T, z_T] = \frac{(P_1 - P_0)}{|P_1 - P_0|}, \\
N &= \frac{1}{[[-y_T \ x_T \ 0]]} [-y_T \ x_T \ 0]^\top, \\
B &= T \times N.
\end{aligned} \tag{18.37}$$

Given that e_N is the the error in the N direction. Then

$$|e_N| = N^\top (P_{uav} - P_0), \tag{18.38}$$

E is the closest point on the path to the UAV. The intersection of L_1 with the desired path is shown by the point F . The down-track distance or EF separation, denoted D_{dt} , can be computed as

$$D_{dt} = \sqrt{|L_1|^2 - |e_N|^2}, \tag{18.39}$$

with the vectors E and F given by

$$E = P_{uav} - |e_N|N, \tag{18.40}$$

$$F = E + D_{dt}T. \tag{18.41}$$

From these relations, the vector L_1 can then be computed as

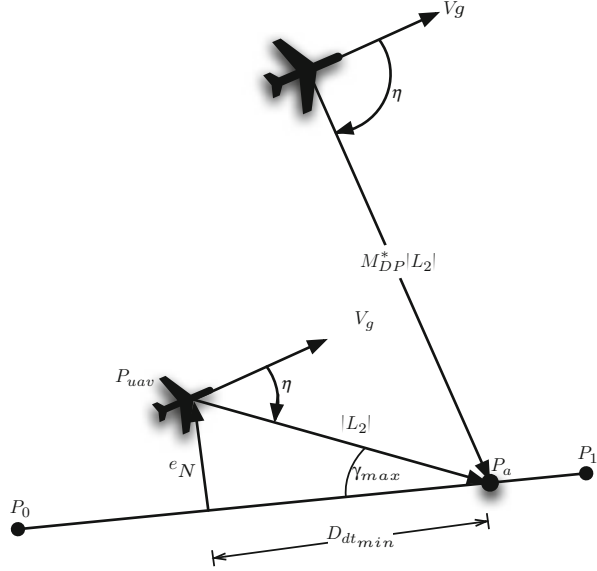
$$L_1 = F - P_{uav} = D_{dt}T - |e_N|N, \tag{18.42}$$

which is then used in Eq. 18.35 to determine η and hence the commanded acceleration. For sufficiently small tracking errors, the L_1 guidance looks like a PD controller on lateral error (Park et al. 2007).

18.4.3 L_2^+ Control

The SLUGS autopilot extends the L_1 pursuit guidance to account for some shortcomings of the control law (and for clarity refers to this as L_2^+ control). Firstly, it was noticed during flight test experiments that the L_1 guidance exhibited large overshoots when turning downwind (hence an increasing V_g). Analysis showed that in order to solve this, the new L_2 vector should be a function of ground speed. Thus, $|L_2| = \mathcal{T}^*|V_g|$, where \mathcal{T}^* is a constant and the commanded acceleration becomes

$$a_{cmd} = 2 \frac{|V_g|}{\mathcal{T}^*} \sin \eta. \tag{18.43}$$

Fig. 18.18 L_2^+ geometry

Furthermore, when the lateral error, $|e_N|$, is larger than $|L_2|$, the aim point cannot be found. To enforce that an aim point is always found, define a maximum intercept angle, γ_{\max} ; the down-track distance to the aim point defined by γ_{\max} is

$$D_{dt} = \frac{|e_N|}{\tan \gamma_{\max}}, \quad (18.44)$$

as shown in the lower aircraft in Fig. 18.18.

When $|e_N|$ is greater than $|L_2|$, such as during initial intercept, D_{dt} may become too large. For example, the aim point may go beyond the next waypoint. To prevent this a bound is placed on the down path distance of the aim point. This bound is set to be a constant, $\mathcal{M}_{\mathcal{DP}}^*$, times $|L_2|$:

$$D_{dt_{\min}} = \min(D_{dt}, |L_2| \cdot \mathcal{M}_{\mathcal{DP}}^*). \quad (18.45)$$

This is shown for the upper aircraft in Fig. 18.18. The down-track distance of the aim point is

$$D_{dt_{\min}}^* = \begin{cases} D_{dt_{\min}} & |e_N| > |L_2| \\ \max(D_{dt_{\min}}, \sqrt{|L_2|^2 - |e_N|^2}) & |e_N| \leq |L_2| \end{cases} \quad (18.46)$$

If the aim point gets beyond P_1 , the UAV will continue along this line without ever changing direction. First compute D_{wp1} , the along-track distance from the UAV to P_1 :

$$D_{wp_1} = T^\top (P_1 - P_{uav}). \quad (18.47)$$

This will be a negative number if the UAV is beyond P_1 . The distance from P_1 back along T to the aim point is

$$D_a = D_{wp_1} - D_{dt_{\min}}^*. \quad (18.48)$$

The final aim point P_a is then computed:

$$P_a = -T_1 \cdot \max(0, D_a) - P_{wp_1}. \quad (18.49)$$

The *max* operation ensures that the aim point does not extend beyond P_1 , i.e., if D_a ever becomes negative due to UAV position or failure of the waypoint switching logic, the acceleration command is then computed with Eq. 18.43, where the $\sin \eta$ is obtained from the cross product of V_g and P_a , Eq. 18.35.

Note that when the aircraft is initially pointed in the opposite direction from T , η is greater than 90° . In this case, a maximum lateral acceleration is imposed to return the vehicle to the correct flight path, where:

$$a_{\max} = g \tan \phi_{\max} \quad (18.50)$$

and ϕ_{\max} is the maximum bank angle permitted (typically set to 30 – 60° for a small UAV). In order to use the L_2^+ controller in this condition, a maximum allowable η is required:

$$\eta_{\max} = \min\left(\frac{\pi}{2}, \mathcal{T}_{LEAD} \frac{a_{\max}/2}{U_{\text{comm}}}\right) \quad (18.51)$$

where U_{comm} is the commanded airspeed from the inner loop controller, and \mathcal{T}_{LEAD} is the lead time required for the aircraft to roll out of a steep bank (and determined experimentally for the airframe).

18.4.4 Waypoint Switching

As previously stated, the guidance strategy is to move from one segment to the next of the flight mission by connecting the two segments with a circular arc. There are myriad ways to switch between segments; this is simply the one implemented on the SLUGS. The L_2^+ implementation uses a policy of early waypoint switching to prioritize path following instead of waypoint precision.

Let C be a circle of radius R given by

$$R = \frac{(U_c + |wind|)^2}{a_{\max}}. \quad (18.52)$$

Here U_c is the commanded airspeed, $|wind|$ is the windspeed, and a_{\max} is the maximum acceleration for the UAV. In the presence of wind the actual radius of

$$p = \mathcal{T}_{LEAD} \cdot |V_g| + \frac{R}{\tan \delta}. \quad (18.58)$$

Finally, the vector position of the switch point P_{sw1} is given by

$$P_{sw1} = P_1 - pT_1 \quad (18.59)$$

Note that during the transition from missions leg, the L_2 vector intercepts the circular arc, not the straight line segments. Also, depending on the exact geometry of the waypoints, and the aircraft speed, it may be that just after switching legs, the aircraft is already beyond the next segment switching point. In this case, the logic immediately switches again to the next leg.

18.4.5 Point Acquisition and RTB

One advantage of the L_2^+ controller is that it is quite robust. For instance, without any change in the logic, the same controller (with its commanded lateral acceleration/bank angle) can just as easily drive the UAV to a point as well as follow an arbitrary curve.

For the case of an acquisition point, P_a , the angle η is computed from the current UAV position and P_a :

$$\sin \eta = \frac{|V_g \times (P_a - P_{uav})|}{|V_g| |(P_a - P_{uav})|}. \quad (18.60)$$

Again with the limits of η_{\max} and downrange distance imposed to limit bank angle and lateral acceleration, implemented in Eqs. 18.33 and 18.36. This controller will essentially “point at the point” until it overflies the acquisition point, P_a , at which point the switch logic will cause it to circle the point. Of interest is that nowhere in the formulation does the point P_a have to be fixed; the same L_2^+ controller can track a moving target using its same logic assuming you have a position estimate of the target. Simulations have shown that for target speeds moving at or below the ground speed of the UAV, the UAV always acquires the target.

The *initial mission point*, P_I , before the first waypoint is determined using concept borrowed from instrument flying in which all aircraft must first fly to a well-defined point before proceeding to land. The initial point is determined by projecting a fixed point in front of the first leg of the mission a constant distance in front of the initial waypoint, Fig. 18.20:

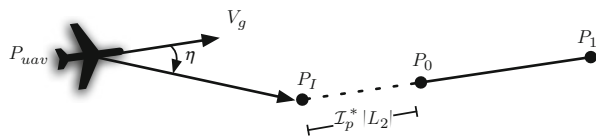


Fig. 18.20 Initial point geometry

$$P_I = P_a = P_0 - T_1 \cdot (\mathcal{I}_P^* \cdot |L_2|) \quad (18.61)$$

where \mathcal{I}_P^* is simply a constant tuned to the specific aircraft.

Lastly, the L_2^+ controller implements a *return to base* (RTB) functionality by simply recording the original base position, P_b , and using this as the acquisition point if either the mission waypoints are completed or if communication to the UAV fails.

18.4.6 Simulation and Flight Test Results

The SLUGS autopilot has both a full simulation (based on MATLAB's Simulink) and a hardware-in-the-loop (HIL) simulation with the algorithms running on the SLUGS embedded hardware, while the aircraft is simulated through a full 6DOF model running on a separate computer. Figure 18.21 demonstrates the full L_2^+ running in simulation, showing the initial point, transitions, and a RTB at the end of the flight.

The SLUGS-based UAV, a small electric RC aircraft (Multex Mentor), was flown at UCSC, using both the inner loop control and outer loop guidance described above. Figure 18.22a shows the flight test results (Lizarraga 2009) in the presence of strong winds and real flight disturbances. Panel (b) shows the L_2 vector in real time as the aircraft transitions through the waypoints.

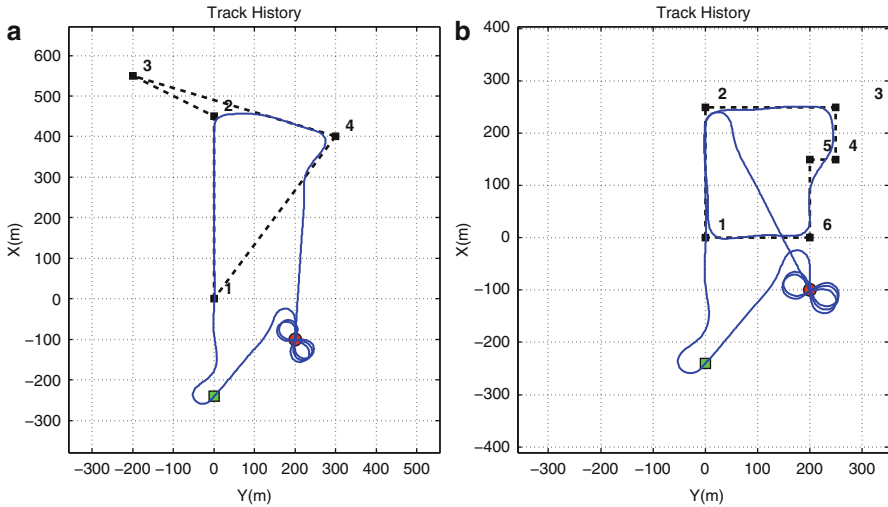
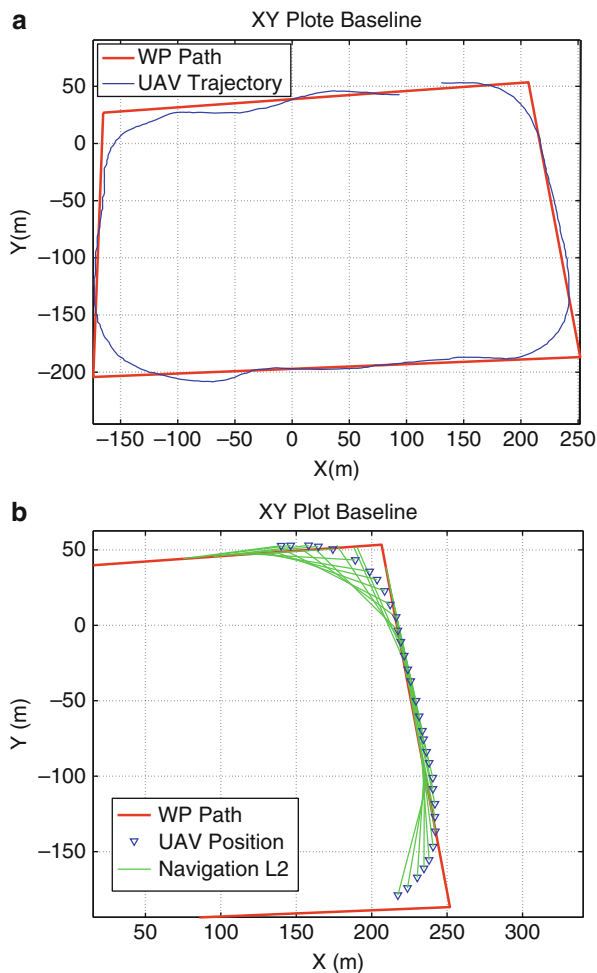


Fig. 18.21 Return to base implementation in two different waypoint paths. Both have an initial point (green square) and return to the base (red circle). Note the logic skips the waypoint 2 in (a) and waypoint 5 in (b)

Fig. 18.22 Flight data for the L_2^+ controller on the Mentor platform, showing path tracking performance in (a) and the evolution of the L_2 vector at the waypoint switches in (b)



18.5 Conclusion

In this chapter the complete system architecture for two UAV GNC solutions is described. The systems described a low-cost and easily reconfigurable solution which is intended to support research efforts associated with guidance, navigation, and control of small UAVs. The purpose of the description and discussion provided was to show how to implement (hardware and software) a low-cost GNC solution. It was not intended, however, to be the definitive GNC solution for small UAVs. Such solutions do not exist, and this fact was the impetus for writing this chapter. Size, weight, and power constraints associated with small UAVs preclude a “one size fits all” GNC solution. Thus, while the guidance, navigation, and control solutions presented in the chapter were suited for the two UAVs described and the mission

they were intended for (flight control and guidance research), it does not mean they are ideal for application based on other sensors, vehicles, or mission. However, they are flexible enough that they can be adapted to support other missions.

References

- O. Amidi, C. Thorpe, Integrated mobile robot control, in *Proceedings of the SPIE*, Boston, vol. 1388, 1991, p. 504
- C.-C. Chu, F.A.P. Lie, L. Lemay, D. Gebre-Egziabher, Performance comparison of tight and loose INS-Camera integration, in *Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2011)*, Portland, 2011, p. 3516
- A. Dorobantu, A. Murch, B. Mettler, G. Balas, Frequency domain system identification for a small, low-cost, fixed-wing UAV, in *AIAA Guidance, Navigation, and Control Conference*, Portland, 2011
- B. Etkin, *Dynamics of Atmospheric Flight* (Dover, Mineola, 2005)
- J. Farrell, M. Barth, *The Global Positioning System and Inertial Navigation* (McGraw-Hill, New York, 1999)
- D. Gebre-Egziabher, Design and performance analysis of low-cost aided dead reckoning navigator. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, Stanford
- S. Gleason, D. Gebre-Egziabher, *GNSS Applications and Methods* (Artech House, Boston, 2009)
- P. Groves, *Principles of GNSS, Inertial, and Integrated Navigation Systems* (Artech House, Boston, 2008)
- E.J. Lefferts, F.L. Markley, M.D. Shuster, Kalman filtering for spacecraft attitude estimation. *J. Guid. Control Dyn.* **5**(5), 417–429 (1982)
- F.A.P. Lie, D. Gebre-Egziabher, A synthetic airdata system, in *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, 2012
- M. Lizarraga, Design, implementation and flight verification of a versatile and rapidly reconfigurable UAV GNC research platform. PhD thesis, Department of Computer Engineering, University of California Santa Cruz, Santa Cruz, 2009
- M. Lizarraga, V. Dobrokhodov, G.H. Elkaim, R. Curry, I. Kaminer, Simulink based hardware-in-the-loop simulator for rapid prototyping of uav control algorithms, in *AIAA Infotech Conference*, Seattle, 2009a
- M. Lizarraga, G.H. Elkaim, G. Horn, R. Curry, V. Dobrokhodov, I. Kaminer, Low cost rapidly reconfigurable uav autopilot for research and development of guidance, navigation and control algorithms, in *ASME/IEEE MESA09*, San Diego, 2009b. International Conference on Mechatronic and Embedded Systems and Applications
- M. Lizarraga, R. Curry, G. Elkaim, Reprogrammable uav autopilot system (part 1) – system hardware and software. *Circuit Cellar* (249), 24–35 (2011a)
- M. Lizarraga, R. Curry, G. Elkaim, Reprogrammable uav autopilot system (part 2) – testing and results. *Circuit Cellar* (250), 36–43 (2011b)
- P. Misra, P. Enge, *Global Positioning System, Signals, Measurements, and Performance* (Ganga-Jamuna Press, Lincoln, 2001)
- A. Murch, UMN UAV Flight Code Documentation (2012a), <http://www.uav.aem.umn.edu/uav/doxygen/html/index.html>
- A. Murch, University of Minnesota UAV Laboratory (2012b), <http://www.uav.aem.umn.edu>
- D. Owens, D. Cox, E. Morelli, Development of a low-cost sub-scale aircraft for flight research: the FASER project, in *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, San Francisco, 2006
- S. Park, J. Deyst, J.P. How, A new nonlinear guidance logic for trajectory tracking, in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Portland, 2004

- S. Park, J. Deyst, J.P. How, Performance and Lyapunov stability of a nonlinear path-following guidance method. *J. Guid. Control Dyn.* **30**(6), 1718 (2007)
- D. Simon, *Optimal state estimation: Kalman, H_∞ , and nonlinear approaches* (Wiley, Hoboken, 2006)
- SLUGS website, <http://slugsuav.soe.ucsc.edu>
- J. Stewart, *Calculus: Early Transcendentals*, 4th edn. (Brooks/Cole, Belmont, 2001)
- D. Titterton, J. Weston, *Strapdown Inertial Navigation Technology* (Institution of Engineering and Technology, Stevenage, 2004)
- UAS CoE, Eyes of the army: US army roadmap for unmanned aircraft systems 2010–2035 (2010), <http://www.rucker.army.mil/usaace/uas/>
- University of California Santa Cruz Autonomous Systems Lab, <http://asl.soe.ucsc.edu>

Section IV

Sensors and Sensing Strategies

Ben Upcroft and Salah Sukkarieh

Kimon P. Valavanis and George J. Vachtsevanos

Sensors and Sensing Strategies enable an unmanned aircraft to “sense,” “see,” “hear,” and “understand” the world around it so that it may function intelligently in an unknown and cluttered environment and in the absence of an onboard pilot. In essence, sensors and sensing strategies are crucial since they provide the technologies that will result in “unmanned aircraft operating as if there were a human pilot onboard.”

► [Sensors for Missions](#) by Mejias, Lai, and Bruggemann sets the tone for sensors used on UAVs that are assigned complex missions. The sensor suite onboard a UAV is tightly coupled with payload capabilities, as payload dictates UAV usability and market value. However, advances in miniaturization of electronics are enabling replacement of multiprocessing, power-hungry general-purpose processors with more integrated and compact electronics that contribute to more onboard sensors. Several common payload sensors are described along with their usefulness to solve real-world problems.

► [Inertial Sensor-Based Simultaneous Localisation and Mapping for UAVs](#) by Bryson and Sukkarieh provides an overview of algorithms for inertial sensor-based simultaneous localization and mapping (SLAM) within the context of UAVs, using the extended Kalman filter (EKF) and the extended information filter (EIF) due to their ease of understanding, applicability to online implementation, and prevalence in airborne localization applications outside of SLAM.

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

► **UAV Localisation Using Inertial Sensors and Satellite Positioning Systems** by Bryson and Sukkarieh provides an overview of UAV localization with a focus on aided inertial localization, that is, algorithms for fusing data from, for example, satellite positioning systems, barometric sensors, and magnetometers with inertial sensors to provide real-time position and orientation. An example implementation of aided inertial localization on a UAV is presented as a tutorial to understand key concepts in airborne localization and as a basic guide toward more complicated implementations.

► **Data Fusion and Tracking with Multiple UAVs** by Ridley, Upcroft, and Sukkarieh describes decentralized data fusion (DDF) algorithms for a team of multiple autonomous platforms. It is shown how through the DDF algorithms each platform can maintain a consistent global solution from which decisions may be made. The overall system design is detailed, providing insight into the overall complexity of implementing a robust DDF system for use in information-gathering tasks in outdoor UAV applications.

Collectively, after reading and understanding the first four sections of the handbook, the reader, novice, or expert will be ready to continue with the actual control of UAVs and all other more advanced technical aspects.

Luis Mejias, John Lai, and Troy Bruggemann

Contents

20.1 Introduction 386

20.2 Navigation Sensors..... 386

 20.2.1 Electro-Optical (EO) 387

 20.2.2 Radio-Wave Sensors..... 389

20.3 Applications 390

 20.3.1 Collision Avoidance 390

 20.3.2 Remote Sensing: Power Line Inspection and Vegetation Management..... 394

References 398

Abstract

An onboard payload may be seen in most instances as the “Raison d’Etre” for a UAV. It will define its capabilities, usability and hence market value. Large and medium UAV payloads exhibit significant differences in size and computing capability when compared with small UAVs. The latter has stringent size, weight, and power requirements, typically referred as SWaP, while the former still exhibit endless appetite for compute capability. The tendency for this type of UAVs (Global Hawk, Hunter, Fire Scout, etc.) is to increase payload density and hence processing capability. An example of this approach is the Northrop Grumman MQ-8 Fire Scout helicopter, which has a modular payload architecture that incorporates off-the-shelf components. Regardless of the UAV size and capabilities, advances in miniaturization of electronics are enabling the replacement of multiprocessing, power-hungry general-purpose processors with more integrated and compact electronics (e.g., FPGAs).

The payload plays a significant role in the quality of ISR (intelligent, surveillance, and reconnaissance) data, and also in how quickly that information can be delivered to the end user. At a high level, payloads are important enablers of greater mission autonomy, which is the ultimate aim in every UAV.

This section describes common payload sensors and introduces two cases in which onboard payloads were used to solve real-world problems. A collision avoidance payload based on electro optical (EO) sensors is first introduced, followed by a remote sensing application for power line inspection and vegetation management.

20.1 Introduction

There are two main categories of payloads onboard a UAV: those that allow the vehicle to navigate and those that allow the vehicle to perform its main task. The distinction between them is sometimes very subtle. In some cases, a payload aimed to perform a task can be used for navigation purposes (e.g., cameras), and navigation payloads may in some instances be integrated with other sensors to perform a specific task (e.g., GPS/INS with LIDAR). This section will describe common sensors that are used in many typical payloads onboard UAVs.

20.2 Navigation Sensors

At the core of most UAV guidance and navigation systems, one can find Global Navigation Satellite Systems (GNSS) (including the Global Positioning System – GPS) and Inertial Navigation Systems (INS). Their complementary nature has been recognized, and as a result, GPS and INS sensors are the preferred sensor couple for the majority of autopilot systems.

The integration of GPS and INS is without doubt the area in which researchers have spent considerable efforts proposing approaches such as uncoupled integration, loosely coupled integration, tightly coupled integration, and deeply coupled integration (Grewal et al. 2007). GPS and INS are not the only two sensors used for navigation. They can be complemented with altimeters (laser-based, barometric, etc.) to enhance the estimation of the vehicle state. Additionally, infrared attitude sensors are typically found in micro UAVs. Recently in Cao et al. (2010), a survey of UAV autopilot alternatives and typical sensor combinations was presented.

At the heart of the integration scheme lies a estimator (usually a form of Kalman filter) which estimates position, velocity, attitude, GPS errors and inertial sensor errors. Due to their complementary nature, GPS and INS are often the preferred core sensor suite. However, researchers have also investigated the integration of other sensor combinations, such as GPS with computer vision (Dusha et al. 2011; Wein et al. 2011; Dusha and Mejias 2012) and INS with computer vision (Merz et al. 2006). Additionally, factors such as the trade-off between cost and accuracy

of INS sensors and the susceptibility of GPS to spoofing and jamming have also contributed to increased interest in alternative sensor combinations.

While alternative integration schemes using other sensors is an attractive option, the low cost and future navigation availability and integrity that Space-Based Augmentation Systems (SBAS) such as WAAS, EGNOS, GAGAN, and MSAS will provide cannot be ignored. Submeter accuracy for civilian users will also be possible with the commissioning of Galileo, Compass, and the modernization of GLONASS. This in turn will encourage interoperability and the possibility of a triple-frequency civilian-band GPS over the next decades.

20.2.1 Electro-Optical (EO)

Nowadays, it is difficult to realize a UAV without an EO sensor. They have become standard fit-out onboard many aerial vehicles. The challenge is now in the processing and interpretation of the information acquired by EO sensors. Perception through EO sensors can be seen as one of the most important tasks in a UAV. Whether it is performed for navigation or for surveillance (as an end application), it defines the necessary peripherals to process the EO data. This section will introduce some of the most common EO sensors typically found in UAVs.

20.2.1.1 Visible Spectrum

A visible-spectrum camera operates about 390 nm (3.9 μm)–750 nm (7.5 μm) wavelength. These cameras can be found in two main categories: digital still cameras and machine vision cameras (including surveillance and webcam). Digital still cameras offer very high resolution, but cannot provide a continuous stream of images. The amount of images they can provide usually depends on the amount of internal memory. This type of camera sees application in remote sensing and aerial photography. Machine vision cameras have relatively lower resolution but can provide a continuous stream of images up to a few hundred frames per second. The speed is related with the resolution and output format used (digital or analog). They are suitable for processes or tasks that require very fast perception of environment. Common output protocols and interfaces for machine vision cameras include IEEE 1394, Camera Link, SD/HD Analog, USB, GigE Vision, and in coming years Thunderbolt cameras.

They can provide color or gray level (or both) images. The data representation or color space usually varies from one manufacturer to another. Typical color spaces or models used in most machine vision cameras are RGB, YUV, YPbPr, and YCbCr, etc.

Regardless of its end use and camera type, camera geometric models are necessary. They provide parameters that are needed to correct lens distortions, perform the mapping or representation of 3D objects onto the 2D surface called *image*, and overall allow manipulation of the data acquired by these devices. These parameters are normally estimated through a calibration process that is regularly

performed. For more details on camera models and calibration theory, refer to Forsyth and Ponce (2002) and Szeliski (2011).

20.2.1.2 Infrared

An infrared (IR) camera is a device that detects and converts light in the same way as common visible-spectrum cameras, but is sensitive to light at longer wavelengths. They form an image using infrared radiation in the spectrum at wavelengths from 5,000 nm (5 μ m) to 14,000 nm (14 μ m). Infrared cameras are used to convey a measure of thermal radiation of bodies. The intensity of each pixel can be converted for use in temperature measurement, where the brightest parts of the image (colored white) represent the warmest temperatures. Intermediate temperatures are shown as reds and yellows, and the coolest parts are shown in blue. Often, IR images are accompanied by a scale next to a false color image to relate colors to temperatures. The resolution of IR cameras (up to a maximum of 640×480 pixels) is considerably lower than the resolution of optical cameras. Furthermore, the price of IR cameras is considerably higher than their visible-spectrum counterparts.

Infrared cameras can be categorized in two main groups:

Cooled Infrared Detectors

Cooled IR detectors are normally inside a vacuum-sealed container that is cryogenically cooled. Cooling is needed for the efficient operation of the semiconductor used. Typical operating temperatures (in Kelvin degrees (K)) range from 4 to 293 K. Most modern IR detectors operate in 60–100 K range. The operating temperature is related with the type of technology used and performance expected. Cooled infrared cameras provide superior image quality compared to uncooled ones. They provide greater sensitivity that allows the use of higher F-number lenses, making high-performance long focal length lenses both smaller, and cheaper for cooled detectors. The drawbacks of cooled infrared cameras are that they are expensive both to produce and to run. Cooling is power hungry and time consuming. A camera may need several minutes to cool down before it can begin working.

Uncooled Infrared Detectors

This type of camera use sensors that are at or close to room temperature. For this reason, they do not require bulky, expensive cryogenic coolers, and therefore are smaller and cheaper than cooled ones, but with the drawback that their resolution and image quality tend to be lower than cooled detectors. This is due to a difference in their fabrication processes, limited by currently available technology.

Given the ability of IR cameras to reveal the hidden world not perceived by the human eye or visible-spectrum camera, they see application in tasks that are visibly challenging such as night vision or bad weather scenarios, inspection tasks, surveillance (bushfire monitoring), search, and rescue.

20.2.1.3 Hyperspectral Imaging

Sensors in this category acquire image data simultaneously in multiple adjacent spectral bands. This gives a wealth of data, but its processing and interpretation

require good knowledge of what specific properties are to be measured and how they relate to the actual measurement made from the sensor. For example, a single cell position in an image will have a set of brightness (or intensity) levels for each wavelength (spectral band). Different materials under examination by a hyperspectral sensor will often exhibit different intensity versus wavelength relationships. Hence, with some prior knowledge, hyperspectral image data can be useful for identifying the type or composition of materials.

Cost and complexity are the two major drawbacks of this technology. Storage is another limiting factor given the multidimensional nature of hyperspectral datasets. The availability of graphics processing units (GPU) as powerful parallel processing hardware could bring this technology a step closer to widespread adoption. However, further research efforts are needed to create analytical techniques and algorithms to unleash the full potential of hyperspectral imaging.

20.2.2 Radio-Wave Sensors

20.2.2.1 Airborne Radio Detection and Ranging (Radar)

Radar is a radio system used to determine range, altitude, direction, or speed of objects. The system transmits controlled radio pulses which are reflected back by objects. The distance to objects is estimated by measuring the signal return time. The received power declines as the fourth power of the range, assuming transmitting and receiving antennas are in the same location, hence the need for high transmitting power in most cases. Speed can be estimated by tracking the change in distance with time or by exploiting the Doppler effect (Stimson 1998).

Airborne radar has been in operation since WWII, and can be considered as an integral part of systems such as Ground Proximity Warning Systems (GPWS) or TCAS. In the automotive industry, radar is starting to appear in the form of collision warning systems (www.ford.com).

In a UAV context, the main drawback of radar is the high power consumption. The size, weight, and power (SWaP) challenges that are faced by UAVs are well known. However, new systems such as Synthetic-Aperture Radar (SAR) (Soumekh 1999) are beginning to make radar technology a feasible option onboard UAVs (Hanlon 2008).

20.2.2.2 Light Detection and Ranging (LIDAR)

LIDAR operates in a similar manner to radar, in that it can estimate distance by measuring the time of return of a signal reflected from an object. LIDARs have been widely used in atmospheric and meteorology research (Couch et al. 1991; Kiemle et al. 1997) and remote sensing (Dubayah and Drake 2000; Lefsky et al. 2002). Given its ability to provide very high definition (under 2.5 cm (Terranean 2011)), LIDAR has become a common sensor for mapping and infrastructure inspection (Yuee et al. 2009).

However, LIDAR shares many of the SWaP obstacles that are faced by radar technology. Hence, LIDARs are not often found in micro/small size UAVs, but have been flown in light general aviation aircraft (Ara 2011). Recent advances in technology have been made, an example of which is the Riegl LMS-Q160 LIDAR in <15 kg and 60 W range (Riegl 2011), which is paving the path for the use of LIDARs in smaller UAVs in the future.

The pace of innovation seems to be accelerating lately. Perhaps this is a result of the challenging economy, but without doubt, the miniaturization of electronics and the availability of small, power-efficient and computationally capable hardware will facilitate the adoption of UAVs in many civilian contexts.

20.3 Applications

20.3.1 Collision Avoidance

A sense-and-avoid capability equivalent to the human pilot see-and-avoid is one of the key prerequisites for UAVs gaining routine access to civil airspace. Sense-and-avoid begins with the detection of potential conflicting air traffic, and as a result, considerable research and development has been dedicated to addressing the “sensing” aspect of sense-and-avoid.

An important design choice in the development of sensing and detection systems is the type of sensor used to collect information about the surrounding environment; this is a choice that must account for the physical and resource limitations of the UAV platform and has implications for the target detection algorithms and other data processing techniques that are employed. Furthermore, the cooperative or uncooperative nature of the system will have also an impact in the choice of sensors. Cooperative implies that all vehicles in the air share information through common communication links. Uncooperative denotes the issue that vehicles in the sky do not communicate with each other and therefore implies that there is no other way to detect other vehicles than a self-contained passive detection system.

20.3.1.1 Vision-Based Collision Avoidance

Electro-optical (EO) sensors possess a unique combination of characteristics that make them an attractive sensor type to utilize in the sense-and-avoid application. The passive means by which EO sensors obtain information about the environment allows EO-based sense-and-avoid systems to detect other aircraft regardless of their onboard equipment. In contrast, the Traffic Alert and Collision Avoidance System (TCAS) is an active sensing approach that relies on the presence of transponder equipment onboard other aircraft (FAA 2000). Robustness against noncooperative threats is one of the key advantages of an EO sensing approach.

The maturity of EO sensor technology is also at a stage suitable for the UAV sense-and-avoid application. The current state-of-the-art EO sensors tend to be compact, light, and low power (due to their passive nature of operation), allowing them to be operationally feasible even in relatively small UAV platforms with

conservative size, weight, and power (SWaP) budgets. Furthermore, sensors supporting high-speed IEEE 1394 and IEEE 802.3-2008 (Gigabit Ethernet) communication interfaces are readily accessible as commercial off-the-shelf (COTS) products and can easily facilitate real-time capture and transfer of image data at high resolutions. Currently, the market offers four preferred standards for transferring digital video signals from the camera to the image-processing computer or workstation, FireWire (IEEE 1394), USB 2.0, Gigabit Ethernet, and Camera Link. Although not covered in this section, analog interfaces have constituted the backbone of the imaging industry, as it is known it today. They are still present in analog video links in many UAVs payloads.

The information provided by EO sensors can be exploited for purposes other than the detection and localization of targets in the image plane. Relative bearing information derived from target location on the image plane can be used to evaluate the risk of collision (constant relative bearings correspond to high risk, whereas large variations in relative bearings correspond to low risk) (Angelov et al. 2008). Furthermore, range information useful for control purposes can be derived from a single sensor when combined with appropriate aircraft maneuvers (Karlsson and Gustafsson 2005).

An EO sensing approach offers many benefits, and studies have suggested that EO-based sense-and-avoid systems hold the greatest potential for gaining regulatory approval (Karthoff et al. 2006). However, there are also many challenges associated with an EO sensing approach. One of the most significant challenges lies in the unpredictable and constantly changing nature of the airborne environment. In particular, for visible light spectrum EO sensors, detection algorithms must be able to handle a variety of image backgrounds (from blue-sky to cloud and ground clutter), lighting conditions, and possibly image artifacts (e.g., lens flare).

Another factor to contend with when using an EO sensing approach is the presence of image jitter noise. Image jitter is caused by movement of the camera sensor and hence is exacerbated in an airborne environment where the platform is in constant motion and subject to unpredictable aerodynamic disturbances. For detection algorithms that exploit target motion dynamics in the image plane, image jitter represents an undesirable noise component that can have significant impact on performance (Utt et al. 2004; Lai et al. 2011a). Various jitter compensation techniques based on aircraft state information and image features have been proposed to reduce the effects of image jitter (but it cannot be completely eliminated).

Finally, it can also be a challenge to achieve real-time processing of EO sensor image data, given the computationally intensive nature of image-processing algorithms. However, this is gradually being overcome with the development of specialized parallel processing hardware (such as graphics processing units (GPU), field-programmable gate arrays (FPGA), and dedicated digital signal processors (DSP)). Several attempts have been made, aiming to characterize the performance of these specialized parallel processing hardware (Chase et al. 2008; Thomas et al. 2009), but their utility is highly dependent on the application and its degree of parallelization.

Over the past decade, government, university, and private research groups have demonstrated EO-based sense-and-avoid technologies of varying levels of sophistication (Lai et al. 2012). One of the most mature EO-based sense-and-avoid technology programs has been conducted by a partnership between Defense Research Associates, Inc. (DRA), the Air Force Research Laboratory (AFRL), and Aeronautical Systems Center (ASC) (Utt et al. 2004, 2005). Over a number of development phases, detection and tracking of noncooperative general aviation aircraft to a range of approximate 7 nautical miles has been demonstrated onboard an Aerostar UAV. The program aims to ultimately provide a combined cooperative and noncooperative collision avoidance capability for seamless integration onto Department of Defense platforms.

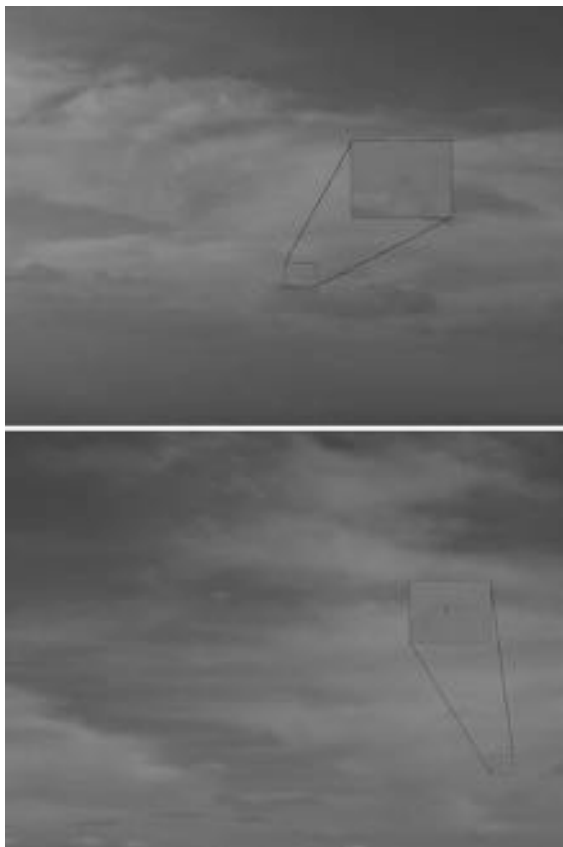
Concurrently, the Australian Research Centre for Aerospace Automation (ARCAA) has been undertaking a program to develop a cost-effective vision-based sense-and-avoid system for civilian UAV applications (Lai et al. 2011a,b). Extensive air-to-air target image data as well as nontarget data has been collected to characterize the detection range and false alarm performance of the proposed detection subsystem. Furthermore, closed-loop flight trials have been conducted, demonstrating the ability of the prototype system to automatically detect intruder aircraft and command the host aircraft autopilot to perform avoidance maneuvers. These development programs and other studies over the past decade have advanced the knowledge and understanding of trade-offs between EO sensor parameters (such as camera field of view) and system performance measures (such as detection range, detection probability, and false alarm rate) to a large extent. For example, many studies have shown that, in general, increasing the field of view reduces detection range and vice versa (Geyer et al. 2009; Lai et al. 2012).

However, the knowledge so far gained is far from complete, and more research and experimentation is required to determine the design trade-offs that will allow an EO-based sense-and-avoid system to demonstrate an equivalent level of safety to human pilot see-and-avoid. Figure 20.1 presents two examples of images taken from an onboard collision avoidance system.

20.3.1.2 Collision Avoidance Using Traffic Alert and Collision Avoidance System (TCAS)

TCAS is a system designed to reduce the incidence of air-to-air collisions and hence improve overall aircraft safety. TCAS was originally designed for manned aviation; however, its adoption for unmanned aviation is feasible and will be dictated by the degree of integration of UAVs in the civil airspace. However, its current cost (between \$25,000 and \$150,000) might prevent widespread adoption of TCAS in the UAV sector. TCAS consists of electronics that generate and interpret radio signals of the transponders from nearby aircraft. TCAS can provide auditory alarms and visual information through cockpit displays. The capabilities of the system in terms of the type of information provided to pilots and the conflict resolution strategies employed are defined by the TCAS generation (e.g., TCAS I or II). TCAS can provide *Traffic Advisories* (TA) that warn pilots of aircraft flying in the vicinity. Furthermore, the system can issue *Resolution Advisories* (RA) that suggest actions

Fig. 20.1 Images example of EO target detection taken from a collision avoidance system onboard a Cessna 172 (Greer et al. 2010). Detected aircraft is highlighted by the rectangular boxes (large box is zoomed version for clarity)



the pilot should follow to avoid a collision. These might be corrective or preventive, depending on whether a deviation in course is suggested or not, respectively. For full technical details and capabilities of TCAS, readers are referred to Harman (1989) and FAA (2000).

TCAS can only be fully effective as a means for collision avoidance if it is equipped in all aircraft in any given airspace, as one might expect with any cooperative collision avoidance system.

20.3.1.3 Automatic Dependent Surveillance-Broadcast (ADS-B) for Collision Avoidance

ADS-B is a relatively new technology that offers great promise for collision avoidance. ADS-B is not restricted to air-to-air surveillance (unlike TCAS), as it can also be utilized in air-to-ground communications with the potential to replace secondary surveillance radars. It resembles TCAS in the way that radio signals are used to transmit and receive information from nearby aircraft. One important and distinctive feature of ADS-B is the type of information exchanged. Each

aircraft would have the capability to share messages including 3D position, speed, heading, time, and intent. This information is highly valuable in the design of collision avoidance systems. However, ADS-B is often criticized for its excessive verbosity, which is not always necessary for collision avoidance. The high volume of information exchanged in ADS-B limits the number of aircraft that can use the system in a given time due to channel data bandwidth constraints. Nonetheless, the affordability of ADS-B will continue to motivate future research and development into its application for UAVs (Lai et al. 2009).

There are ADS-B systems already in the market for under U.S. \$3,000 (Kunzi 2009; NavWorx 2011). A detailed description of ADS-B systems, including its operational capabilities, can be found in RTCA (2002).

20.3.2 Remote Sensing: Power Line Inspection and Vegetation Management

Power line infrastructure is a large and costly asset to manage, costing many millions of dollars per year. Inspecting the power lines for failures (or worn, broken parts) and vegetation encroachment is a significant task particularly for electricity companies such as Ergon Energy who manages one of the largest networks in the world, consisting of over 150,000 km of power lines covering an area of 120,000 km² of Queensland, Australia (Li et al. 2012).

Traditionally, power line inspection is carried out by manned crews visiting the power lines by ground vehicle. However, this is time consuming, and due to terrain or proximity limitations, this task can be difficult. Unpiloted aircraft is an ideal technology for streamlining the inspection task carried out by ground crews. UAVs can achieve similar results to ground inspection but with greater efficiency. In many cases, UAVs can gain visibility of power lines in difficult locations which are inaccessible by ground vehicle.

Alternatively, inspecting power lines with a manned aerial inspection aircraft involves precise flying at low altitude above the infrastructure to be inspected. This is a potentially hazardous task for a human pilot due to fatigue from sitting in aircraft for long periods of time and the intense concentration and effort required to ensure that the infrastructure to inspect is within the sensor swath or field of view (usually LIDAR and high-resolution camera) (Li et al. 2012). A LIDAR and camera combination provides excellent sources of data with complementary characteristics. However, LIDAR data files are large due to the amount of data it can capture, often in the order of hundreds of Gigabytes or more. The processing of LIDAR captured by UAV is an important task which currently must be done after flight missions.

A future avenue is to perform LIDAR processing onboard the aerial vehicle. Processing may be done by a number of techniques. During the processing, one of the tasks is to distinguish the wanted feature from unwanted features in the data. For example, trees, building, and other obstacles which are not near the power line infrastructure are unwanted. Information such as the precise height, location

Fig. 20.2 Typical infrared image with 10 cm spatial resolution taken from an aircraft onboard camera (image courtesy of Ergon Energy and the Cooperative Research Centre for Spatial Information (CRCSI))



of power lines and their displacement from potentially hazardous objects such as vegetation (which may start bushfires) is among some of the wanted data to be extracted.

LIDAR and images (IR or visible spectrum) might be combined to provide a comprehensive analysis of the area under examination. Figure 20.2 shows a typical infrared image taken from an aircraft. Vegetation and other objects are clearly distinguishable; this facilitates the identification of potentially dangerous vegetation (in the vicinity of power lines). Image data can be combined in a multilayered processing approach with other sources of information as depicted in Fig. 20.3. Here, an image (top left) has been used to detect power lines, creating a region of interest around the lines; the same image after georeferencing is then merged with LIDAR data (top right) to improve feature detection rates. Typical LIDAR data representation can be seen in the middle, while in the bottom, an example representation of the merged data can be observed.

Because the task requires tracking of linear infrastructure from the air (not merely enroute waypoint-to-waypoint navigation), specialized UAV control systems may be required (Bruggemann and Ford 2011; Bruggemann et al. 2011; Li et al. 2012). Without a human pilot to control the aircraft, the stabilization and control of the platform such that the assets to be inspected fall within the field of view of the LIDAR or EO sensor becomes more challenging. The control system must also maintain steady altitude and speed as per LIDAR/camera scan/frame rate and swath/field-of-view requirements. Also, accurate spatial knowledge of the power-line infrastructure to be tracked is required, and the UAVs own flight management, navigation, and sensing systems must be capable for the task.

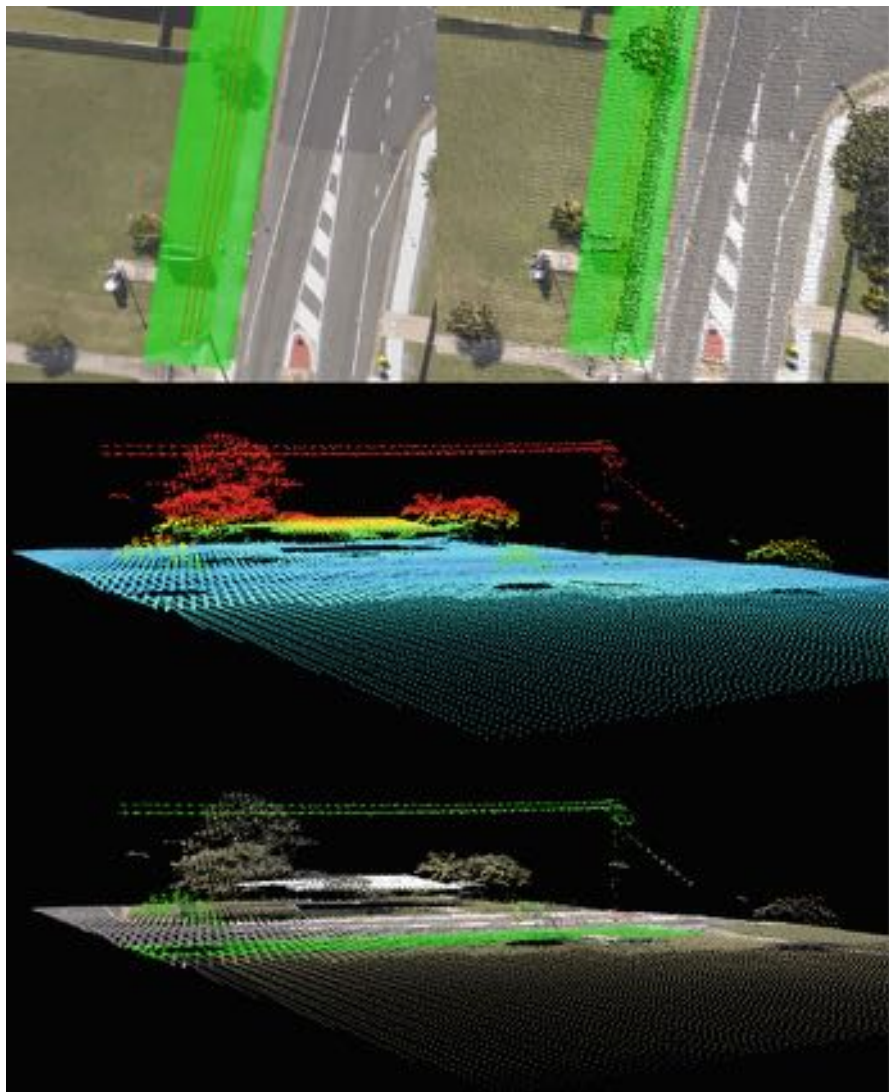


Fig. 20.3 Example of data merging image and LIDAR (Source data courtesy of Ergon Energy and the Cooperative Research Centre for Spatial Information (CRCSI))

However currently, there are a number of outstanding challenges impeding the use of UAVs for power line inspection. The UAV has to be dynamically capable to fly at low altitudes in windy or turbulent conditions and be able to carry a sizeable navigation and sensor payload system (including onboard power and LIDAR/image data storage systems) while keeping the UAV steady while flying over the power line infrastructure.

Intuitively, this means the UAV must be of reasonably large size. This is for a number of reasons. Firstly, unless the sensor payload can compensate for unwanted dynamic motion, there is a requirement for stable flight over the power lines. This means many small to medium UAVs which are sensitive to wind and turbulence at low altitudes are not suitable. Secondly, the size of the LIDAR/camera sensor payloads to capture fine details (such as tree branches and leaves, wires, and bolts) is quite large (can be in the order of 80 kg weight and of large volume). This also rules out many smaller UAV platforms. Thirdly, the use of a large UAV in civilian airspace weighing 100 kg or more presents its own technical, legal, and regulatory challenges to overcome. In future, smaller and lighter LIDAR/camera sensor payloads that have the required resolutions for power line inspection could help to facilitate the use of smaller UAV platforms for this task. This application illustrates the important coupling between the UAV platform and the sensor payload.

Conclusion

This chapter has outlined some of the most common payloads for civilian applications. Two examples of the use of sensors for missions such as collision avoidance and remote sensing were presented also. These are two examples of experimental use of sensors for civilian mission. Collision avoidance represents one of the critical capabilities for UAVs in order to allow these systems to routinely perform missions in shared airspace. In a civilian context, the remote sensing field is considered as the one that can benefit the most for UAV technology.

However, the establishment of UAVs in the civilian arena will be dictated by how the regulators define and establish requirements and guidelines for UAVs. This will affect similarly payload manufacturers, UAV manufacturers, and UAV service providers.

Overall, the UAV sector is increasingly demanding sensors that can provide more onboard autonomy, less unnecessary weight, and more ISR capabilities without an exponential increase in costs or weight. Applications such as ISR will require or emphasize sensors like cameras or signal intelligence receivers, robust radio relay, etc. Commercial payloads will emphasize very high data rate communications and sensors such as visible light and multispectral (hyperspectral) cameras.

In the near future, enduring and persistent UAVs flying for days, months, or years at high altitude will require a completely new approach to design and manufacture of payloads. These new payloads will form new communications and sensing infrastructures. UAVs flying in the upper atmosphere will behave much like satellites, but with the clear advantage that if something goes wrong the vehicle might be commanded to land, at the same time that its replacement is sent to continue the mission. This capability will be very difficult to replicate in traditional satellites. Hence, the gain is using UAVs for missions traditionally assigned to satellites.

References

- P. Angelov, C.D. Bocaniala et al., A passive approach to autonomous collision detection and avoidance, in *Proceedings of the 10th International Conference on Computer Modeling and Simulation* (IEEE computer Society, Washington, DC 2008), pp. 64–69
- ARA, Airborne Research Australia. Flinders University (2011), <http://ara.es.flinders.edu.au>.
- T.S. Bruggemann, J.J. Ford, *Compensation of Unmodeled Aircraft Dynamics in Airborne Inspection of Linear Infrastructure Assets* (The Australian Control Conference (AUCC), Melbourne, 2011)
- T.S. Bruggemann, J.J. Ford et al., Control of aircraft for inspection of linear infrastructure. *IEEE Trans. Control Syst. Technol.* **19**(6), 1409 (2011)
- Y. Cao, H. Chao et al., Autopilots for small unmanned aerial vehicles: a survey. *Int. J. Control Autom. Syst.* **8**(1), 44 (2010)
- J. Chase, B. Nelson et al., Real-time optical flow calculations on FPGA and GPU architectures: a comparison study, in *16th International Symposium on Field-Programmable Custom Computing Machines, FCCM '08* (IEEE Computer Society, Los Alamitos, 2008)
- R.H. Couch, C.W. Rowland et al., Lidar in-space technology experiment: NASA's first in-space lidar system for atmospheric research. *Opt. Eng.* **30**(1), 88–95 (1991)
- R.O. Dubayah, J.B. Drake, Lidar remote sensing for forestry. *J. For.* **98**(6), 44–46 (2000)
- D. Dusha, L. Mejias, Error analysis and attitude observability of a monocular GPS/visual odometry integrated navigation filter. *Int. J. Robot. Res.* **31**(6), 714–737 (2012)
- D. Dusha, L. Mejias et al., Fixed-wing attitude estimation using temporal tracking of the horizon and optical flow. *J. Field Robot.* **28**(2), 372 (2011)
- FAA, Introduction to TCAS II – Version 7 (2000)
- D.A. Forsyth, J. Ponce, *Computer Vision: A Modern Approach* (Prentice Hall, Upper Saddle River, 2002)
- C.M. Geyer, D. Dey et al., Prototype sense-and avoid system for UAVs. CMU-RI-TR-09-09, Robotics Institute, Carnegie Mellon University (2009)
- D. Greer, R. Mudford et al. Airborne systems laboratory for automation research, in *Proceedings of the 27th Congress of the International Council of the Aeronautical Sciences (ICAS)* Nice (Optimage, 2010)
- M.S. Grewal, L.R. Well et al. *Global Positioning Systems, Inertial Navigation, and Integration* (Wiley-Interscience, Hoboken 2007)
- M. Hanlon, ScanEagle UAV gets Synthetic Aperture Radar (SAR). Gizmag., (2008), <http://www.gizmag.com/scaneagle-uav-gets-synthetic-aperture-radar-sar/9007/>
- W.H. Harman, TCAS – A system for preventing midair collisions. *Linc. Lab. J.* **2**, 458 (1989)
- B.C. Karhoff, J.I. Limb et al., Eyes in the domestic sky: an assessment of sense and avoid technology for the Army's warrior unmanned aerial vehicle, in *IEEE Systems and Information Engineering Design Symposium, Charlottesville* (IEEE, 2006), pp. 36–42
- R. Karlsson, F. Gustafsson, Recursive Bayesian estimation: bearings-only applications. *IEE Proc. Radar Sonar Navig.* **152**(5), 313 (2005)
- C. Kiemle, G. Ehret et al., Estimation of boundary layer humidity fluxes and statistics from airborne differential absorption lidar (DIAL). *J. Geophys. Res.* **102**(D24), 29189–29203 (1997)
- F. Kunzi, Development of High User Benefit ADS-B Applications: Conflict Detection for General Aviation, MIT. International Center for Air Transportation (2009)
- C.-P. Lai, Y.-J. Ren et al., ADS-B based collision avoidance radar for unmanned aerial vehicles, in *IEEE MTT-S International Microwave Symposium Digest, 2009. MTT '09* (IEEE, Piscataway, 2009), p. 88
- J. Lai, J.J. Ford et al. Detection versus false alarm characterisation of a vision-based airborne Dim-target collision detection system. *Manuscript Accepted to International Conference on Digital Image Computing: Techniques and Applications (DICTA)* (IEEE, Washington, DC 2011a)
- J. Lai, L. Mejias et al., Airborne vision-based collision-detection system. *J. Field Robot.* **28**(2), 157 (2011b)

- J. Lai, J.J. Ford et al. Field-of-view, detection range, and false alarm trade-offs in vision-based aircraft detection. *Proceedings of the 28th Congress of the International Council of the Aeronautical Sciences (ICAS)*, Brisbane (Optimage, 2012)
- M.A. Lefsky, W.B. Cohen et al., Lidar remote sensing for ecosystem studies. *BioScience* **52**(1), 19–30 (2002)
- Z. Li, T. Bruggemann et al., Towards automated power line corridor monitoring using advanced aircraft control and multi-source feature fusion. *J. Field Robot.* **29**(1), 4–24 (2012)
- T. Merz, S. Duranti et al., Autonomous landing of an unmanned helicopter based on vision and inertial sensing, in *Experimental Robotics IX*, vol. 21 (Springer, Berlin/Heidelberg, 2006), p. 352
- NavWorx, ADS600-B Remote transceiver. NavWorx Inc. (2011), <http://www.navworx.com/>
- Riegl RIEGL Laser Measurement Systems GmbH. Retrieved Feb., 2011, from www.riegl.com (2001)
- RTCA, Minimum Aviation System Performance Standard for Automatic Dependent Surveillance Broadcast (ADS-B). Retrieved Feb, 2011, from www.rtca.org (2002)
- M. Soumekh, *Synthetic Aperture Radar Signal Processing With Matlab Algorithms* (Wiley, New York, 1999)
- G.W. Stimson, *Introduction to Airborne Radar* (SciTech Pub, Mendham, 1998)
- R. Szeliski, *Computer Vision: Algorithms and Applications* (Springer, New York/London, 2011)
- Terranean, Terranean Mapping Technologies (2011), <http://www.terranean.com.au/>
- D.B. Thomas, L. Howes et al., A comparison of CPUs, GPUs, FPGAs, and massively parallel processor arrays for random number generation, in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays* (ACM, Monterey, 2009), pp. 63–72
- J. Utt, J. McCalmont et al., Test and integration of a detect and avoid system, in *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, 20–23 Sept. 2004 (Chicago, American Institute of Aeronautics and Astronautics (AIAA), 2004)
- J. Utt, J. McCalmont et al., Development of a sense and avoid system, in *Infotech@Aerospace. American Institute of Aeronautics and Astronautics (AIAA)*, Arlington, Virginia, 26–29 Sept. 2005
- L. Wein, C. Capelle et al., GPS and stereovision-based visual odometry: application to urban scene mapping and intelligent vehicle localization. *Int. J. Veh. Technol.* (Article ID 439074) (2011)
- L. Yuee, L. Zhengrong et al., Classification of airborne LIDAR intensity data using statistical analysis and Hough transform with application to power line corridors, in *Digital Image Computing: Techniques and Applications, 2009. DICTA '09*, Melbourne, Australia (IEEE, 2009)

Mitch Bryson and Salah Sukkarieh

Contents

| | | |
|--------|---|-----|
| 21.1 | Introduction | 402 |
| 21.2 | Inertial SLAM Sensor Equations | 403 |
| 21.2.1 | Global Vs. Local SLAM | 404 |
| 21.2.2 | Inertial Localization and SLAM Process Model Equations | 405 |
| 21.2.3 | Landmark/Terrain Sensor Equations | 409 |
| 21.3 | Inertial SLAM with Range and Bearing Sensors | 411 |
| 21.3.1 | State Prediction | 411 |
| 21.3.2 | Feature Initialization | 412 |
| 21.3.3 | Data Association..... | 413 |
| 21.3.4 | State Update | 414 |
| 21.4 | Inertial SLAM with Bearing-Only Sensors | 415 |
| 21.4.1 | State Prediction and Update..... | 416 |
| 21.4.2 | Feature Initialization | 417 |
| 21.4.3 | Data Association..... | 420 |
| 21.5 | Multi-vehicle Inertial SLAM Algorithm | 423 |
| 21.5.1 | Global Vs. Local SLAM for Multiple Vehicles | 424 |
| 21.5.2 | Centralized Architectures for Multi-vehicle Inertial SLAM | 424 |
| 21.5.3 | Decentralized Architectures for Multi-vehicle Inertial SLAM | 426 |
| 21.5.4 | Delayed Observations, Network Outages, and Communication Bandwidth Constraints | 428 |
| 21.6 | Conclusion | 430 |
| | References | 430 |

M. Bryson (✉)

Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW, Australia
e-mail: m.bryson@acfr.usyd.edu.au

S. Sukkarieh

Australian Centre for Field Robotics (ACFR), School of Aerospace, Mechanical & Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW, Australia
e-mail: salah@acfr.usyd.edu.au

Abstract

This chapter provides an overview of algorithms for inertial sensor-based Simultaneous Localization and Mapping (SLAM) within the context of Unmanned Aerial Vehicles (UAVs). The presentation in this chapter is based on the use of the Extended Kalman Filter (EKF) and the Extended Information Filter (EIF) due to their ease of understanding, applicability to online implementation, and prevalence in airborne localization applications outside of SLAM (such as aided inertial localization). The discussion here includes an examination of SLAM for both small- and large-scale operation over the surface of the Earth, inertial SLAM using both range-bearing and bearing-only observations of the terrain, and a look at several different centralized and decentralized architectures for performing multi-vehicle SLAM.

21.1 Introduction

Simultaneous Localization and Mapping (SLAM) is the process of determining the position and orientation of a moving platform within an environment, while also building a map of the environment, primarily using observations of environmental features measured from and with respect to the moving platform. Since the seminal work by Smith et al. (1990), there have been several demonstrated implementations of SLAM using land (Bosse et al. 2003; Gutmann and Konolige 1999; Dissanayake et al. 2001; Guivant and Nebot 2001; Thrun and Lui 2003b) and underwater vehicles (Williams et al. 2001; Olson et al. 2004) where two-dimensional, horizontal localization and mapping is performed. In each of these implementations, SLAM involves the fusion of two particular sets of information usually coming from sensors onboard the vehicle; firstly the vehicle uses exteroceptive sensors to sense the terrain/landmarks relative to the vehicle (such as vision, laser, and radar sensors), and secondly the vehicle uses proprioceptive sensors and information to sense its own motion (such as wheel encoders, inertial sensors, and vehicle model information such as holonomic/nonholonomic constraints).

In the context of Unmanned Aerial Vehicles (UAVs), where the platform undergoes rapid six degree-of-freedom (6-DoF) motion, inertial sensors are a logical choice for the proprioceptive core of a SLAM implementation, due to their high frequency of information and ability to track platform motion in 6-DoF without relying on external information such as a vehicle dynamic model. Exteroceptive sensors (i.e., terrain sensors) come in a variety of forms depending on the application, including RADio Detection And Ranging (RADAR), LIght Detection And Ranging (LIDAR), and electro-optical sensors such as cameras. Such sensors provide a variety of different observations of the terrain including range and bearing to features and bearing-only observations of features (i.e., camera images), necessitating SLAM algorithms that account for this variability.

The utility of SLAM in the context of UAV operations can be considered from both a localization and a mapping perspective. Traditionally, airborne localization has relied on external navigation aids such as satellite/radio positioning

systems (i.e., the Global Positioning System (GPS)) or terrain/landmark maps (i.e., in terrain-aided navigation systems (TANS)). SLAM allows for intermittent localization when external navigation aids fail due to, for example, satellite signal occlusion/jamming or when operating over unknown terrain, thus improving localization system robustness. During mapping and surveying tasks, a UAV uses terrain observations (e.g., aerial photography) along with information about the UAV's own position and orientation at the time of the observation to construct a map. Consistent and accurate mapping relies on accounting for the correlation between the errors in multiple features in the map, induced by a common error in the estimate of the platform's position and orientation; SLAM maintains these relationships, computing a joint estimate of the localization and mapping states, improving accuracy in the platform's localization estimates and thus in the final map. Large-scale mapping tasks may also necessitate the use of multiple aerial platforms; algorithms for multi-vehicle inertial SLAM are a natural extension of the single-vehicle case where shared map information can be used to assist in localization across platforms and to build more accurate maps than achievable with a single UAV.

This chapter provides an overview of algorithms for inertial sensor-based SLAM within the context of UAVs. The presentation in this chapter is based on the use of the Extended Kalman Filter (EKF) and the Extended Information Filter (EIF) due to their ease of understanding, applicability to online implementation, and prevalence in airborne localization applications outside of SLAM (i.e., aided inertial localization, see Giovanni 1979; Bar-Itzhack et al. 1982; Meyer-Hilberg and Jacob 1994). The discussion here includes an examination of SLAM for both small- and large-scale operation over the surface of the Earth, inertial SLAM using both range-bearing and bearing-only observations of the terrain and a look at several different centralized and decentralized architectures for performing multi-vehicle SLAM. Section 21.2 provides an overview of the equations of motion and sensor models used in inertial SLAM with an examination of the issue of SLAM in both global and local coordinate systems and how this affects the way in which inertial SLAM is implemented. Section 21.3 examines the structure of the inertial SLAM algorithm when terrain observations are made using a range/bearing sensor. Section 21.4 examines a special case of inertial SLAM when only bearing observations of the terrain are available, such as in the case of electro-optical sensing. Section 21.5 examines the inertial SLAM algorithm when applied to multiple vehicles where vehicles share map information with each other.

21.2 Inertial SLAM Sensor Equations

The inertial SLAM algorithm is formulated using an EKF and uses information from inertial sensors and from feature observations from onboard terrain sensors in order to estimate the vehicle's position, velocity, attitude, inertial sensor biases, and map feature locations. The algorithm works on a two-stage process of state prediction (where equations for predicting the vehicle states from inertial sensor data are used)

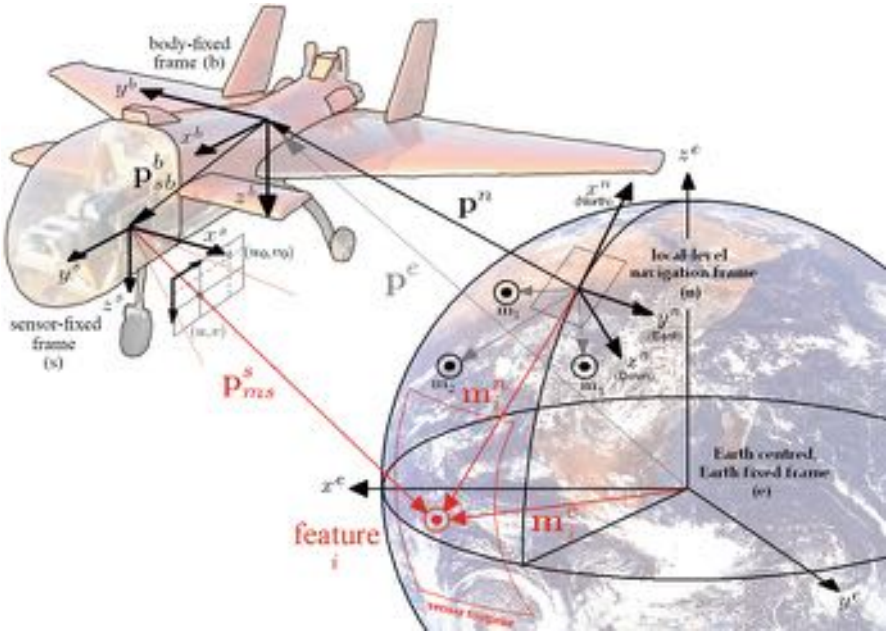


Fig. 21.1 Vectors and frames of reference in the inertial SLAM algorithm. The Earth-Centered, Earth-Fixed (ECEF) frame (e), local-level navigation frame (n), body-fixed frame (b), and sensor-fixed frame (s) and the relationship between the vehicle position (p^n, p^e), feature position (m^n, m^e), and sensor observation are shown. Also shown is the observation of a terrain feature coming from an example terrain sensor (in this case a vision camera)

and state observation/update (where equations describing the geometry of terrain feature observations are used).

This section describes the background equations behind inertial sensor-based SLAM algorithms, including equations of motion for inertial localization and sensor equations for terrain-observing sensors. Figure 21.1 illustrates the relevant relationships between the frames of reference and vectors used in the SLAM algorithm. The structure of the process and observation model equations for both global and local SLAM is discussed in the following subsections.

21.2.1 Global Vs. Local SLAM

Localization and mapping tasks vary in environmental scale depending on the application. Some tasks require knowledge of the location of each feature in the map with respect to global co-ordinates (such as an ECEF coordinate system (see Fig. 21.1)). This form of localization and mapping requires external information either in the form of global position information (such as from GPS) or global map information (i.e., the global position of one or more features in the map). Other

tasks may only require a map of the environment in which the positions of features are known only with respect to one another or with respect to the starting location of the vehicle, a task which when using SLAM does not require external map or navigation aids.

These two forms of SLAM are referred to as “global SLAM” or “local SLAM” depending on whether map and location information must be referenced to a global coordinate system or merely and local arbitrary coordinate system (such as the starting position of the UAV). This issue is particularly important when using inertial sensors as inertial localization relies to some degree on global information about the local coordinate system’s motion with the rotation of the Earth. Local SLAM using inertial sensors is therefore only possible when certain assumptions are made which treat the local map coordinates as an inertial frame of reference by ignoring the effect of the Earth’s rotation.

21.2.2 Inertial Localization and SLAM Process Model Equations

The presentation here begins by considering the process model equations from a global SLAM perspective. In this case, the location of the vehicle and the terrain is estimated with respect to the center of the Earth. The estimated state vector $\hat{\mathbf{x}}(k)$ at time step k in this case contains the three-dimensional vehicle position (\mathbf{p}^e), velocity (\mathbf{v}^e) and Euler angles ($\Psi^n = [\phi, \theta, \psi]^T$), IMU sensor biases ($\delta \mathbf{f}^b$ and $\delta \omega_{ib}^b$), and the N three-dimensional feature locations (\mathbf{m}_i^e) in the environment:

$$\hat{\mathbf{x}}(k) = [\mathbf{p}^e(k), \mathbf{v}^e(k), \Psi^n(k), \delta \mathbf{f}^b(k), \delta \omega_{ib}^b(k), \mathbf{m}_1^e(k), \mathbf{m}_2^e(k), \dots, \mathbf{m}_N^e(k)]^T \quad (21.1)$$

where $i = 1, \dots, N$, the superscript e indicates Earth-Centered, Earth-Fixed (ECEF) frame referenced vectors, the superscript b indicates body-fixed frame referenced vectors and the superscript n indicates the Euler angles Ψ^n parameterize the body-fixed to local navigation frame Direction Cosine Matrix (DCM) transformation \mathbf{C}_b^n (see Chapter 3 of this book for details on frames of reference). Figure 21.1 illustrates the relevant relationships between the frames of reference and vectors used in the SLAM algorithm.

Euler angles are used to represent the attitude of the platform rather than quaternions in order to reduce the number of parameters in the estimator. It is assumed that the vehicle’s pitch angle is constrained so as to avoid the Euler angle singularity at $\theta = 90^\circ$; however, the process model could be adapted to implement quaternions for the parameterization of \mathbf{C}_b^n in the event that the platform motion is not constrained in such a way.

The state estimate $\hat{\mathbf{x}}$ is predicted forward in time by integrating the first-order nonlinear dynamic equation:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}_c[\hat{\mathbf{x}}(t), \mathbf{u}(t)] + \mathbf{G}_c[\mathbf{w}(t)] \quad (21.2)$$

where $\mathbf{F}_c[.,.]$ is the continuous-time process model function, $\mathbf{G}_c[.]$ is the continuous-time input function, $\mathbf{u}(t)$ is the system input (which is comprised of inertial sensor readings), and $\mathbf{w}(t)$ is uncorrelated, zero-mean vehicle process noise vector of covariance \mathbf{Q} (which is comprised of inertial sensor noise errors). The process model equations for the vehicle position, velocity, and attitude are based on the 6-DoF inertial localization equations in which an Earth-frame mechanization (Titterton and Weston 1997) is applied:

$$\dot{\mathbf{p}}^e = \mathbf{v}^e \quad (21.3)$$

$$\dot{\mathbf{v}}^e = \mathbf{C}_b^e \hat{\mathbf{f}}^b - \mathbf{C}_b^e \delta \mathbf{f}^b - \mathbf{C}_b^e \mathbf{w}_{\text{accel}} - 2(\omega_{ie}^e \times \mathbf{v}^e) + \mathbf{g}_l^e \quad (21.4)$$

$$\dot{\Psi}^n = \mathbf{E}_b^n (\hat{\omega}_{ib}^b - \delta \omega_{ib}^b - \mathbf{w}_{\text{gyro}} - \mathbf{C}_n^b \omega_{ie}^n) \quad (21.5)$$

where ω_{ie}^e and ω_{ie}^n are the Earth's rotation rate vectors measured in the ECEF and local navigation frames, respectively:

$$\omega_{ie}^e = \begin{bmatrix} 0 \\ 0 \\ \omega_{\text{Earth}} \end{bmatrix} \quad (21.6)$$

$$\omega_{ie}^n = \begin{bmatrix} \omega_{\text{Earth}} \cos(\lambda) \\ 0 \\ -\omega_{\text{Earth}} \sin(\lambda) \end{bmatrix} \quad (21.7)$$

where λ is the latitude angle of the vehicle and $\omega_{\text{Earth}} = 7.292115 \times 10^{-5}$ rad/s is the rotation rate of the Earth. \mathbf{E}_b^n is the body to navigation frame rotation rate transformation matrix:

$$\mathbf{E}_b^n = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (21.8)$$

\mathbf{g}_l^e is the local gravity term:

$$\mathbf{g}_l^e = \mathbf{g}^e - \omega_{ie}^e \times (\omega_{ie}^e \times \mathbf{p}^e) \quad (21.9)$$

$$= -g \begin{bmatrix} \frac{\mathbf{p}_x^e}{|\mathbf{p}^e|} \\ \frac{\mathbf{p}_y^e}{|\mathbf{p}^e|} \\ \frac{\mathbf{p}_z^e}{|\mathbf{p}^e|} \end{bmatrix} - [\times \omega_{ie}^e]^2 \mathbf{p}^e \quad (21.10)$$

where g is the acceleration acting on the vehicle due to gravity, referenced in ECEF coordinates and $[\times \omega_{ie}^e]$ is the skew symmetric matrix of the Earth's rotation rate vector. \mathbf{C}_b^n is the DCM transformation from the body to local navigation frame, and \mathbf{C}_b^e is the DCM transformation from the body to ECEF frame which are related via

$$\mathbf{C}_b^e = \mathbf{C}_n^e \mathbf{C}_b^n \quad (21.11)$$

$$\mathbf{C}_n^e = \begin{bmatrix} -\sin(\lambda) \cos(l) & -\sin(l) & -\cos(\lambda) \cos(l) \\ -\sin(\lambda) \sin(l) & \cos(l) & -\cos(\lambda) \sin(l) \\ \cos(\lambda) & 0 & -\sin(\lambda) \end{bmatrix} \quad (21.12)$$

where l is the longitude angle of the vehicle's position. The vectors $\hat{\mathbf{f}}^b$ and $\hat{\omega}_{ib}^b$ are the accelerometer specific force vector reading and gyroscope rotation rate reading, respectively (where $\mathbf{u}(t) = [\hat{\mathbf{f}}^b(t), \hat{\omega}_{ib}^b(t)]^T$), $\delta \mathbf{f}^b$ and $\delta \omega_{ib}^b$ are the accelerometer and gyro biases, respectively, and $\mathbf{w}_{\text{accel}}$ and \mathbf{w}_{gyro} are the accelerometer and gyro noise values. The true acceleration and rotation rates, \mathbf{f}^b and ω_{ib}^b , are defined:

$$\mathbf{f}^b = \hat{\mathbf{f}}^b - \delta \mathbf{f}^b - \mathbf{w}_{\text{accel}} \quad (21.13)$$

$$\omega_{ib}^b = \hat{\omega}_{ib}^b - \delta \omega_{ib}^b - \mathbf{w}_{\text{gyro}} \quad (21.14)$$

Accelerometer and gyro biases are known to fluctuate by very small amounts due to temperature changes and bias wander, particularly in low-cost IMUs. The accelerometer and gyro bias process models are given by

$$\delta \dot{\mathbf{f}}^b = \mathbf{w}_{b,\text{accel}} \quad (21.15)$$

$$\delta \dot{\omega}_{ib}^b = \mathbf{w}_{b,\text{accel}} \quad (21.16)$$

where $\mathbf{w}_{b,\text{accel}}$ and $\mathbf{w}_{b,\text{gyro}}$ are bias drift noises for the accelerometers and gyros, respectively. The complete noise vector $\mathbf{w}(t)$ for the process model based on errors from the IMU is composed:

$$\mathbf{w}(t) = [\mathbf{w}_{\text{accel}}(t), \mathbf{w}_{\text{gyro}}(t), \mathbf{w}_{b,\text{accel}}(t), \mathbf{w}_{b,\text{gyro}}(t)]^T \quad (21.17)$$

Finally, map feature locations are estimated in the ECEF frame and are also assumed to be constant as the state of the terrain is stationary and the process model of the i th feature is given by

$$\dot{\mathbf{m}}_i^e = \mathbf{0} \quad (21.18)$$

Equations 21.3–21.5, 21.15, 21.16, and 21.18 can also be expressed in discrete-time, recursive form by assuming a first-order Euler integration step:

$$\hat{\mathbf{x}}(k) = \mathbf{F}[\hat{\mathbf{x}}(k-1), \mathbf{u}(k), k] + \mathbf{G}[\mathbf{w}(k)] \quad (21.19)$$

where $\mathbf{F}[\cdot, \cdot, k]$ is the nonlinear state transition function at time k and $\mathbf{G}[\cdot, k]$ is the input model transition function at time k . The discrete process model thus becomes

$$\mathbf{p}^e(k) = \mathbf{p}^e(k-1) + \mathbf{v}^e \Delta t \quad (21.20)$$

$$\mathbf{v}^e(k) = \mathbf{v}^e(k-1) + [\mathbf{C}_b^e \hat{\mathbf{f}}^b - \mathbf{C}_b^e \delta \mathbf{f}^b - \mathbf{C}_b^e \mathbf{w}_{\text{accel}} - 2(\omega_{ie}^e \times \mathbf{v}^e) + \mathbf{g}_l^e] \Delta t \quad (21.21)$$

$$\Psi^n(k) = \Psi^n(k-1) + [\mathbf{E}_b^n (\hat{\omega}_{ib}^b - \delta \omega_{ib}^b - \mathbf{w}_{\text{gyro}} - \mathbf{C}_n^b \omega_{ie}^n)] \Delta t \quad (21.22)$$

$$\delta \mathbf{f}^b(k) = \delta \mathbf{f}^b(k-1) + \mathbf{w}_{b,\text{accel}}(k) \quad (21.23)$$

$$\delta \omega_{ib}^b(k) = \delta \omega_{ib}^b(k-1) + \mathbf{w}_{b,\text{gyro}}(k) \quad (21.24)$$

$$\mathbf{m}_i^e(k) = \mathbf{m}_i^e(k-1) \quad (21.25)$$

where Δt is the time difference between the k and $k-1$ discrete-time segments.

21.2.2.1 Process Model Equation Approximations for Local SLAM

When performing SLAM with respect to a local coordinate system where no global information is available, several approximations must be made in the inertial SLAM equations. In local SLAM, the position and orientation of the vehicle is maintained with respect to a local navigation frame which is fixed to an arbitrary and unknown location on the Earth's surface rather than the ECEF frame. The local SLAM equations are thus derived under the assumption that local navigation frame is an inertial frame of reference by ignoring the small Coriolis and centripetal accelerations which are incurred by the Earth's rotation.

In local SLAM, the estimated state vector is $\hat{\mathbf{x}}_{\text{local}}(k)$, in which the vehicle position and velocity and the position of map features are now reference in local navigation frame coordinates:

$$\hat{\mathbf{x}}_{\text{local}}(k) = [\mathbf{p}^n(k), \mathbf{v}^n(k), \Psi^n(k), \delta \mathbf{f}^b(k), \delta \omega_{ib}^b(k), \mathbf{m}_1^n(k), \mathbf{m}_2^n(k), \dots, \mathbf{m}_N^n(k)]^T \quad (21.26)$$

The continuous-time and discrete-time process models for local SLAM are thus given by

$$\dot{\hat{\mathbf{x}}}_{\text{local}}(t) = \mathbf{F}_{c,\text{local}}[\hat{\mathbf{x}}_{\text{local}}(t), \mathbf{u}(t)] + \mathbf{G}_{c,\text{local}}[\mathbf{w}(t)] \quad (21.27)$$

$$\hat{\mathbf{x}}_{\text{local}}(k) = \mathbf{F}_{\text{local}}[\hat{\mathbf{x}}_{\text{local}}(k-1), \mathbf{u}(k), k] + \mathbf{G}_{\text{local}}[\mathbf{w}(k)] \quad (21.28)$$

The vehicle process model equations in continuous-time form are the following:

$$\dot{\mathbf{p}}^n = \mathbf{v}^n \quad (21.29)$$

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \hat{\mathbf{f}}^b - \mathbf{C}_b^n \delta \mathbf{f}^b - \mathbf{C}_b^n \mathbf{w}_{\text{accel}} + \mathbf{g}^n \quad (21.30)$$

$$\dot{\Psi}^n = \mathbf{E}_b^n (\hat{\omega}_{ib}^b - \delta \omega_{ib}^b - \mathbf{w}_{\text{gyro}}) \quad (21.31)$$

where $\mathbf{g}^n = [0, 0, g]^T$ is the vector of acceleration due to gravity in the local navigation frame and $g = 9.81 \text{ m/s}^2$. Terrain map features are now referenced in local navigation frame coordinates, and their process model is given by

$$\dot{\mathbf{m}}_i^n = \mathbf{0} \quad (21.32)$$

The complete form of the discrete-time process model equations in local SLAM are thus

$$\mathbf{p}^n(k) = \mathbf{p}^n(k-1) + \mathbf{v}^n \Delta t \quad (21.33)$$

$$\mathbf{v}^n(k) = \mathbf{v}^n(k-1) + [\mathbf{C}_b^n \hat{\mathbf{f}}^b - \mathbf{C}_b^n \delta \mathbf{f}^b - \mathbf{C}_b^n \mathbf{w}_{\text{accel}} + \mathbf{g}^n] \Delta t \quad (21.34)$$

$$\Psi^n(k) = \Psi^n(k-1) + [\mathbf{E}_b^n (\hat{\omega}_{ib}^b - \delta \omega_{ib}^b - \mathbf{w}_{\text{gyro}})] \Delta t \quad (21.35)$$

$$\delta \mathbf{f}^b(k) = \delta \mathbf{f}^b(k-1) + \mathbf{w}_{b, \text{accel}}(k) \quad (21.36)$$

$$\delta \omega_{ib}^b(k) = \delta \omega_{ib}^b(k-1) + \mathbf{w}_{b, \text{gyro}}(k) \quad (21.37)$$

$$\mathbf{m}_i^n(k) = \mathbf{m}_i^n(k-1) \quad (21.38)$$

21.2.3 Landmark/Terrain Sensor Equations

The observation model equations describe the relationship between the sensor observation of a map feature in the sensor coordinates of the terrain sensor to the estimated states in SLAM. Figure 21.1 illustrates the relevant relationships between the sensor frame, sensor observation and map, and vehicle positions. The observation $\mathbf{z}_i(k)$ is related to the estimated states using Eq. 21.39 for global SLAM and Eq. 21.40 for local SLAM:

$$\mathbf{z}_i(k) = \mathbf{H}_i(\mathbf{p}^e(k), \Psi^n(k), \mathbf{m}_i^e(k), k) + \mathbf{v}(k) \quad (21.39)$$

$$\mathbf{z}_i(k) = \mathbf{H}_{i, \text{local}}(\mathbf{p}^n(k), \Psi^n(k), \mathbf{m}_i^n(k), k) + \mathbf{v}(k) \quad (21.40)$$

where $\mathbf{H}_i(\cdot, \cdot, \cdot, k)$ and $\mathbf{H}_{i, \text{local}}(\cdot, \cdot, \cdot, k)$ are functions of the feature location, vehicle position, and Euler angles and $\mathbf{v}(k)$ is uncorrelated, zero-mean observation noise errors of covariance \mathbf{R} .

Terrain observations could come from a variety of different sensors such as RADAR, LIDAR, or an electro-optical camera. The SLAM algorithm requires that point features can be extracted from the observation sensor data which go on to be mapped terrain features. In the case of large objects that do not show up as a “point” in sensor data, the centroid of the object is found or else several points can be used to represent a single object. Example feature extraction algorithms for vision include Scale-Invariant Feature Transform (SIFT) features (Lowe 2004) or model-based feature matching (Nixon and Aguado 2001). Features in this sense are points in the sensor data that are distinct and easily recognizable or else points in

the sensor data that appear to correlate well with a given feature model or template that is specified offline.

Sensor observations can be broken up into two types: firstly range/bearing observations (as might be available from RADAR or a LIDAR) and secondly bearing-only observations (as might be available from an electro-optical sensor such as a camera). When both range and bearing observations are made, the observation model is given by

$$\mathbf{z}_i(k) = \begin{bmatrix} \rho_i \\ \varphi_i \\ \vartheta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i^s)^2 + (y_i^s)^2 + (z_i^s)^2} \\ \tan^{-1} \left(\frac{y_i^s}{x_i^s} \right) \\ \tan^{-1} \left(\frac{z_i^s}{\sqrt{(x_i^s)^2 + (y_i^s)^2}} \right) \end{bmatrix} \quad (21.41)$$

where ρ_i , φ_i , and ϑ_i are the observed range, azimuth, and elevation angles to the feature and x_i^s , y_i^s , and z_i^s are the Cartesian coordinates of $\mathbf{p}_{ms,i}^s$, the relative position of the i th feature with respect to the sensor, measured in the sensor frame.

There are two forms that can be used to represent a bearing-only observation. The observation $\mathbf{z}_i(k)$ can be represented by azimuth (φ_i) and elevation angles (ϑ_i):

$$\mathbf{z}_{i,\text{ang}}(k) = \begin{bmatrix} \varphi_i \\ \vartheta_i \end{bmatrix} = \begin{bmatrix} \tan^{-1} \left(\frac{y_i^s}{x_i^s} \right) \\ \tan^{-1} \left(\frac{z_i^s}{\sqrt{(x_i^s)^2 + (y_i^s)^2}} \right) \end{bmatrix} \quad (21.42)$$

where \mathbf{R}_{ang} is the angular noise covariance. For vision camera, the observation is better represented as pixels in the image of the camera, using a pinhole camera model:

$$\mathbf{z}_{i,\text{pix}}(k) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u \left(\frac{y_i^s}{x_i^s} \right) + u_0 \\ f_v \left(\frac{z_i^s}{x_i^s} \right) + v_0 \end{bmatrix} \quad (21.43)$$

where u_0 , v_0 , f_u , and f_v are calibration parameters for the camera and the pixel noise covariance is \mathbf{R}_{pix} . The relationship between the pixel coordinates and the azimuth and elevation angles is given by Eq. 21.44

$$\begin{bmatrix} \varphi \\ \vartheta \end{bmatrix} = \begin{bmatrix} \tan^{-1} \left(\frac{(u-u_0)}{f_u} \right) \\ \tan^{-1} \left(\frac{(v-v_0)}{f_v} \cos \varphi \right) \end{bmatrix} \quad (21.44)$$

$\mathbf{p}_{ms,i}^s$, the relative position of the i th feature with respect to the sensor, measured in the sensor frame is given by:

$$\begin{aligned} \mathbf{p}_{ms,i}^s &= \mathbf{C}_b^s \mathbf{C}_n^b [\mathbf{m}_i^n - \mathbf{p}^n - \mathbf{C}_b^n \mathbf{p}_{sb}^b] \\ &= \mathbf{C}_b^s \mathbf{C}_e^b [\mathbf{m}_i^e - \mathbf{p}^e - \mathbf{C}_b^e \mathbf{p}_{sb}^b] \end{aligned} \quad (21.45)$$

where \mathbf{C}_b^s is the transformation matrix from the body frame to the sensor frame and \mathbf{p}_{sb}^b is the sensor offset from the vehicle center of mass, measured in the body frame, otherwise known as the “lever arm” and $\mathbf{C}_e^b = (\mathbf{C}_b^e)^T$.

21.3 Inertial SLAM with Range and Bearing Sensors

When both range and bearing observations to features are available, the inertial SLAM algorithm is broken up into the following processes:

1. *State Prediction.* The process model equations in Sect. 21.2.2 are used to “predict” forward in time the estimated state vector and state covariance matrix in an EKF prediction step.
2. *Data Association.* Every time a new feature observation is made, the data association step is used to determine whether the observation is of a new feature or a known feature, and which feature it is of.
3. *Feature Initialization.* When a feature is observed by the terrain sensor for the first time, its position is computed and augmented into the estimated state vector and its initial position uncertainty augmented into the state covariance matrix.
4. *State Update.* Each time an observation is made of a previously seen feature, it is used in an EKF update step to correct the value of the estimated state and update the state covariance matrix.

Figure 21.2 provides an overview of the range/bearing inertial SLAM algorithm. At each time step, inertial sensor data is used for the prediction stage. When features are extracted from the sensor data, they are run through a data association stage. New features are augmented into the state vector and state covariance matrix, and observations of existing features are used to update the estimated state vector and state covariance matrix. The following subsections summarize the equations and methods in each step of the algorithm. For further details of inertial SLAM with range and bearing sensors and an implementation example, the reader is referred to Kim and Sukkarieh (2003).

21.3.1 State Prediction

The state prediction stage is run recursively each time a new reading is taken from the inertial sensors. In the case of global SLAM, the estimated state vector $\hat{\mathbf{x}}(k+1)$ is predicted from the previous time step state estimate using Eqs. 21.20–21.25. The state covariance $\mathbf{P}^-(k+1)$, which is the state covariance at time step k after prediction, is computed as

$$\mathbf{P}^-(k+1) = \nabla \mathbf{F} \mathbf{P}(k) \nabla \mathbf{F}^T + \nabla \mathbf{G} \mathbf{Q}(k) \nabla \mathbf{G}^T \quad (21.46)$$

where $\nabla \mathbf{F}$ and $\nabla \mathbf{G}$ are the Jacobians of the state transition function $\mathbf{F}[\cdot, \cdot, k]$ with respect to the state vector $\hat{\mathbf{x}}(k+1)$ and input model transition function $\mathbf{G}[\cdot, k]$ with

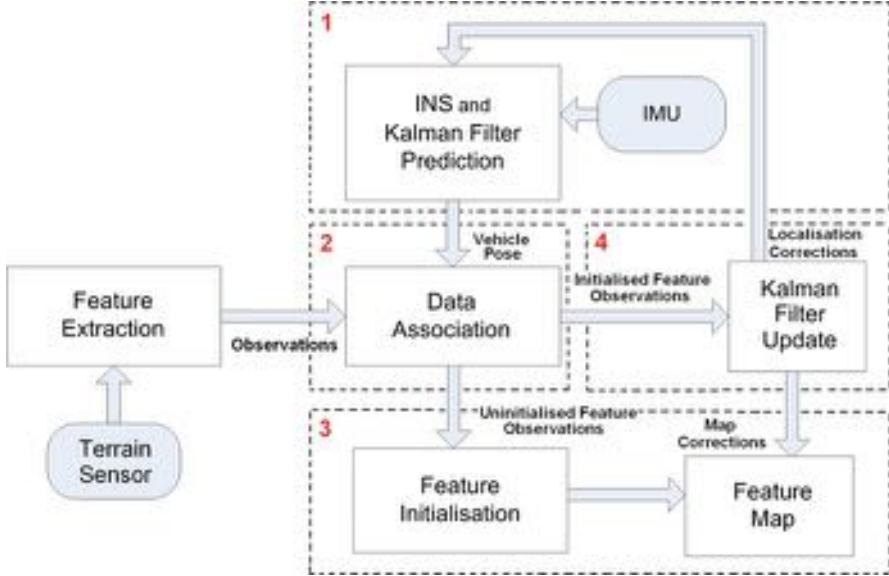


Fig. 21.2 Overview of the range/bearing inertial SLAM algorithm. the algorithm is recursive; EKF state prediction (1) is performed when inertial sensor data is available. Data association (2), feature initialization (3), and EKF state update (4) are performed when terrain sensor observations are made

respect to the noise input $\mathbf{w}(k+1)$, respectively. In the case of local SLAM, the estimated state vector $\hat{\mathbf{x}}_{\text{local}}(k)$ is predicted forward using Eqs. 21.33–21.38. The state covariance $\mathbf{P}^-(k+1)$ is computed as

$$\mathbf{P}^-(k+1) = \nabla \mathbf{F}_{\text{local}} \mathbf{P}(k) \nabla \mathbf{F}_{\text{local}}^T + \nabla \mathbf{G}_{\text{local}} \mathbf{Q}(k) \nabla \mathbf{G}_{\text{local}}^T \quad (21.47)$$

where $\nabla \mathbf{F}_{\text{local}}$ and $\nabla \mathbf{G}_{\text{local}}$ are the Jacobians of the state transition function $\mathbf{F}_{\text{local}}[\cdot, \cdot, k]$ with respect to the state vector $\hat{\mathbf{x}}_{\text{local}}(k+1)$ and input model transition function $\mathbf{G}_{\text{local}}[\cdot, k]$ with respect to the noise input $\mathbf{w}(k+1)$, respectively.

21.3.2 Feature Initialization

When the first range/bearing observation of a particular feature is obtained, its position is calculated using the initialization function $\mathbf{J}_1[\hat{\mathbf{x}}(k), \mathbf{J}_2(\mathbf{z}_i(k))]$ which is given as

$$\mathbf{J}_1[\hat{\mathbf{x}}(k), \mathbf{J}_2(\mathbf{z}_i(k))] \longrightarrow \mathbf{m}_i^e = \mathbf{p}^e + \mathbf{C}_n^e \mathbf{C}_b^n \mathbf{p}_{sb}^b + \mathbf{C}_n^e \mathbf{C}_b^n \mathbf{C}_s^b \mathbf{p}_{ms}^s \quad (21.48)$$

for the case of global SLAM and

$$\mathbf{J}_{1,\text{local}}[\hat{\mathbf{x}}_{\text{local}}(k), \mathbf{J}_2(\mathbf{z}_i(k))] \longrightarrow \mathbf{m}_i^n = \mathbf{p}^n + \mathbf{C}_b^n \mathbf{p}_{sb}^b + \mathbf{C}_b^n \mathbf{C}_s^b \mathbf{p}_{ms}^s \quad (21.49)$$

for the case of local SLAM, where

$$\mathbf{J}_2(\mathbf{z}_i(k)) \longrightarrow \mathbf{p}_{ms,i}^s = \begin{bmatrix} \rho_i \cos(\varphi_i) \cos(\vartheta_i) \\ \rho_i \sin(\varphi_i) \cos(\vartheta_i) \\ \rho_i \sin(\vartheta_i) \end{bmatrix} \quad (21.50)$$

for both the global and local SLAM cases. The state vector and covariance are then augmented to include the new feature position:

$$\hat{\mathbf{x}}_{\text{aug}}(k) = \begin{bmatrix} \hat{\mathbf{x}}(k) \\ \mathbf{m}_i^e(k) \end{bmatrix} \quad (21.51)$$

$$\mathbf{P}_{\text{aug}}(k) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \nabla \mathbf{J}_x & \nabla \mathbf{J}_z \end{bmatrix} \begin{bmatrix} \mathbf{P}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \nabla \mathbf{J}_x & \nabla \mathbf{J}_z \end{bmatrix}^T \quad (21.52)$$

for the case of global SLAM and

$$\hat{\mathbf{x}}_{\text{local,aug}}(k) = \begin{bmatrix} \hat{\mathbf{x}}_{\text{local}}(k) \\ \mathbf{m}_i^n(k) \end{bmatrix} \quad (21.53)$$

$$\mathbf{P}_{\text{local,aug}}(k) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \nabla \mathbf{J}_x & \nabla \mathbf{J}_z \end{bmatrix} \begin{bmatrix} \mathbf{P}_{\text{local}}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \nabla \mathbf{J}_x & \nabla \mathbf{J}_z \end{bmatrix}^T \quad (21.54)$$

for the case of local SLAM where $\nabla \mathbf{J}_x$ and $\nabla \mathbf{J}_z$ are the Jacobians of the initialization function \mathbf{J}_1 with respect to the state estimate $\hat{\mathbf{x}}(k)$ and the observation $\mathbf{z}_i(k)$, respectively. The position of this feature becomes correlated to all of the vehicle states including position, velocity, and attitude along with inertial sensor biases and also to the position of other features in the map.

21.3.3 Data Association

Data association is the process of matching observations of features from the terrain sensor with the estimated 3D position of the feature within the map. The validity of potential associations between observations and features is assessed using the Mahalanobis distance (γ) (Neira and Tardos 2001) in the sensor space (range, azimuth, and elevation):

$$\gamma_i = v_i(k)^T \mathbf{S}_i(k)^{-1} v_i(k) \quad (21.55)$$

where $v_i(k)$ and $\mathbf{S}_i(k)$ are the innovation and innovation covariance (see Sect. 21.3.4) for an observation of the i th feature in the map at time segment k .

When checking the association of a given observation with the initialized features in the map, γ_i is calculated for each initialized feature. Matchings that fall within a defined threshold of γ_i corresponding to a 95 % level of confidence are considered

acceptable, in which case the observation is associated to a known feature in the map and an EKF state update is performed (see Sect. 21.3.4 below). If the observation does not fall within the threshold of an existing map feature, it is assumed that the feature has not been seen before, and thus the observation data is used to initialize the feature into the map as is shown in Sect. 21.3.2.

This method of data association works well when feature observations are well spaced in the sensor coordinates with respect to the uncertainty of the vehicle position and orientation. In the case of very dense feature observations, it may be necessary to consider not only matches of individual features to individual sensor observations but also the joint compatibility of several features to several observations simultaneously as to overcome association ambiguity. For efficient techniques for joint compatibility data association based on the innovation gate in Eq. 21.55, the reader is referred to Neira and Tardos (2001).

21.3.4 State Update

Once a feature has been initialized into the state vector, subsequent observations of this feature are used to update the entire state vector consisting of the vehicle pose, velocity, inertial sensor biases, and the position of this feature and other features in the environment. The state estimate is updated in an EKF update step where the updated state estimate and state covariance matrix after the update are as follows:

$$\hat{\mathbf{x}}^+(k+1) = \hat{\mathbf{x}}^-(k+1) + \mathbf{W}(k+1)\mathbf{v}(k+1) \quad (21.56)$$

$$\mathbf{P}^+(k+1) = \mathbf{P}^-(k+1) - \mathbf{W}(k+1)\mathbf{S}_i(k+1)\mathbf{W}(k+1)^T \quad (21.57)$$

$$\mathbf{v}_i(k+1) = \mathbf{z}_i - \mathbf{H}_i(\hat{\mathbf{x}}^-(k+1)) \quad (21.58)$$

$$\mathbf{S}_i(k+1) = \nabla \mathbf{H}_i \mathbf{P}^-(k+1) \nabla \mathbf{H}_i^T + \mathbf{R}(k+1) \quad (21.59)$$

$$\mathbf{W}(k+1) = \mathbf{P}^-(k+1) \nabla \mathbf{H}_i^T \mathbf{S}_i^{-1}(k+1) \quad (21.60)$$

for the case of global SLAM and

$$\hat{\mathbf{x}}_{\text{local}}^+(k+1) = \hat{\mathbf{x}}_{\text{local}}^-(k+1) + \mathbf{W}(k+1)\mathbf{v}(k+1) \quad (21.61)$$

$$\mathbf{P}_{\text{local}}^+(k+1) = \mathbf{P}_{\text{local}}^-(k+1) - \mathbf{W}(k+1)\mathbf{S}_i(k+1)\mathbf{W}(k+1)^T \quad (21.62)$$

$$\mathbf{v}_i(k+1) = \mathbf{z}_i - \mathbf{H}_{i,\text{local}}(\hat{\mathbf{x}}_{\text{local}}^-(k+1)) \quad (21.63)$$

$$\mathbf{S}_i(k+1) = \nabla \mathbf{H}_{i,\text{local}} \mathbf{P}_{\text{local}}^-(k+1) \nabla \mathbf{H}_{i,\text{local}}^T + \mathbf{R}(k+1) \quad (21.64)$$

$$\mathbf{W}(k+1) = \mathbf{P}_{\text{local}}^-(k+1) \nabla \mathbf{H}_{i,\text{local}}^T \mathbf{S}_i^{-1}(k+1) \quad (21.65)$$

for the case of local SLAM where $\mathbf{H}_i(\hat{\mathbf{x}}^-(k+1))$ or $\mathbf{H}_{i,\text{local}}(\hat{\mathbf{x}}_{\text{local}}^-(k+1))$ is the predicted feature observation which is computed from the estimated vehicle

position and attitude and estimate map location using Eqs. 21.41 and 21.45. $\nabla \mathbf{H}_i$ and $\nabla \mathbf{H}_{i,\text{local}}$ are the Jacobians of the observation function with respect to the predicted state vector. Once a feature leaves the field of view of the sensor, its position remains in the state vector and continues to be updated via its correlations to other visible features in the state vector.

21.4 Inertial SLAM with Bearing-Only Sensors

When bearing-only observations are made using a terrain sensor such as in the case of a vision system, additional elements must be added to the SLAM algorithm. Performing SLAM with bearing-only observations poses two main additional challenges to the range and bearing case. Firstly, a single bearing-only observation provides insufficient information alone to localize a feature in 3D. Instead observations from two sufficiently different poses are required. Secondly, data association is complicated by bearing-only observations. Since the 3D position of the feature is not known from a single observation, the Mahalanobis distance (Neira and Tardos 2001), commonly used for validation gating in tracking tasks, and as shown for the range-bearing case in Sect. 21.3.3, cannot be calculated in the standard way.

This section details variations to the inertial SLAM algorithms that account for complications arising from the use of bearing-only terrain sensors. The bearing-only inertial SLAM algorithm is broken up into the following processes:

1. *State Prediction.* The prediction step in the bearing-only case is performed in a similar manner as in the range/bearing case.
2. *Data Association.* Data association for initialized features follows the same methods as shown in the range/bearing case. Data association of uninitialized features is tackled by creating multi-hypothesis distributions of the possible feature locations in 3D (i.e., along the line of sight of an observation). Subsequent observations of the same feature can be associated by matching the most likely hypotheses and culling the hypotheses that do not match.
3. *Feature Initialization.* A delayed initialization technique is used to store information from bearing-only observations until there exists two observations with a sufficient baseline from which to initialize the 3D position of the feature. Once this is available, all of the information contained in stored observations is recovered.
4. *State Update.* The update step in the bearing-only case is performed in a similar manner as in the range/bearing case where bearing-only observations are used to update the estimated state which includes vehicle states, stored pose data, and features whose 3D positions have been initialized into the map.

Figure 21.3 provides an overview of the bearing-only inertial SLAM algorithm. The following subsections summarize the equations and methods in each step of the algorithm. For further details of inertial SLAM with bearing-only sensors and an implementation example, the reader is referred to Bryson and Sukkarieh (2007).

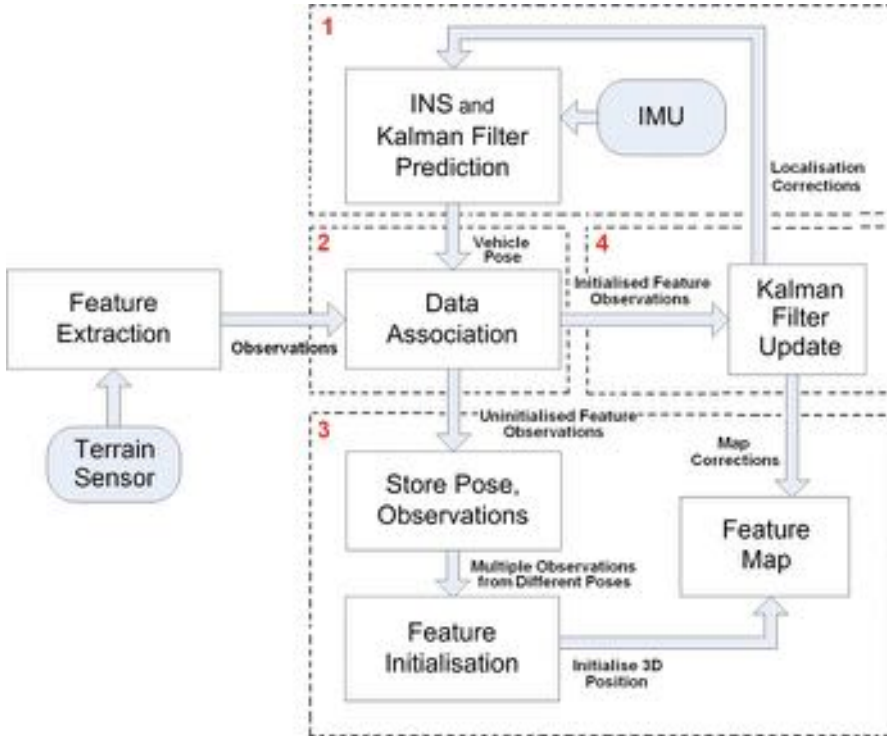


Fig. 21.3 Overview of the bearing-only inertial SLAM algorithm. The algorithm is recursive; EKF state prediction (1) is performed when inertial sensor data is available. Data association (3) and EKF state update (4) are performed when terrain sensor observations are made of features that have been already initialized into the map. Observations of new features are stored until enough observations exist to initialize the position of the feature into the state vector (2)

21.4.1 State Prediction and Update

The bearing-only SLAM equations rely on the storing of vehicle pose data into the estimated state vector. The methods for storing pose data is shown below in Sect. 21.4.2. The state prediction for the bearing-only case follows the same equations and methods as shown in the range/bearing observation case as shown in Sect. 21.3.1 except that stored pose data is predicted forward in the predict step in the same way as stored map features. The only difference in the state update between the bearing-only case and range/bearing case (shown in Sect. 21.3.4) is that now no range information is available in the observation. Equations 21.56–21.65 are used for the update where Eqs. 21.42 or 21.43 is used to compute $\mathbf{H}_i(\hat{\mathbf{x}}^-(k+1))$ or $\mathbf{H}_{i,\text{local}}(\hat{\mathbf{x}}_{\text{local}}^-(k+1))$, the predicted feature observation, depending on the exact form of bearing observation from the terrain sensor (i.e., bearing angles or pixels).

21.4.2 Feature Initialization

A single bearing-only observation is insufficient to initialize the 3D position of a feature into the SLAM filter with Gaussian uncertainty. The following subsection outlines a method for delayed initialization of a feature into the filter by using stored observations and vehicle pose information. For simplification, the feature position is always initialized into the local navigation frame first before being transformed into the ECEF frame if global SLAM is performed. Otherwise, the feature position remains in the local navigation frame.

21.4.2.1 Storing Feature Observations and Vehicle Pose Information

When an observation of an uninitialized feature is made, the current bearing-only observation is stored, and the SLAM state vector is augmented to include the current vehicle pose (three position states and three Euler angle states):

$$\hat{\mathbf{x}}_v = \begin{bmatrix} \mathbf{p}^e(k) \\ \mathbf{v}^e(k) \\ \boldsymbol{\Psi}^n(k) \\ \delta \mathbf{f}^b(k) \\ \delta \omega_{ib}^b(k) \end{bmatrix}, \hat{\mathbf{x}}_p = \begin{bmatrix} \mathbf{C}_e^n \mathbf{p}^e(k) \\ \boldsymbol{\Psi}^n(k) \end{bmatrix} = \begin{bmatrix} \mathbf{p}^n(k) \\ \boldsymbol{\Psi}^n(k) \end{bmatrix} \quad (21.66)$$

$$\hat{\mathbf{x}}_{\text{aug}} = \begin{bmatrix} \hat{\mathbf{x}}_v(k) \\ \mathbf{m}^n(k) \\ \hat{\mathbf{x}}_p(k) \end{bmatrix} \quad (21.67)$$

The state covariance matrix is then augmented with the stored vehicle pose:

$$\mathbf{P}_{\text{aug}}(k) = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} & \mathbf{P}_{vp} \\ \mathbf{P}_{mv} & \mathbf{P}_{mm} & \mathbf{P}_{mp} \\ \mathbf{P}_{pv} & \mathbf{P}_{pm} & \mathbf{P}_{pp} \end{bmatrix} \quad (21.68)$$

$\hat{\mathbf{x}}_v$ is the concatenation of the vehicle position, velocity, attitude, and inertial sensor bias states where $\hat{\mathbf{x}}_p$ is the concatenation of the vehicle position and attitude (i.e., the vehicle pose states) at the time of the observation. $\hat{\mathbf{x}}_{\text{aug}}$ is the augmented state vector which is comprised of the vehicle states ($\hat{\mathbf{x}}_v$), the 3D positions of all of the map features (\mathbf{m}^n), and the added vehicle pose states ($\hat{\mathbf{x}}_p$). The observation (coordinates of the feature in the image plane) is stored separately from the EKF state vector.

In local SLAM, the covariance term \mathbf{P}_{pp} (covariance of the pose states) is derived by taking the position and attitude covariance matrix subblocks from within \mathbf{P}_{vv} (since these states have the same value and the same covariance as the current vehicle position and attitude). Similarly the covariance subblock \mathbf{P}_{pm} is taken from the existing cross correlations between the current vehicle states and map states (i.e., subblocks of \mathbf{P}_{vm} corresponding just to position and attitude). \mathbf{P}_{pv} is the cross correlation between all of the current vehicle states (i.e., position, attitude, velocity,

and inertial sensor biases) and the current pose states. This is thus taken from subblocks of \mathbf{P}_{vv} itself since there are states in common (position and attitude), and thus parts of the added vehicle pose states ($\hat{\mathbf{x}}_p$) are completely correlated to the current vehicle state ($\hat{\mathbf{x}}_v$). In the case of global SLAM, a \mathbf{C}_e^n transformation is applied to the vehicle position covariances in order to represent the stored pose in the local navigation frame.

As the process model of the vehicle comes into play and the filter moves forward in time, the correlations between the stored pose and the new time vehicle states (i.e., $\hat{\mathbf{x}}_v(k+1)$) will decrease (due to the process noise on $\hat{\mathbf{x}}_v(k+1)$).

21.4.2.2 Deciding When to Initialize a Feature

Eventually enough feature observations will be made from varying vehicle poses to initialize the position of the feature. Initializing a feature too early (i.e., by not using enough observations or observations with insufficient separating angle) can result in inconsistency as the true probability distribution of the feature location is not well represented by a Gaussian. The disadvantage with overdelaying the initialization is that the uncertainty in the vehicle states continues to grow before the initialization. As the uncertainty in the current vehicle state grows, linearization errors can affect the consistency of the filter. The information should be initialized and recovered as quickly as possible to limit the effect the inconsistency can have. Additionally, it may be desired to quickly recover the stored information which contributes toward the accuracy of the vehicle localization estimates which may be used as feedback for the control of the vehicle. Another upper limit on deciding how long to delay the initialization is driven by reducing the increased computational burden imposed by adding stored observations to the state vector.

In Bailey (2003), the author discusses a method for testing the conditioning of the initialization by using a Kullback-Leibler distance measure between the linearized update and an approximation of the update using particles to represent the final probability distribution of the feature position. This method however is very computationally intensive; instead, a more practical approach is to set a conservative threshold for the minimum angle between observations necessary to initialize a feature.

21.4.2.3 Initializing a 3D Feature Position Estimate

When it is decided to initialize the 3D position of a feature into the map, the two stored observations of the feature which are separated by the largest angle are used to create an initial estimate of the feature position. Each bearing-only observation can be represented by a 3D point in space \mathbf{y}^n from where the observation was made (at the origin of the sensor) along with a unit vector $\bar{\mathbf{u}}^n$ pointing along the line of sight of the observation; thus,

$$\mathbf{y}^n = \mathbf{p}^n + \mathbf{C}_b^n \mathbf{p}_{sb}^b \quad (21.69)$$

$$\bar{\mathbf{u}}^n = \mathbf{C}_b^n \mathbf{C}_s^b \bar{\mathbf{p}}_{ms}^s \quad (21.70)$$

where $\bar{\mathbf{x}}$ indicates the unit vector of a vector \mathbf{x} . \mathbf{p}^n and C_b^n are determined from the stored pose data associated to each observation, and $\bar{\mathbf{p}}_{ms}^s$ is determined from the observation data itself using Eq. 21.44 to convert the pixel observation to azimuth and elevation angles and Eq. 21.71 to convert to a unit vector:

$$\bar{\mathbf{p}}_{ms}^s = \begin{bmatrix} \cos(\varphi_i) \cos(\vartheta_i) \\ \sin(\varphi_i) \cos(\vartheta_i) \\ \sin(\vartheta_i) \end{bmatrix} \quad (21.71)$$

The lines of sight generated by each observation should intersect at one point in 3D space corresponding to the feature location. Since the observations and stored vehicle pose information is noisy, the lines of sight will generally not intersect perfectly. Instead, the initial feature position is computed as the closest point between the two lines for each observation:

$$\begin{aligned} \mathbf{m}_i^n &= \mathbf{G}(\mathbf{p}_1^n, \mathbf{p}_2^n, \Psi_1^n, \Psi_2^n, \mathbf{z}_1, \mathbf{z}_2) \\ &= \frac{1}{2}(\mathbf{y}_1^n + \mathbf{y}_2^n + p_1 \cdot \bar{\mathbf{u}}_1^n + p_2 \cdot \bar{\mathbf{u}}_2^n) \end{aligned} \quad (21.72)$$

$$p_1 = \frac{((\mathbf{y}_2^n - \mathbf{y}_1^n) \times \bar{\mathbf{u}}_2^n) \cdot (\bar{\mathbf{u}}_1^n \times \bar{\mathbf{u}}_2^n)}{|\bar{\mathbf{u}}_1^n \times \bar{\mathbf{u}}_2^n|^2} \quad (21.73)$$

$$p_2 = \frac{((\mathbf{y}_1^n - \mathbf{y}_2^n) \times \bar{\mathbf{u}}_1^n) \cdot (\bar{\mathbf{u}}_2^n \times \bar{\mathbf{u}}_1^n)}{|\bar{\mathbf{u}}_2^n \times \bar{\mathbf{u}}_1^n|^2} \quad (21.74)$$

In the event that there is a large discrepancy between the two lines (i.e., the minimum distance between the closest two points, one on each line, is larger than a threshold), the observations may be incorrect. This could be due to a misassociation of one of the observations or if the feature is moving for some reason. In this case, the observations are discarded, and the feature is not initialized. Provided there is no large discrepancy between the lines, the state vector and covariance matrix in the SLAM filter are then augmented to include the estimate of the new feature:

$$\hat{\mathbf{x}}_{\text{aug}}(k) = \begin{bmatrix} \hat{\mathbf{x}}(k) \\ \mathbf{m}_i^n(k) \end{bmatrix} \quad (21.75)$$

$$\mathbf{P}_{\text{aug}}(k) = \begin{bmatrix} \mathbf{I} & 0 \\ \nabla \mathbf{G}_p & \nabla \mathbf{G}_z \end{bmatrix} \begin{bmatrix} \mathbf{P}(k) & 0 \\ 0 & \mathbf{R}_{2 \times 2} \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 \\ \nabla \mathbf{G}_p & \nabla \mathbf{G}_z \end{bmatrix}^T \quad (21.76)$$

where $\nabla \mathbf{G}_p$ and $\nabla \mathbf{G}_z$ are the Jacobians of the initialization function $\mathbf{G}(\cdot)$ with respect to the pose states $(\mathbf{p}_1^n, \mathbf{p}_2^n, \Psi_1^n, \Psi_2^n)$ and the observations $(\mathbf{z}_1, \mathbf{z}_2)$, respectively, and $\mathbf{R}_{2 \times 2}$ is

$$\mathbf{R}_{2 \times 2} = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix} \quad (21.77)$$

21.4.2.4 Recovering the Information from Remaining Stored Observations

Once two observations have been used to initialize the 3D position of the feature into the filter, the remaining stored observations ($\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_j$) are run through a batch EKF update. The update corrects not only the current feature being initialized but also the other features in the map and the current vehicle state estimates. The updated state vector and state covariance is calculated using EKF update equations described above for the range-bearing algorithm (i.e., Eqs. 21.56 and 21.57) where the innovation $\mathbf{v}(k)$ is composed using all of the stored observations for the feature:

$$\mathbf{v}(k) = \begin{bmatrix} \mathbf{z}_{1,\text{pix}} - \mathbf{H}(\mathbf{p}_1^n, \Psi_1^n, \mathbf{m}_i^n) \\ \mathbf{z}_{2,\text{pix}} - \mathbf{H}(\mathbf{p}_2^n, \Psi_2^n, \mathbf{m}_i^n) \\ \vdots \\ \mathbf{z}_{j,\text{pix}} - \mathbf{H}(\mathbf{p}_j^n, \Psi_j^n, \mathbf{m}_i^n) \end{bmatrix} \quad (21.78)$$

Once the update has been completed, pose states that no longer have any associated stored observations been removed from the state vector and their corresponding rows and columns removed from the covariance matrix. Finally, in the case of global SLAM, the newly initialized feature position is transformed from the local navigation frame into the ECEF frame. If local SLAM is performed, the feature position remains in the local navigation frame.

21.4.3 Data Association

When performing data association for features that have already been initialized into the map, the same methods are used as in the range/bearing SLAM case shown in Sect. 21.3.3. Issues arise when attempting to find a data association test that can be performed for uninitialized features. Since the exact 3D position of the feature is not known, one cannot consistently calculate the innovation or innovation covariance of the feature. Instead, from only one or a small number of observations with a small baseline, the observation could lie anywhere in 3D space along the line-of-sight of the observation.

In order to associate observations of features that have not yet been initialized into the 3D map, a multi-hypothesis of Gaussian distributions is created for the possible 3D locations of the feature along the line of sight vector for the first observation of the feature.

21.4.3.1 Starting a New Feature

When an observation is made in the image that cannot be associated to any other previously seen feature, initialized or uninitialized, it is assumed that this observation has come from a new feature that has not been seen before. The process begins by storing the observation and augmenting the EKF state vector with the current vehicle pose (see Sect. 21.4.2.1).

From the single observation, a set of equally weighted hypotheses are created for where the feature could lie in 3D space along the line of sight. The mean ($\hat{\mathbf{x}}_j$) and covariance (\mathbf{P}_j) for each hypothesis are calculated for several different values of range (r_j) in equal increments from an expected minimum and maximum sensor range as shown in the left of Fig. 21.4 using Eqs. 21.79 and 21.80:

$$\begin{aligned}\hat{\mathbf{x}}_j &= \mathbf{G}(\mathbf{p}^n(k), \Psi^n(k), \mathbf{z}_i(k), r_j) \\ &= \mathbf{p}^n + C_b^n \mathbf{p}_{sb}^b + r_j \cdot (C_b^n C_s^b \mathbf{p}_{ms}^s)\end{aligned}\quad (21.79)$$

$$\mathbf{P}_j = \nabla \mathbf{G}_v \mathbf{P}_{vv} \nabla \mathbf{G}_v^T + \nabla \mathbf{G}_z \mathbf{R}_{\text{ang}} \nabla \mathbf{G}_z^T \quad (21.80)$$

where \mathbf{p}_{ms}^s is calculated from the observation data using Eq. 21.71 and $\nabla \mathbf{G}_v$, $\nabla \mathbf{G}_z$ are the Jacobians of the function $\mathbf{G}(\cdot)$ with respect to vehicle states and the observation and range data, respectively. The number of hypotheses used and the maximum and minimum range and thus the spacing between the hypotheses depend on the desired accuracy in the initial feature position with more hypotheses resulting in a better initialization. A record of the multi-hypothesis distribution is maintained separately from the state vector and is used only to assist in associating future observations of the feature.

21.4.3.2 Associating Future Observations and Maintaining Feature Hypotheses

Since each hypothesis is Gaussian with a mean defined in 3D space, the innovation and innovation covariance can be calculated for each hypotheses for each uninitialized feature using

$$v(k) = \mathbf{z}_{i,\text{ang}}(k) - \mathbf{H}(\mathbf{p}^n(k), \Psi^n(k), \hat{\mathbf{x}}_j) \quad (21.81)$$

$$\mathbf{S}(k) = \nabla \mathbf{H}_x(k) \mathbf{P}_{\text{hyp}}(k) \nabla \mathbf{H}_x^T(k) + \mathbf{R}_{\text{ang}} \quad (21.82)$$

$$\mathbf{P}_{\text{hyp}}(k) = \begin{bmatrix} \mathbf{P}_{pp} & 0 \\ 0 & \mathbf{P}_j \end{bmatrix} \quad (21.83)$$

where \mathbf{P}_{pp} is the covariance subblock of the current vehicle position and attitude states. An approximation is made that the current vehicle state and the hypothesis are uncorrelated in order to simplify the data association process; to account for these correlations in the data association process would require a large computational resource. The result of this approximation is that during large SLAM loop closures, the EKF may not associate observations of a new feature and some observations may be discarded. It may thus take longer for new features at this time to be integrated into the map. For a given observation and for each hypothesis for a given uninitialized feature, γ is calculated using Eqs. 21.55, 21.81, and 21.82. If the value of γ is below the threshold corresponding to a 95 % confidence for at least one of the hypotheses, then the observation is matched to this uninitialized feature.

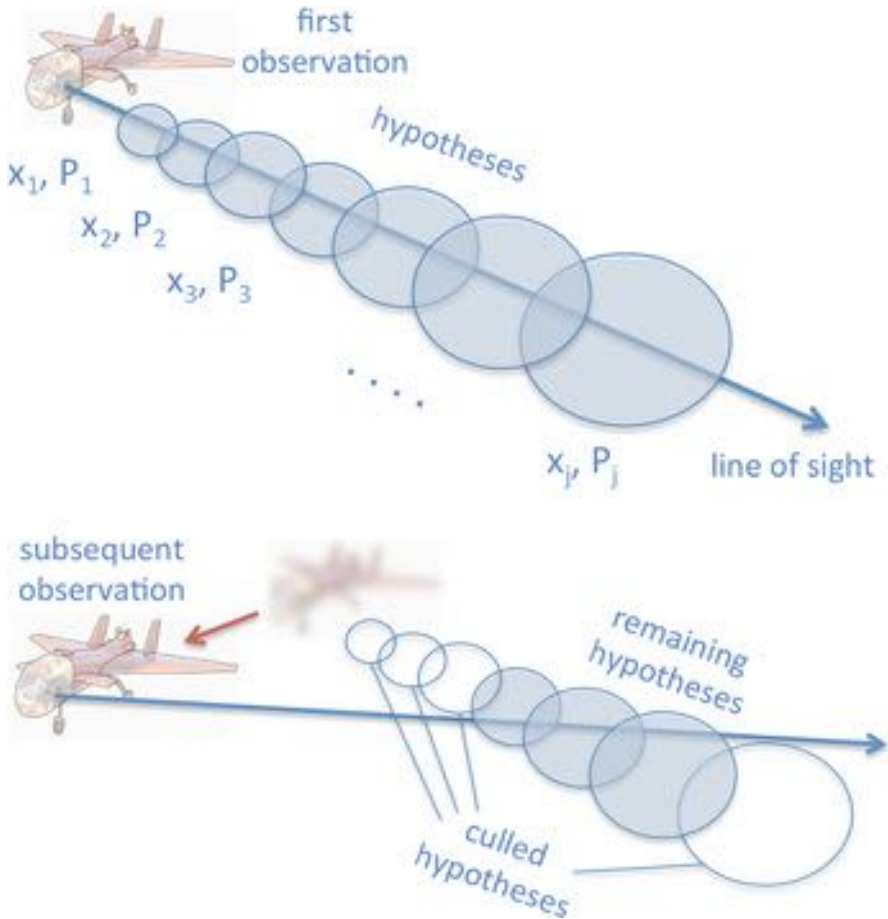


Fig. 21.4 Data association of observations to uninitialized features. When a feature is seen for the first time, a set of hypotheses for the 3D position of the feature is generated at equal range increments along the line of sight (*top*). Future observations are checked for matches to any of the hypotheses. When a match is made to one of the hypotheses, the remaining hypotheses that do not match are culled (*bottom*)

In order to simplify the association of uninitialized features, when an association is made between an observation and one of the hypotheses for a given feature, all other hypotheses of this feature for which the observation does not match are culled from the set of hypotheses from which to associate future observations. As the vehicle moves around an uninitialized feature, the number of hypotheses gradually drops until only one hypothesis matches, the one that is closest to the true 3D feature location. The bottom sub-figure of Fig. 21.4 illustrates this process.

21.4.3.3 Data Association Procedure

Each time observations from the feature extraction process are received, the procedure begins by using Eq. 21.55 to evaluate the potential matching between

each observation and each of the 3D initialized features. Observations that match 3D initialized features are associated and sent on to the SLAM filter to be updated. In the event of multiple features matching a single observation, the matching with the lowest value of γ will be accepted.

The remaining observations are tested for matches with each of the hypotheses for each uninitialized feature. Observations that match with at least one hypothesis of an uninitialized feature are associated to this feature. The observation itself is stored, and the vehicle pose at the current time is then added to the state vector (see Sect. 21.4.2.1). In the event of multiple uninitialized features matching a single observation, all matchings to this observation are rejected.

For each remaining observation not matched to an initialized or un-initialized feature, a new set of hypotheses is created (see Sect. 21.4.3.1).

The proposed multi-hypothesis method for data association could also potentially be used for initializing the feature position, as has been demonstrated in Kwok and Dissanayake (2004), where hypotheses are pruned until only one is left, which then becomes the initialized feature. In this approach, the line-of-sight intersection method is used for calculating the initial feature position; in order to achieve the same accuracy with the multi-hypothesis method, one would require a prohibitively large number of hypotheses (i.e., 1-m resolution for a feature at a range of 200 m would require 200 hypotheses).

21.5 Multi-vehicle Inertial SLAM Algorithm

In multi-vehicle SLAM, several vehicles move over a common section of terrain where the task is to build a common map of the environment while providing localization estimates to each platform. The use of multiple cooperating vehicles has many advantages over SLAM on a single vehicle. Multiple vehicles provide wider sensor coverage and thus can be used to build more extensive terrain maps in less time. The accuracy of the constructed terrain map is greater than in the single-vehicle case as multiple vehicles contribute information toward a given feature location. This increased terrain map accuracy also creates an increase in the accuracy of the localization estimates for each vehicle.

The work by Fenwick et al. (2002), Mourikis and Roumeliotis (2005), Walter and Leonard (2004), and Thrun and Lui (2003a) provides examples of multi-vehicle SLAM where all vehicles send their sensor data to a central Kalman filter. These approaches are fully centralized, involving the communication of raw sensor data from each vehicle to a central source and thus requiring a large amount of communication bandwidth. In Nettleton et al. (2003) and Ong et al. (2003), the authors demonstrate a decentralized architecture for multi-vehicle SLAM using a mathematically equivalent form of the EKF known as the Extended Information Filter (EIF). The use of the EIF in these approaches avoids the need to communicate raw sensor data, where instead this raw data is used in SLAM by each vehicle locally before communicating the map information.

In this section, data fusion architectures are considered for sharing map information built by each vehicle using the inertial SLAM algorithms discussed in Sects. 21.3 and 21.4. The core principle behind the data fusion schemes is the use of the EIF which allows for the map data contributions from each platform to be summed together where the processes of the combined estimation task are distributed among the vehicles. Both centralized and decentralized architectures are discussed in Sects. 21.5.2 and 21.5.3.

21.5.1 Global Vs. Local SLAM for Multiple Vehicles

When global SLAM is performed by each of the vehicles in the data fusion network, each vehicle uses the ECEF frame for referencing the position of terrain features in the environment. In the case of local SLAM, however, each vehicle may use its own independent local navigation frame in which it builds its local map. In order to fuse information from multiple maps, the relative transformations between each local navigation frame must be known. In the case where no localization or prior terrain reference information is available, the vehicles can compute a relative transformation by matching at least two features from each local map (for an example of this process, the reader is referred to Fox et al. (2006)). Once the transformation is known, a common representation of the environment can be built with respect to each vehicle's local navigation frame.

21.5.2 Centralized Architectures for Multi-vehicle Inertial SLAM

This section presents a centralized, distributed data fusion architecture for multi-vehicle inertial SLAM. The architecture is centralized, where some part of the data fusion process is performed at a central node. The architecture is also distributed; rather than communicate all of the raw sensor data from both the inertial and terrain sensors to a central data fusion source, each vehicle firstly performs single-vehicle inertial SLAM as shown in Sects. 21.3 and 21.4. The local terrain map built on each vehicle is then communicated in information form to a central source at regular intervals where data fusion is performed. Finally, the central data fusion node communicates the fused map information back to each vehicle, where this information is fused back into the local map. The following subsections describe the process in more detail.

21.5.2.1 Centralized, Distributed Data Fusion

The centralized, distributed data fusion is based on the independent opinion pool architecture shown in Manyika and Durrant-Whyte (1994). At regular intervals, each vehicle takes its current state estimate relating to the map estimates only, that is, $\mathbf{x}_m(k)$ and $\mathbf{P}_{mm}(k)$, where

$$\mathbf{x}_m(k) = \begin{bmatrix} \mathbf{m}_1^e(k) \\ \mathbf{m}_2^e(k) \\ \vdots \\ \mathbf{m}_N^e(k) \end{bmatrix} \quad (21.84)$$

for the case of global SLAM and

$$\mathbf{x}_m(k) = \begin{bmatrix} \mathbf{m}_1^n(k) \\ \mathbf{m}_2^n(k) \\ \vdots \\ \mathbf{m}_N^n(k) \end{bmatrix} \quad (21.85)$$

for the case of local SLAM. $\mathbf{P}_{mm}(k)$ is a $3N \times 3N$ matrix of the elements of \mathbf{P}_k relating to the map feature estimates. Each vehicle then calculates its posterior information:

$$\mathbf{Y}_j(k) = \mathbf{P}_{mm}^{-1}(k) \quad (21.86)$$

$$\mathbf{y}_j(k) = \mathbf{Y}_j(k)\mathbf{x}_m(k) \quad (21.87)$$

for the j th vehicle where $j = 1, \dots, M$, where M is the number of vehicles, and communicates this to the central map filter. The information that is sent will obviously be correlated to the information that was sent in the previous communication (since each vehicle's posterior information is based on the entire history of observations it has made). To overcome this, the central data filter maintains a record of the information that it has been sent in the previous communication ($\mathbf{Y}_j(k-1), \mathbf{y}_j(k-1)$) by each vehicle. When the new information arrives, the old information is subtracted from it before adding it to the central map information, in order to remove the correlations and only count new information. The central map information update at the central data filter is thus

$$\mathbf{Y}_{\text{central}}(k) = \mathbf{Y}_{\text{central}}(k-1) + \sum_{j=1}^M (\mathbf{Y}_j(k) - \mathbf{Y}_j(k-1)) \quad (21.88)$$

$$\mathbf{y}_{\text{central}}(k) = \mathbf{y}_{\text{central}}(k-1) + \sum_{j=1}^M (\mathbf{y}_j(k) - \mathbf{y}_j(k-1)) \quad (21.89)$$

Once the information is combined in the central filter, a state-space estimate of the map feature locations and covariance can be recovered using Eqs. 21.98 and 21.99:

$$\mathbf{P}_{mm,\text{central}}(k) = \mathbf{Y}_{\text{central}}^{-1}(k) \quad (21.90)$$

$$\mathbf{x}_{m,\text{central}}(k) = \mathbf{P}_{mm,\text{central}}(k)\mathbf{y}_{\text{central}}(k) \quad (21.91)$$

21.5.2.2 Applying Local Node Feedback to the Independent Opinion Pool

So that each vehicle's localization estimates can benefit from the observations of features made by other vehicles, information about the central map should be fed back to each of the local nodes. In the same way that was done on the central data filter, each vehicle must store the last information update that it received from the central filter ($\mathbf{Y}_{\text{central}}(k-1), \mathbf{y}_{\text{central}}(k-1)$) so as not to double count the information that has been sent to it. Thus, when each vehicle receives the communicated central information, it firstly computes its posterior information over the entire state space consisting of local vehicle estimate and map features using Eqs. 21.86 and 21.87 and updates this information using Eqs. 21.92 and 21.93:

$$\mathbf{Y}_{\text{local}}(k) = \mathbf{Y}_{\text{local}}(k) + (\mathbf{Y}_{\text{central}}(k) - \mathbf{Y}_{\text{central}}(k-1)) \quad (21.92)$$

$$\mathbf{y}_{\text{local}}(k) = \mathbf{y}_{\text{local}}(k) + (\mathbf{y}_{\text{central}}(k) - \mathbf{y}_{\text{central}}(k-1)) \quad (21.93)$$

The local information is then transformed back into state-space and covariance form to provide the updated estimate of the vehicle localization and map features, which is substituted back into the EKF in the single-vehicle SLAM architecture. The operation of the central filter with local node feedback is illustrated in Fig. 21.5.

This centralized, distributed architecture has several advantages over a completely centralized filter such as a reduction in the required communication bandwidth (as only local estimates must be communicated, not observations and process model inputs) and the ability to deal with intermittent communications and delays as the information is maintained on the local vehicle.

21.5.3 Decentralized Architectures for Multi-vehicle Inertial SLAM

The multi-vehicle inertial SLAM algorithm can be decentralized by removing the central filter, where each vehicle now communicates directly to each other vehicle in the network. This type of architecture was demonstrated for feature tracking tasks and SLAM in Nettleton (2003). At regular intervals each UAV takes its current state estimate relating to the map estimates only, that is, $\mathbf{x}_m(k)$ and $\mathbf{P}_{mm}(k)$, and calculates its posterior information using Eqs. 21.86 and 21.87. Each UAV maintains a record of the information sent during the last communication (i.e., $\mathbf{Y}_j(k-1), \mathbf{y}_j(k-1)$) which is subtracted from the current information to form the new information that UAV has about the feature map:

$$\mathbf{Y}_{j,\text{new}}(k) = \mathbf{Y}_j(k) - \mathbf{Y}_j(k-1) \quad (21.94)$$

$$\mathbf{y}_{j,\text{new}}(k) = \mathbf{y}_j(k) - \mathbf{y}_j(k-1) \quad (21.95)$$

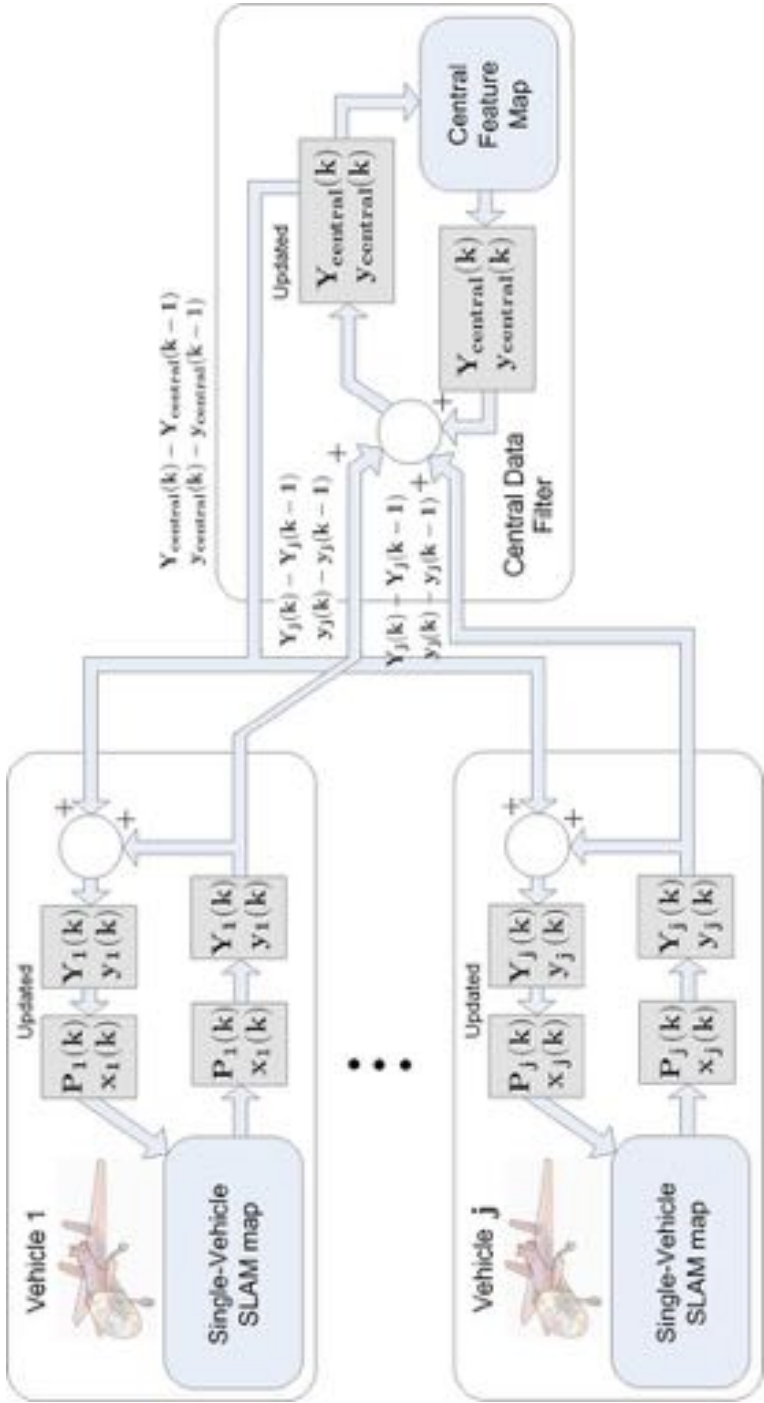


Fig. 21.5 Distributed centralized multi-vehicle SLAM: independent opinion pool architecture with local node feedback. Each vehicle communicates its posterior map estimates in information form which are added together at a central data filter. The central data filter then feeds back the information to each of the vehicles in order to update their local maps

This new information is then communicated to each of the other UAVs. When each UAV receives all of the information updates from each of the other UAVs, this information is summed together along with the current UAV information to form the updated estimate of the map features in information form:

$$\mathbf{Y}_{j,\text{update}}(k) = \mathbf{Y}_j(k) + \sum_{i=1}^M \mathbf{Y}_{i,\text{new}}(k) \quad (21.96)$$

$$\mathbf{y}_{j,\text{update}}(k) = \mathbf{y}_j(k) + \sum_{i=1}^M \mathbf{y}_{i,\text{new}}(k) \quad (21.97)$$

Once all of the information from other UAVs is combined in the update, a state-space estimate of the map feature locations and covariance can be recovered back into the EKF using Eqs. 21.98 and 21.99:

$$\mathbf{P}_{j,mm,\text{update}}(k) = \mathbf{Y}_{j,\text{update}}^{-1}(k) \quad (21.98)$$

$$\mathbf{x}_{j,m,\text{update}}(k) = \mathbf{P}_{j,mm,\text{update}}(k) \mathbf{y}_{j,\text{update}}(k) \quad (21.99)$$

The operation of the decentralized SLAM filter is illustrated in Fig. 21.6.

21.5.4 Delayed Observations, Network Outages, and Communication Bandwidth Constraints

Realistic communication networks between the vehicles will not be able to provide continuous and instantaneous communication of information. Instead, real networks will contain significant delays and outages between different nodes when vehicles move out of range of one another and will not always be able to communicate all of the map information when the map becomes very large.

Delayed observations are not a significant issue in multi-vehicle SLAM as features are considered stationary, and thus information about a feature's location is independent of time and can be added in a delayed fashion and out of order. When there are outages in the network communications, this can cause the vehicles to lose track of the common information they possess. This can be overcome by constraining the structure of the communications network to tree structures (Nettleton 2003). When communication bandwidth constraints apply across the network, the vehicles may only communicate information representing a subset of the features contained in their map. In this case, information must be fused together using the covariance intersect algorithm (Julier and Uhlmann 2001) due to correlations between the submap and the rest of the map features, which is not accounted for in further communications. These concepts are all discussed further in Nettleton (2003).

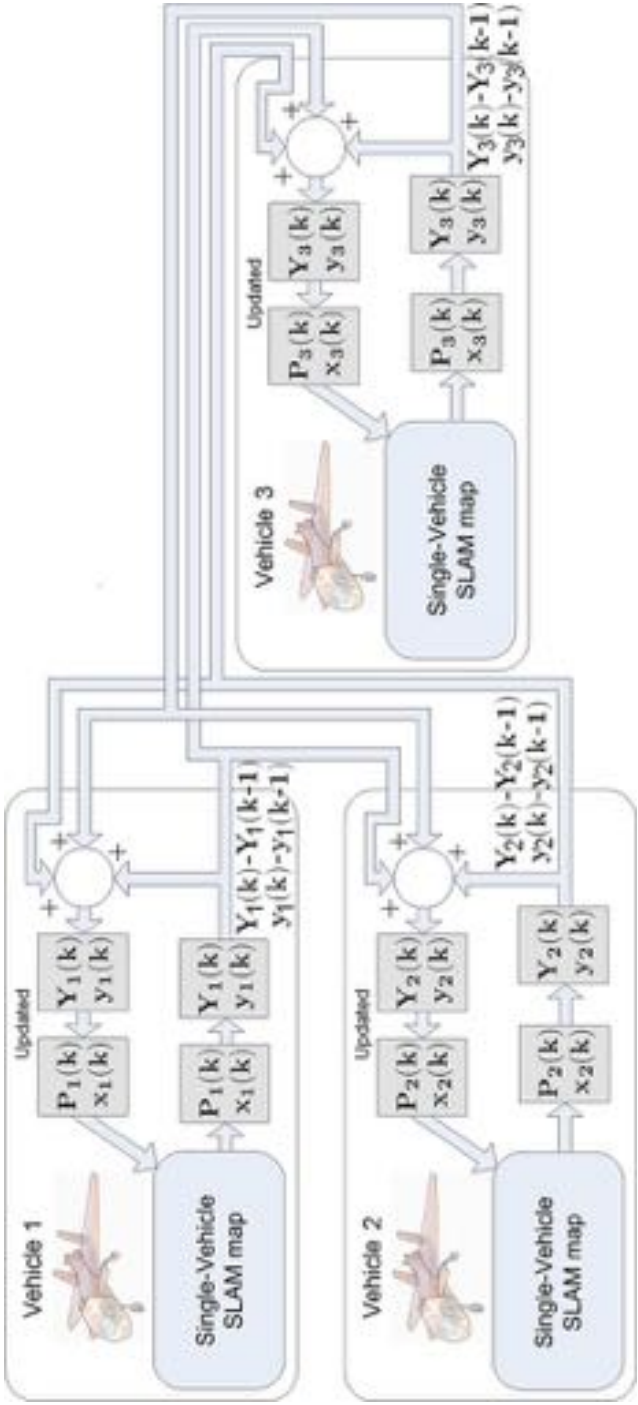


Fig. 21.6 Decentralized SLAM data fusion architecture. Each UAV communicates to each other UAV in the team its latest map state estimate in inverse covariance (information) form. Information is then added at the receiving end and converted from information space back into state space into the EKF

21.6 Conclusion

This chapter has developed the basic equations and methods for inertial sensor-based SLAM. The fundamental equations that model inertial sensors and inertial localization were analyzed; two different applications were examined for when localization and mapping is performed in either a global frame of reference or in an arbitrary local frame of reference. The inertial SLAM algorithms were examined for the case of range and bearing observations from a terrain sensor and also when terrain observations were made from a bearing-only sensor. Finally, the problem of multi-vehicle inertial SLAM was examined. Two different types of data fusion architecture were considered: firstly, centralized architectures in which map information is shared among vehicles via a central communications source and, secondly, decentralized architectures where the vehicles communicate and share data with each other directly.

References

- T. Bailey, Constrained initialisation for bearing-only SLAM, in *IEEE International Conference on Robotics and Automation*, Institute of Electrical Engineers, Piscataway, USA (2003)
- I. Bar-Itzhack, D. Serfaty, Y. Vitek, Doppler-aided low-accuracy strapdown inertial navigation system. *J. Guid. Control* **5**(3), 236–242 (1982)
- M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, S. Teller, An Atlas framework for scalable mapping, in *IEEE International Conference on Robotics and Automation*, Institute of Electrical Engineers, Piscataway, USA (2003)
- M. Bryson, S. Sukkarieh, Building a Robust implementation of bearing-only inertial SLAM for a UAV. *J. Field Robot. Spl. Issue SLAM Field* **24**(2), 113–143 (2007)
- M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **17**(3), 229–241 (2001)
- J. Fenwick, P. Newman, J. Leonard, Cooperative concurrent mapping and localisation, in *IEEE International Conference on Robotics and Automation*, Institute of Electrical Engineers, Piscataway, USA (2002)
- D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Shulz, B. Stewart, Distributed multirobot exploration and mapping. *Proc. IEEE* **94**(7), 1325–1339 (2006)
- C.S. Giovanni, Performance of a ring laser strapdown attitude and heading reference for aircraft. *J. Guid. Control* **2**(4), 320–327 (1979)
- J. Guivant, E. Nebot, Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. Robot. Autom.* **17**(3), 242–257 (2001)
- J. Gutmann, K. Konolige, Incremental mapping of large cyclic environments, in *International Symposium on Computational Intelligence in Robotics and Automation*, Institute of Electrical Engineers, Piscataway, USA (1999)
- S. Julier, J. Uhlmann, *General Decentralised Data Fusion with Covariance Intersection (CI): Handbook of Data Fusion* (CRC, Boca Raton, 2001)
- J. Kim, S. Sukkarieh, Airborne simultaneous localization and map building, in *IEEE International Conference on Robotics and Automation*, Institute of Electrical Engineers, Piscataway, USA (2003)
- N. Kwok, G. Dissanayake, An efficient multiple hypothesis filter for bearing only SLAM, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Institute of Electrical Engineers, Piscataway, USA (2004)

- D. Lowe, Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
- J. Manyika, H. Durrant-Whyte, *Data Fusion and Sensor Management: a decentralised information-theoretic approach*, (Ellis Horwood Ltd., London, 1994)
- J. Meyer-Hilberg, T. Jacob, High accuracy navigation and landing system using GPS/IMU system integration, in *IEEE Position, Location and Navigation Symposium*, Institute of Electrical Engineers, Piscataway, USA (1994)
- A. Mourikis, S. Roumeliotis, Performance bounds for cooperative simultaneous localisation and mapping (C-SLAM), in *Robotics: Science and Systems Conference*, <http://www.roboticsproceedings.org/rss01/p10.html> (2005)
- J. Neira, J. Tardos, Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robot. Autom.* **17**(6), 890–897 (2001)
- E. Nettleton, Decentralised Architectures for Tracking and Navigation with Multiple Flight Vehicles. Ph.d, University of Sydney, 2003
- E. Nettleton, S. Thrun, H. Durrant-Whyte, S. Sukkarieh, Decentralised SLAM with low-bandwidth communication for teams of vehicles, in *4th International Conference on Field and Service Robotics*, Springer-Verlag, New York (2003)
- M. Nixon, A. Aguado, *Feature Extraction and Image Processing* (Butterworth Heinmann/Newnes, Oxford, 2001)
- E. Olson, J. Leonard, S. Teller, Robust range-only Beacon localization, in *IEEE Autonomous Underwater Vehicle Conference*, Institute of Electrical Engineers, Piscataway, USA (2004)
- S. Ong, M. Ridley, J. Kim, E. Nettleton, S. Sukkarieh, Six DoF decentralised SLAM, in *Australasian Conference on Robotics and Automation*, <http://www.araa.asn.au/acra/acra2003/papers/31.pdf> (2003)
- R. Smith, M. Self, P. Cheeseman, *Estimating Uncertain Spatial Relationships in Robotics: Autonomous Robot Vehicles* (Springer, New York, 1990)
- S. Thrun, Y. Lui, Multi-Robot SLAM with sparse extended information filters, in *International Symposium on Robotics Research*, Springer-Verlag, New York (2003a)
- S. Thrun, Y. Lui, Results for outdoor-SLAM using sparse extended information filters, in *IEEE International Conference on Robotics and Automation*, Institute of Electrical Engineers, Piscataway, USA (2003b)
- D. Titterton, J. Weston, *Strapdown Inertial Navigation Technology* (Peter Peregrinus Ltd., London, 1997)
- M. Walter, J. Leonard, An experimental investigation of cooperative SLAM, in *5th International Symposium on Intelligent Autonomous Vehicles* (2004)
- S. Williams, G. Dissanayake, H. Durrant-Whyte, Towards terrain-aided navigation for underwater Robotics. *Adv. Robot.* **15**(5), 533–550 (2001)

Mitch Bryson and Salah Sukkarieh

Contents

| | | |
|--------|--|-----|
| 22.1 | Introduction | 434 |
| 22.1.1 | Aided Inertial Localization and Related Work | 435 |
| 22.2 | Frames of Reference and Attitude Representation | 436 |
| 22.2.1 | Earth-Centered, Earth-Fixed (ECEF) Frame | 436 |
| 22.2.2 | Local-Vertical, Local-Horizontal Navigation Frame | 437 |
| 22.2.3 | Body-Fixed Frame | 437 |
| 22.2.4 | IMU-Fixed Frame | 437 |
| 22.2.5 | Direction Cosine Matrix | 437 |
| 22.2.6 | Euler Angles | 438 |
| 22.2.7 | Quaternions | 439 |
| 22.3 | Inertial Localization Equations | 439 |
| 22.3.1 | Earth-Frame Mechanization: Position and Velocity Equations | 439 |
| 22.3.2 | Earth-Frame Mechanization: Attitude Equations | 441 |
| 22.3.3 | Simplification of Inertial Navigation Equations for Small Area Operation | 442 |
| 22.3.4 | Numerical Integration of Inertial Equations in the ILS | 443 |
| 22.4 | Aided Inertial Localization Using the EKF | 444 |
| 22.4.1 | Extended Kalman Filter Basics | 444 |
| 22.4.2 | Indirect Kalman Filter Architecture for Aided ILS | 444 |
| 22.4.3 | Kalman Filter Error Process Model | 446 |
| 22.4.4 | Kalman Filter Update and Observation Models | 448 |
| 22.4.5 | Kalman Filter Prediction Stage | 451 |
| 22.4.6 | Kalman Filter Update Stage | 452 |
| 22.5 | Example: Inertial-GPS Localization on a UAV | 453 |
| 22.5.1 | Experimental Setup | 454 |
| 22.5.2 | Results and Discussion | 454 |
| | References | 460 |

M. Bryson (✉)

Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW, Australia

e-mail: m.bryson@acfr.usyd.edu.au

S. Sukkarieh

Australian Centre for Field Robotics (ACFR), School of Aerospace, Mechanical & Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW, Australia

e-mail: salah@acfr.usyd.edu.au

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

433

DOI 10.1007/978-90-481-9707-1_3,

© Springer Science+Business Media Dordrecht 2015

Abstract

This chapter provides an overview of UAV localization with a focus on aided inertial localization, that is, algorithms for fusing data from, for example, satellite positioning systems, barometric sensors, and magnetometers with inertial sensors to provide real-time position and orientation. The presentation is based in the use of the Extended Kalman Filter (EKF) as the core tool for data fusion, which is commonly used for its simplicity, computational efficiency, and optimality (with respect to sensor noise characteristics). This chapter presents an example implementation of aided inertial localization on a UAV as a tutorial in the key concepts in airborne localization and as a basic guide toward more complicated implementation.

22.1 Introduction

Localization is the process of determining the one's position and orientation in the world, often achieved by combining various measurements together. In the context of Unmanned Aerial Vehicles (UAVs), localization is typically performed in real time onboard the platform, using information from various sensors. Tasks that require localization range from automatic stabilization and flight control of the UAV to georegistration of exteroceptive sensor data such as that performed during target tracking and mapping. UAV localization thus often requires a localization system to provide position and orientation information that has low latency, high frequency, and high accuracy, requirements that typically necessitate multiple sensors.

Inertial sensors and satellite positioning systems (such as the Global Positioning System (GPS)) have been a popular choice for UAV systems (and airborne platforms at large) that operate outdoors. Inertial sensors provide high-frequency information (typically 100–1,000 Hz) relating to the first- and second-order time derivatives of a platform's position and orientation, but the accuracy of localization provided by this sensor alone decreases during flight, due to dead reckoning. Modern satellite positioning systems provide accurate information about a platform's position but at a much lower frequency (typically 1–10 Hz) and higher latency (typically 20–200 ms); the fusion of information from both these sensors allows for both accuracy and high-rate information about location. Other localization information typically available on airborne platforms (such as barometric altitude, barometric airspeed, and magnetic field measurements) can also be used to improve the accuracy and robustness of a UAV localization system.

This chapter provides an overview of UAV localization with a focus on aided inertial localization, that is, algorithms for fusing data from, for example, satellite positioning systems, barometric sensors, and magnetometers with inertial sensors to provide real-time position and orientation. The presentation is based in the

use of the Extended Kalman Filter (EKF) as the core tool for data fusion, which is commonly used for its simplicity, computational efficiency, and optimality (with respect to sensor noise characteristics). This chapter presents an example implementation of aided inertial localization on a UAV as a tutorial in the key concepts in airborne localization and as a basic guide toward more complicated implementation, which are beyond the scope of this chapter. The presentation begins with an overview of frames of reference and representations of three-axis orientation in Sect. 22.2. Equations for inertial localization are derived in Sect. 22.3. A set of algorithms for aided inertial localization based on the EKF are presented in Sect. 22.4 including a description of various aiding sensors typically used in UAV localization. Example results of aided inertial localization taken from an experimental flight test of a UAV are provided in Sect. 22.5 and are used to demonstrate concepts developed throughout the chapter. Conclusions are presented in Sect. 22.3.6.

22.1.1 Aided Inertial Localization and Related Work

Since its initial development in the 1940s in guided rocket applications (King and Kutta 1998), inertial localization has been a popular system for localization of aerial vehicles. Originally, inertial localization systems were designed to run independently of external sensors, thus the main research interest being in increased sensor accuracy. The disadvantage of these systems was the large size, power requirements, and high cost of these units, particularly detrimental in airborne applications.

The development of the Kalman filter (Kalman 1960) in the early 1960s provided a method to optimally combine inertial sensor measurements with other localization sensors in order to reduce the drift when using low-cost sensors. Good examples of architectures for an aided inertial localization system with doppler radar, magnetic heading, and airspeed sensor measurements using the Kalman filter are provided in (Giovanni 1979) and (Bar-Itzhack et al. 1982).

Interest in guided missile applications in which external navigation signals (like those used in radio ranging, etc.) could be unreliable or jammed by interference, motivated the development of terrain-aided navigation systems. Most commonly, these systems use radar-altimeter measurements to correlate the observed terrain profile with a stored contour map of the operating area (systems such as TERCOM (TERrain CONtour Matching) (Baker and Clem 1979) and TERPROM (TERrain PROfile Matching) (Fountain 1997)).

Since the abandonment of the policy of selective availability by the United States of America and the development of Differential GPS (DGPS), GPS-aided inertial applications have become a popular choice for UAV and ground vehicle localization systems alike due to their relatively low cost and independence of platform specifics. The Kalman filter is commonly used (Meyer-Hilberg and Jacob 1994; Knight

1996); however, GPS-aided inertial applications using other filtering techniques like complimentary filters (Gebre-Egziabher et al. 1998; Yang et al. 2000), particle filters (Carvalho et al. 1997), and even fuzzy inference schemes (Hiliuta et al. 2004) have also been demonstrated.

22.2 Frames of Reference and Attitude Representation

This section describes the relevant frames of reference and attitude representations used in UAV localization. Figure 22.1 illustrates each of the frames of reference described below.

22.2.1 Earth-Centered, Earth-Fixed (ECEF) Frame

The Earth-Centered, Earth-Fixed (ECEF) frame is fixed to the center of the Earth with x-axis pointing at the intersection of the prime meridian and the equator

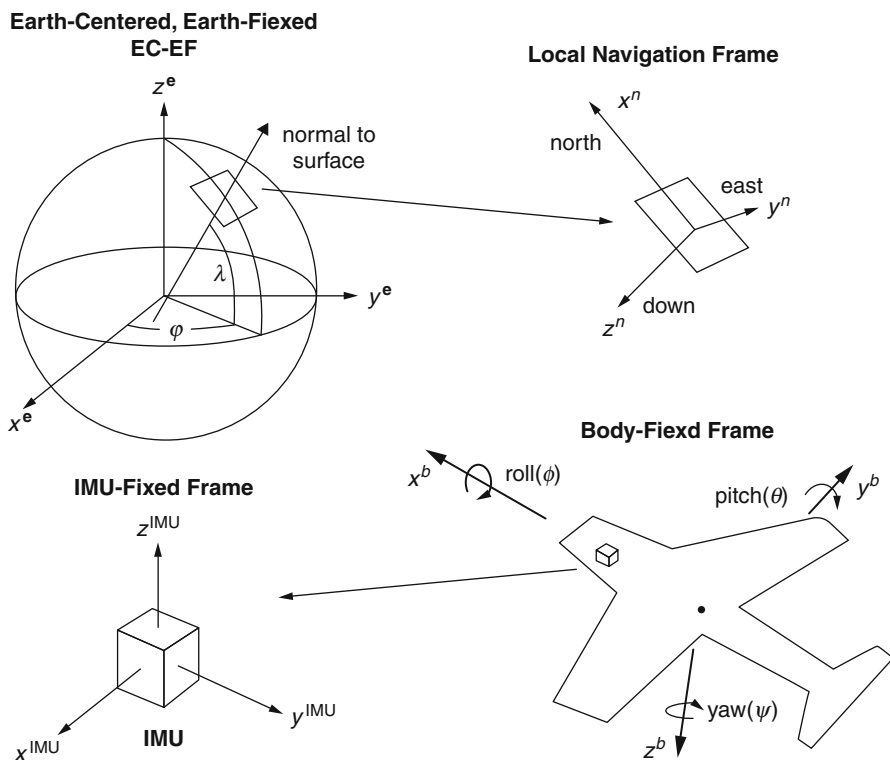


Fig. 22.1 Frames of reference in localization: Earth-Centered, Earth-Fixed (ECEF), local navigation, IMU-fixed, and body-fixed frames

(0° longitude, 0° latitude), z-axis pointing through the north pole and y-axis pointing along the equator, perpendicular to the other two axes. Vectors are referenced in this frame with the superscript “e,” that is, \mathbf{x}^e .

22.2.2 Local-Vertical, Local-Horizontal Navigation Frame

The Local-Vertical, Local-Horizontal (LVLH) frame has its origin at an arbitrary point on the vehicle’s body but whose axes rotate as the vehicle moves such that the x-axis points north, y-axis points east, and z-axis points downward, perpendicular to the tangent of the Earth’s surface at the frame’s origin. Vectors are referenced in this frame with the superscript “n,” that is, \mathbf{x}^n .

22.2.3 Body-Fixed Frame

The body-fixed frame is centered at an arbitrary, but fixed point on the vehicle with x-axis pointing out the aircraft nose, y-axis pointing out the right wing, and z-axis pointing downward through the bottom of the vehicle. Vectors are referenced in this frame with the superscript “b,” that is, \mathbf{x}^b .

22.2.4 IMU-Fixed Frame

The IMU-fixed frame is centered at the location of the Inertial Measuring Unit (IMU). It is assumed that the sensor placement (three accelerometers and three gyros) inside the IMU is in an orthogonal arrangement. Therefore, each axis of the sensor-fixed frame is aligned with each pair of accelerometer and gyro on each axis (i.e., x-axis is aligned with the direction of the x-accelerometer and x-gyro). Vectors are referenced in this frame with the superscript “IMU,” that is, \mathbf{x}^{IMU} .

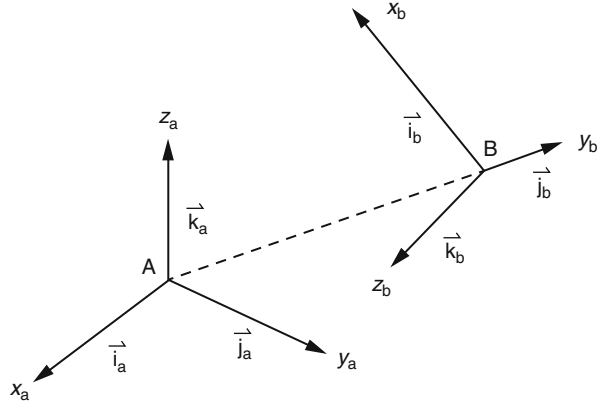
22.2.5 Direction Cosine Matrix

The Direction Cosine Matrix (DCM) is a 3×3 matrix, denoted $\mathbf{C}_{(\cdot)}^{(\cdot)}$, used to transform a vector from one reference frame to another. For example, \mathbf{C}_b^n is the DCM used to transform a vector \mathbf{x}^b , referenced in the body frame, into local navigation coordinates as shown in Eq. 22.1.

$$\mathbf{x}^n = \mathbf{C}_b^n \mathbf{x}^b \quad (22.1)$$

Consider two reference frames A and B as shown in Fig. 22.2. The unit vectors $\mathbf{i}^a, \mathbf{j}^a$, and \mathbf{k}^a and $\mathbf{i}^b, \mathbf{j}^b$, and \mathbf{k}^b are aligned with the x-, y-, and z-axes of each reference frame. The direction cosine matrix that transforms a vector in frame A into frame B (\mathbf{C}_a^b) is given by the dot products between each unit vector as shown

Fig. 22.2 Direction cosine matrix (DCM): relationship between two frames of reference A and B



in Eq. 22.2, where the unit vectors \mathbf{i}^b , \mathbf{j}^b , and \mathbf{k}^b are along the axes of the frame A. These dot products are equal to the cosines of the relative angles between each unit vector.

$$\mathbf{C}_a^b = \begin{bmatrix} \mathbf{i}^a \cdot \mathbf{i}^b & \mathbf{j}^a \cdot \mathbf{i}^b & \mathbf{k}^a \cdot \mathbf{i}^b \\ \mathbf{i}^a \cdot \mathbf{j}^b & \mathbf{j}^a \cdot \mathbf{j}^b & \mathbf{k}^a \cdot \mathbf{j}^b \\ \mathbf{i}^a \cdot \mathbf{k}^b & \mathbf{j}^a \cdot \mathbf{k}^b & \mathbf{k}^a \cdot \mathbf{k}^b \end{bmatrix} \quad (22.2)$$

22.2.6 Euler Angles

Euler angles ($\Psi = [\phi, \theta, \psi]^T$) can be used to parameterize a DCM via the use of three successive rotations about three different axes, the order in which the rotation is applied being significant. In this presentation, a positive yaw(ψ)-pitch(θ)-roll(ϕ) (z-axis, y-axis, x-axis) ordering is used to represent the transformation from either ECEF or LVLH to body-fixed reference. When applying a transformation from the body-fixed reference to either the ECEF or LVLH frame, the sequence of rotations is applied in the opposite direction, that is, a negative roll(ϕ)-pitch(θ)-yaw(ψ) ordering. The direction cosine matrix \mathbf{C}_b^n (transformation from body to local navigation axes) is related to the Euler angles via Eq. 22.3:

$$\mathbf{C}_b^n = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (22.3)$$

22.2.7 Quaternions

Quaternions use four parameters ($\mathbf{q} = [e_0, e_1, e_2, e_3]^T$) to define a transformation between two frames of reference by using a single rotation about a defined axis with respect to the original reference frame. Quaternions are related to the Euler angles and DCM using Eqs. 22.4 and 22.5.

$$\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} \\ \cos \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} - \sin \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} \\ \cos \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} \\ -\cos \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} + \sin \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} \end{bmatrix} \quad (22.4)$$

$$\mathbf{C}_b^n = \begin{bmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_1e_2 - e_0e_3) & 2(e_1e_3 + e_0e_2) \\ 2(e_1e_2 + e_0e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2e_3 - e_0e_1) \\ 2(e_1e_3 - e_0e_2) & 2(e_2e_3 + e_0e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{bmatrix} \quad (22.5)$$

22.3 Inertial Localization Equations

This section describes algorithms used to derive the position, velocity, and attitude of a platform/vehicle (i.e., a UAV) from inertial sensor readings. An Earth-frame mechanization (Titterton and Weston 1997) of the inertial localization equations is presented. The use of the Earth frame allows for localization in a global sense and over long periods and provides a convenient reference frame for all of the different sensor modalities that are used in the Kalman filter. The equations are presented as first-order differential equations that can then be numerically integrated from a known starting position and orientation to compute the position and orientation of a platform as a function of time. The use of initial estimates of position, velocity, and attitude and the continuing numerical integration of the following equations using inertial sensor readings onboard a moving platform is collectively referred to as an inertial localization system (ILS) and is a key component of a working aided inertial localization system.

22.3.1 Earth-Frame Mechanization: Position and Velocity Equations

The position of the vehicle is defined as the vector from the ECEF to the body-fixed frame (\mathbf{p}^e). It is assumed that the body frame axes are defined as being at the center of the IMU in order to simplify the equations without loss of generality.

The position vector can be described in either Cartesian or polar coordinates where Eq. 22.6 provides the relationship between each:

$$\mathbf{p}^e = \begin{bmatrix} \left(\frac{a}{\sqrt{1-e^2 \sin^2 \lambda}} + h \right) \cos(\lambda) \cos(\varphi) \\ \left(\frac{a}{\sqrt{1-e^2 \sin^2 \lambda}} + h \right) \cos(\lambda) \sin(\varphi) \\ \left(\left[\frac{a}{\sqrt{1-e^2 \sin^2 \lambda}} \right] (1-e^2) + h \right) \sin(\lambda) \end{bmatrix} \quad (22.6)$$

where $a = 6,378,137$ m is the semimajor axis and $e = 0.08181919$ is the eccentricity of the Earth's ellipsoid, h is the altitude of the vehicle (above mean sea level (MSL)), and λ, φ are the geodetic latitude and longitude of the vehicle. The rate of change of the position and velocity of the vehicle in the ECEF frame is given by Eqs. 22.7 and 22.8:

$$\dot{\mathbf{p}}^e = \mathbf{v}^e \quad (22.7)$$

$$\dot{\mathbf{v}}^e = \mathbf{C}_b^e \mathbf{f}^b - 2(\omega_{ie}^e \times \mathbf{v}^e) + \mathbf{g}_l^e \quad (22.8)$$

where \mathbf{f}^b is the specific force measurement from the IMU, transformed into body-axis coordinates:

$$\mathbf{f}^b = \mathbf{f}^{\text{IMU}} - \omega_{ib}^b \times (\omega_{ib}^b \times \mathbf{l}^{\text{IMU}}) \quad (22.9)$$

where \mathbf{f}^{IMU} is the specific force vector measurement from the IMU (corrected for IMU biases), ω_{ib}^b is the IMU gyro measurement, and \mathbf{l}^{IMU} is the lever arm of the IMU with respect to body coordinates.

ω_{ie}^e is the rotation rate vector of the Earth with respect to inertial space in the Earth frame, given by Eq. 22.10:

$$\omega_{ie}^e = \begin{bmatrix} 0 \\ 0 \\ \omega_{\text{Earth}} \end{bmatrix} \quad (22.10)$$

where $\omega_{\text{Earth}} = 7.292115 \times 10^{-5}$ rad/s, and the term \mathbf{g}_l^e is given by

$$\mathbf{g}_l^e = \mathbf{g}^e - \omega_{ie}^e \times (\omega_{ie}^e \times \mathbf{p}^e) \quad (22.11)$$

$$= -g \begin{bmatrix} \frac{\mathbf{p}_x^e}{|\mathbf{p}^e|} \\ \frac{\mathbf{p}_y^e}{|\mathbf{p}^e|} \\ \frac{\mathbf{p}_z^e}{|\mathbf{p}^e|} \end{bmatrix} - [\times \omega_{ie}^e]^2 \mathbf{p}^e \quad (22.12)$$

where g is the acceleration acting on the vehicle due to gravity, referenced in ECEF coordinates and $[\times \omega_{ie}^e]$ is the skew symmetric matrix of the Earth's rotation rate vector.

22.3.2 Earth-Frame Mechanization: Attitude Equations

The attitude equations maintain a representation of the body to LVLH frame transformation \mathbf{C}_b^n which is used to compute the body to ECEF frame transformation \mathbf{C}_b^e :

$$\mathbf{C}_b^e = \mathbf{C}_n^e \mathbf{C}_b^n \quad (22.13)$$

where \mathbf{C}_n^e is the transformation from LVLH coordinates to ECEF coordinates, dependent on the global position of the vehicle:

$$\mathbf{C}_n^e = \begin{bmatrix} -\sin(\lambda) \cos(\varphi) & -\sin(\varphi) & -\cos(\lambda) \cos(\varphi) \\ -\sin(\lambda) \sin(\varphi) & \cos(\varphi) & -\cos(\lambda) \sin(\varphi) \\ \cos(\lambda) & 0 & -\sin(\lambda) \end{bmatrix} \quad (22.14)$$

The rates of change of quaternions and Euler angles for a frame transformation \mathbf{C}_b^n are found via Eqs. 22.15 and 22.17:

$$\dot{\mathbf{q}} = \mathbf{Q}_b^n \omega_{nb}^b \quad (22.15)$$

$$\mathbf{Q}_b^n = \frac{1}{2} \begin{bmatrix} -e_1 & -e_2 & -e_3 \\ e_0 & -e_3 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{bmatrix} \quad (22.16)$$

$$\dot{\Psi}^n = \mathbf{E}_b^n \omega_{nb}^b \quad (22.17)$$

$$\mathbf{E}_b^n = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (22.18)$$

where \mathbf{E}_b^n is the rotation rate transformation matrix and \mathbf{Q}_b^n is the quaternion rotation rate transformation matrix between the body frame and the LVLH frame and ω_{nb}^b is the rotation rate of the body frame with respect to the LVLH frame, measured in the body frame, which is computed:

$$\omega_{nb}^b = \omega_{ib}^b - \mathbf{C}_n^b (\omega_{ie}^n + \omega_{en}^n) \quad (22.19)$$

where $C_n^b = (C_b^n)^T$, ω_{ib}^b is the rotation rate measured by the gyroscopes inside the IMU, transformed into body-fixed coordinates, ω_{ie}^n , the Earth's rotation rate measured in the navigation frame, is given by

$$\omega_{ie}^n = \begin{bmatrix} \omega_{\text{EARTH}} \cos(\lambda) \\ 0 \\ -\omega_{\text{EARTH}} \sin(\lambda) \end{bmatrix} \quad (22.20)$$

and ω_{en}^n is the transport rate (rotation rate of the LVLH frame with respect to the ECEF frame as the vehicle moves around the curvature of the Earth):

$$\omega_{en}^n = \begin{bmatrix} \frac{v_y}{|\mathbf{p}^e|} \\ \frac{-v_x}{|\mathbf{p}^e|} \\ \frac{-v_y \tan(\lambda)}{|\mathbf{p}^e|} \end{bmatrix} \quad (22.21)$$

where $|\mathbf{p}^e|$ is the magnitude of the vehicle's position vector with respect to the surface of the Earth.

22.3.3 Simplification of Inertial Navigation Equations for Small Area Operation

When a platform operates over a small geographic area of the Earth's surface (with respect to the curvature of the Earth) and for a short amount of time (relative to the rotational period of the Earth), several simplifications can be made to the equations for inertial localization in order to simplify their implementation and analysis. In this case, the position and attitude of the vehicle's body-fixed frame are maintained with respect to a local navigation frame which is fixed to the Earth's surface rather than the ECEF frame. An assumption is made that the local navigation frame is an inertial frame of reference by ignoring the small Coriolis and centripetal accelerations and rotation rate which are incurred by the Earth's rotation. In this case, the position and velocity rates of change of the vehicle reduce to Eqs. 22.22 and 22.23:

$$\dot{\mathbf{p}}^n = \mathbf{v}^n \quad (22.22)$$

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b + \mathbf{g}^n \quad (22.23)$$

where $\mathbf{g}^n = [0, 0, g]^T$ is the vector of acceleration due to gravity in the local navigation frame (typical value for $g = 9.81 \text{ m/s}^2$ at sea level). The attitude of

the vehicle is now represented using the DCM C_b^n , and Eqs. 22.15 and 22.17 are used to find the attitude where $\omega_{ba}^a = \omega_{nb}^b$, the rotation rate of the body frame with respect to the local navigation frame measured in the body frame, which is approximated as

$$\omega_{nb}^b = \omega_{ib}^b \quad (22.24)$$

22.3.4 Numerical Integration of Inertial Equations in the ILS

The first-order differential equations described above are typically numerically integrated, starting from initial estimates of the UAV's position, velocity, and attitude and discrete-time samples of specific force and rotation rate provided by the IMU. For simplicity, a first-order numerical Euler integration step is described here which at each IMU measurement is used to predict forward the current position ($\mathbf{p}^e(k)$), velocity ($\mathbf{v}^e(k)$), and attitude ($\Psi^n(k)$ or $\mathbf{q}(k)$) estimates using Eqs. 22.7, 22.8, and 22.15 or 22.17 (depending on the use of quaternions or Euler angles):

$$\mathbf{p}^e(k) = \mathbf{p}^e(k-1) + \dot{\mathbf{p}}^e \Delta t \quad (22.25)$$

$$\mathbf{v}^e(k) = \mathbf{v}^e(k-1) + \dot{\mathbf{v}}^e \Delta t \quad (22.26)$$

$$\Psi^n(k) = \Psi^n(k-1) + \dot{\Psi}^n \Delta t \quad (22.27)$$

$$\mathbf{q}(k) = \mathbf{q}(k-1) + \dot{\mathbf{q}} \Delta t \quad (22.28)$$

where Δt is the time between subsequent inertial sensor measurements and k is used to indicate the timestep (corresponding to IMU measurements). Quaternion and/or Euler angle integration is performed before position and velocity integration in order to provide a more accurate attitude value for the computation of $\dot{\mathbf{p}}^e$ and $\dot{\mathbf{v}}^e$. If quaternions are used, they should also be normalized after each integration step:

$$\mathbf{q}(k) = \frac{\mathbf{q}(k)}{\sqrt{e_0^2 + e_1^2 + e_2^2 + e_3^2}} \quad (22.29)$$

Initial estimates of the UAV position, velocity, and attitude are usually derived from aiding sensor measurement; position and velocity are measured directly using satellite positioning systems, and attitude can be derived from a combination of IMU and magnetometer measurements while the vehicle is stationary (e.g., using the TRIAD algorithm (Shuster and Oh 1981)).

22.4 Aided Inertial Localization Using the EKF

In this section, algorithms are presented for using the EKF for fusing additional sensor measurements with position, velocity, and attitude dead reckoned by inertial localization.

22.4.1 Extended Kalman Filter Basics

The EKF is an algorithm for estimating the state of a dynamic system using noisy observations which are related to the state. The algorithm is recursive and requires a dynamic model (also known as a process model) of the estimated state and an observation model, both which are modeled as having additive noisy errors, where the noise probability distributions are white and Gaussian. The EKF works by a two-step cycle (a predict step followed by an observation step) where the estimate of the state is stored as a multivariate Gaussian probability distribution with mean $\hat{\mathbf{x}}(k)$ and covariance $\mathbf{P}(k)$ where k indicates the k th timestep in the estimation cycle:

$$\hat{\mathbf{x}}(k) = E[\mathbf{x}(k)] \quad (22.30)$$

$$\mathbf{P} = E[(\mathbf{x}(k) - \hat{\mathbf{x}}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T] \quad (22.31)$$

where $E[.]$ is the expectation operator. For more details on the operation of the EKF in general applications, the reader is referred to Gelb (1974) and Maybeck (1979).

22.4.2 Indirect Kalman Filter Architecture for Aided ILS

Although the equations that govern the process and observation models in inertial localization are nonlinear, a time-variant linear representation can be attained via the process of linearization. This can be achieved by recasting the perspective of the estimation task into an “indirect” form (Kim and Sukkarieh 2003). In the “indirect” form, a nominal system state is used which represents the position, velocity, and attitude of the UAV. This nominal system state contains an error which is the difference between the nominal state and the true state of the system (due to errors in the estimation process). In the “indirect” form, the task of the estimator is to estimate the error and thus use it to continuously correct the nominal system state computed by the inertial localization equations of motion (see Fig. 22.3). “Indirect” architectures are commonly used in aided inertial localization systems (Sukkarieh 1999; Kim and Sukkarieh 2004; Titterton and Weston 1997; Meyer-Hilberg and Jacob 1994); they provide equivalent results to the use of a “direct” EKF architecture (where the position, velocity, and attitude states themselves are estimated; see (Bar-Shalom and Fortmann 1998)) but are typically easier to implement and allow

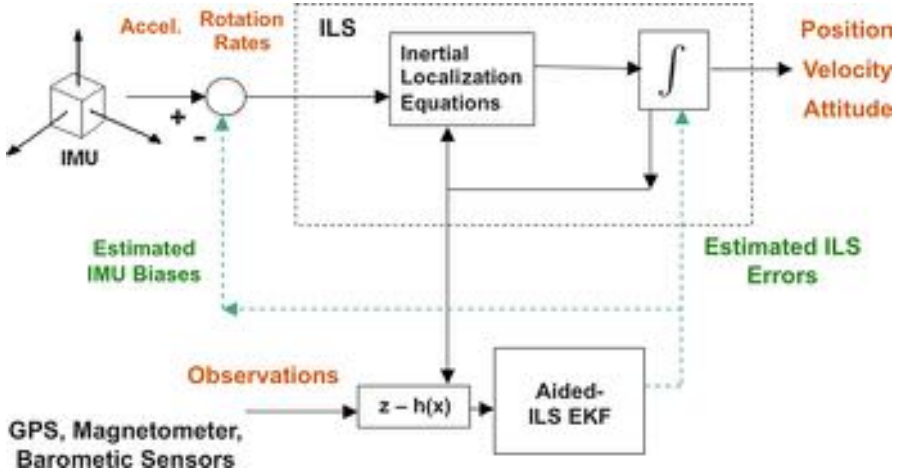


Fig. 22.3 Indirect Kalman filter architecture for aided ILS. An inertial localization system (ILS) is used to derived position, velocity, and attitude estimates through a process of state integration using measurements from an inertial measuring unit (IMU). When aiding sensor observations are available, they are used to estimate and correct for the errors in the ILS state estimates while also estimating and correcting for biases in the IMU measurements

for more flexibility in the use of different attitude representations in the inertial equations of motion through the use of an angle error model.

In this presentation, a psi-angle error model (Benson 1975) is used to represent the inertial localization equations in time-variant linear form. In the psi-angle error model, the estimated states are the vehicle position error ($\delta \mathbf{p}^e$), vehicle velocity error ($\delta \mathbf{v}^e$), and vehicle misalignment angles (or “psi angle”) ($\delta \Psi^n$). Typically inertial sensor biases must also be estimated online in practical aided inertial localization systems, due to their prevalence and time-varying nature in low-cost sensors; accelerometer bias ($\delta \mathbf{f}^b$) and gyro bias ($\delta \omega^b$) states are thus included in this presentation. Consider that the inertial sensor measurements have been made over a given time interval of flight, and by using an initial estimate of position, velocity, and attitude, a nominal set of localization states have been derived inside an ILS using Eqs. 22.25–22.28 recursively. The errors are defined as the difference between these computed states and the true state:

$$\delta \mathbf{p}^e = \hat{\mathbf{p}}^e - \mathbf{p}^e \quad (22.32)$$

$$\delta \mathbf{v}^e = \hat{\mathbf{v}}^e - \mathbf{v}^e \quad (22.33)$$

$$\delta \mathbf{f}^b = \hat{\mathbf{f}}^b - \mathbf{f}^b \quad (22.34)$$

$$\delta \omega_{ib}^b = \hat{\omega}_{ib}^b - \omega_{ib}^b \quad (22.35)$$

where $\hat{\mathbf{p}}^e, \hat{\mathbf{v}}^e, \hat{\mathbf{f}}^b, \hat{\omega}^b$ are the estimated values of the vehicle position, velocity, body-axis specific force, and rotation rates as computed in the ILS and $\mathbf{p}^e, \mathbf{v}^e, \mathbf{f}^b, \omega^b$ are the actual values of these states.

The misalignment angles ($\delta\Psi^e$) relate the true and inertial computed direction cosine matrices via

$$\mathbf{C}_b^e = [\mathbf{I}_{3 \times 3} - [\times \delta\Psi^e]] \hat{\mathbf{C}}_b^e \quad (22.36)$$

where \mathbf{C}_b^e is the true direction cosine matrix and $\hat{\mathbf{C}}_b^e$ is the computed direction cosine matrix within the ILS. In this presentation, the attitude relationship between the UAV body frame and the ECEF frame is maintained directly (i.e., using \mathbf{C}_b^e) for simplicity in the EKF model equations described in the section below. The advantage of the psi-angle model over a direct Jacobian of the Euler angle or quaternion equations (as used in a “direct” data fusion architecture) is that the error dynamics are based on the DCM, which avoids singularities during gimbal lock in Euler angles (when the UAV’s pitch equals $\pm 90^\circ$) and normalization constraints which are required when using quaternions.

The ILS maintains an internal state vector comprising the nominal states:

$$\hat{\mathbf{x}}(k) = [\mathbf{p}^e(k), \mathbf{v}^e(k), \Psi^n(k), \mathbf{f}^b(k), \omega_{ib}^b(k)]^T \quad (22.37)$$

The EKF maintains an internal state vector representing the current time (k) estimate of the ILS errors:

$$\delta\hat{\mathbf{x}}(k) = [\delta\mathbf{p}^e(k), \delta\mathbf{v}^e(k), \delta\Psi^e(k), \delta\mathbf{f}^b(k), \delta\omega_{ib}^b(k)]^T \quad (22.38)$$

Process and observation models for aided inertial localization are used in the EKF to maintain an estimate of the error states for each timestep k , which are discussed below.

22.4.3 Kalman Filter Error Process Model

The Kalman filter error process model maintains an estimate of the error and the error covariance of the ILS states in Eq. 22.37 which is used to help update and correct for the errors when additional localization sensor information is available. Based on the ILS equations in Sects. 22.3.1 and 22.3.2, an error process model that describes the propagation of the errors is used:

$$\dot{\delta\mathbf{p}}^e = \delta\mathbf{v}^e \quad (22.39)$$

$$\dot{\delta\mathbf{v}}^e = -([\times \omega_{ie}^e]^2 + \mathbf{M}_g) \delta\mathbf{p}^e - 2[\times \omega_{ie}^e] \delta\mathbf{v}^e + [\times \hat{\mathbf{f}}^e] \delta\Psi + \hat{\mathbf{C}}_b^e \delta\mathbf{f}^b + \hat{\mathbf{C}}_b^e \mathbf{w}_{\text{accel}}^b \quad (22.40)$$

$$\dot{\delta\Psi} = [\times \omega_{ie}^e] \delta\Psi + \hat{\mathbf{C}}_b^e \delta\omega^b + \hat{\mathbf{C}}_b^e \mathbf{w}_{\text{gyro}}^b \quad (22.41)$$

where $\mathbf{w}_{\text{accel}}^b$ and $\mathbf{w}_{\text{gyro}}^b$ are the accelerometer and gyro noise values, respectively, and

$$\hat{\mathbf{F}}^e = \begin{bmatrix} \hat{\mathbf{f}}_x^e & 0 & 0 \\ 0 & \hat{\mathbf{f}}_y^e & 0 \\ 0 & 0 & \hat{\mathbf{f}}_z^e \end{bmatrix} \quad (22.42)$$

where $[\hat{\mathbf{f}}_x^e, \hat{\mathbf{f}}_y^e, \hat{\mathbf{f}}_z^e]^T$ are the components of the specific force vector in ECEF coordinates. \mathbf{M}_g is the Jacobian of \mathbf{g}_l^e (see Eq. 22.12) with respect to $\delta \mathbf{p}^e$:

$$\mathbf{M}_g = \mathbf{g} \begin{bmatrix} \frac{x^2}{r^3} - \frac{1}{r} & \frac{xy}{r^3} & \frac{xz}{r^3} \\ \frac{xy}{r^3} & \frac{y^2}{r^3} - \frac{1}{r} & \frac{yz}{r^3} \\ \frac{xz}{r^3} & \frac{yz}{r^3} & \frac{z^2}{r^3} - \frac{1}{r} \end{bmatrix} \quad (22.43)$$

where $[x, y, z]^T = \mathbf{p}^e$ are the cartesian components of the ECEF position and $r = \sqrt{x^2 + y^2 + z^2}$.

The IMU measurement errors and biases can be modeled in various ways depending on the complexity of the sensors including states for biases, scale factor errors, and misalignments errors. For simplicity, a first-order, wandering bias and noise model is used:

$$\dot{\delta \mathbf{f}}^b = \mathbf{w}_{b,\text{accel}} \quad (22.44)$$

$$\dot{\delta \omega}_{ib}^b = \mathbf{w}_{b,\text{gyro}} \quad (22.45)$$

where $\mathbf{w}_{b,\text{accel}}$ and $\mathbf{w}_{b,\text{gyro}}$ are bias drift noises for the accelerometers and gyros, respectively. These first-order drift models account for slow, random drift in the biases over time.

The total error equations can thus be expressed in continuous-time form:

$$\dot{\delta \hat{\mathbf{x}}} = \mathbf{F}_c \delta \hat{\mathbf{x}} + \mathbf{G}_c \mathbf{w} \quad (22.46)$$

where

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -([\times \omega_{ie}^e]^2 + \mathbf{M}_g) & -2[\times \omega_{ie}^e] & [\times \hat{\mathbf{f}}^e] & \hat{\mathbf{C}}_b^e & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\times \omega_{ie}^e] & \mathbf{0} & \hat{\mathbf{C}}_b^e \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (22.47)$$

$$\mathbf{G}_c = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{C}}_b^e & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{C}}_b^e & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (22.48)$$

where \mathbf{w} is the combined state noise vector:

$$\mathbf{w} = [\mathbf{w}_{\text{accel}}^b, \mathbf{w}_{\text{gyro}}^b, \mathbf{w}_{b,\text{accel}}, \mathbf{w}_{b,\text{gyro}}]^T \quad (22.49)$$

The error equations are used to formulate a discrete process model via a first-order Euler integration:

$$\delta \hat{\mathbf{x}}(k) = \mathbf{F} \delta \hat{\mathbf{x}}(k-1) + \mathbf{G} \mathbf{w}(k) \quad (22.50)$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -([\times \omega_{ie}^e]^2 + \mathbf{M}_g) \Delta t & \mathbf{I}_{3 \times 3} - 2[\times \omega_{ie}^e] \Delta t & [\times \hat{\mathbf{f}}^e] \Delta t & \hat{\mathbf{C}}_b^e \Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} + [\times \omega_{ie}^e] \Delta t & \mathbf{0} & \hat{\mathbf{C}}_b^e \Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (22.51)$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{C}}_b^e \Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{C}}_b^e \Delta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} \Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} \Delta t \end{bmatrix} \quad (22.52)$$

22.4.4 Kalman Filter Update and Observation Models

Measurements made from other sensors onboard the UAV (satellite positioning systems and barometric and magnetic measurements) are used within the EKF update stage to estimate the error states of the ILS in Eq. 22.38 via the use of an observation model, which describes the relationship between the measurements

made and the EKF error states. The nominal form of the observation model relates the measurements to the nominal ILS states via a general function:

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{D}\mathbf{v}(k) \quad (22.53)$$

where $\mathbf{z}(k)$ is the observation data measured by the updating sensor, $\mathbf{h}(\cdot)$ is a set of functions relating the ILS state to the measurement (generally nonlinear), and \mathbf{v} is a vector which models the error, which is assumed to be a zero-mean, white Gaussian noise signal with covariance $\mathbf{R} = E[\mathbf{v}\mathbf{v}^T]$. In the “indirect” architecture, the EKF uses a form of the observation model which relates the measurements to the EKF error states:

$$\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}(k)) \approx \nabla \mathbf{H} \delta \hat{\mathbf{x}}(k) + \mathbf{D}\mathbf{v}(k) \quad (22.54)$$

$$\delta \mathbf{z}(k) \approx \nabla \mathbf{H} \delta \hat{\mathbf{x}}(k) + \mathbf{D}\mathbf{v}(k) \quad (22.55)$$

where $\delta \mathbf{z}(k)$, referred to as the “error observation,” is the difference between the sensor measurement and the expected sensor measurement $\mathbf{h}(\hat{\mathbf{x}}(k))$ computed by using the latest ILS state. This subsection describes models for several types of sensors typically used in UAV localization.

22.4.4.1 Satellite Positioning System Measurements

Satellite positioning systems typically provide measurements of the UAV’s position and velocity with respect to a global coordinate system (such as ECEF coordinates). Position measurements made by the sensor are related to the nominal system state via

$$\mathbf{z}_{\text{pos}} = \mathbf{p}^e - \mathbf{C}_b^e \mathbf{I}_{\text{GPS}} + \mathbf{v}_{\text{pos}} \quad (22.56)$$

where \mathbf{z}_{pos} is the measurement provided by the satellite positioning system (in ECEF coordinates), \mathbf{p}^e is the ECEF position of the UAV, \mathbf{v}_{pos} is the measurement noise, and \mathbf{I}_{GPS} is the lever arm of the satellite positioning receiver with respect to the UAV body frame. The observation model parameters used by the EKF are given by

$$\nabla \mathbf{H}_{\text{pos}} = [\mathbf{I}_{3 \times 3} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}] \quad (22.57)$$

$$\mathbf{D}_{\text{pos}} = \mathbf{I}_{3 \times 3} \quad (22.58)$$

Velocity measurements made by the sensor are related to the nominal system state via

$$\mathbf{z}_{\text{vel}} = \mathbf{v}^e - \mathbf{C}_b^e (\boldsymbol{\omega}_{ib}^b \times \mathbf{I}^{\text{GPS}}) + \mathbf{v}_{\text{vel}} \quad (22.59)$$

where \mathbf{z}_{vel} is the velocity measurement provided by the satellite positioning system (in ECEF coordinates), \mathbf{v}^e is the ECEF velocity of the UAV, and \mathbf{v}_{pos} is the measurement noise. The observation model parameters used by the EKF are given by

$$\nabla \mathbf{H}_{\text{vel}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (22.60)$$

$$\mathbf{D}_{\text{vel}} = \mathbf{I}_{3 \times 3} \quad (22.61)$$

22.4.4.2 Magnetometer Measurements

Magnetometers typically provide two types of measurements: a measurement of the Earth's magnetic field vector referenced in the UAV body-fixed reference frame (\mathbf{B}_b) and a measurement of the magnetic heading angle ψ_{mag} which is derived from these first measurements. Depending on the type of information available, an aided inertial localization system may use either of these measurements.

When the complete magnetic field vector measurement is used, knowledge of the Earth's magnetic field pointing direction with respect to the ECEF frame (i.e., the vector \mathbf{B}_e) in the local region of the UAV's operation must also be known from a model of the Earth's magnetic field. The discussion of these models is beyond the scope of this chapter; the reader is referred to (Roithmayr 2004) for more details. The magnetometer measurement is related to the nominal states via

$$\mathbf{z}_{\text{mag}} = \mathbf{B}_b = \mathbf{C}_e^b \mathbf{B}^e + v_{\text{mag}} \quad (22.62)$$

$$= \hat{\mathbf{C}}_e^b [\mathbf{I}_{3 \times 3} - [\delta \Psi^e \times]] \mathbf{B}^e + v_{\text{mag}} \quad (22.63)$$

where v_{mag} is the measurement noise. The observation model parameters used by the EKF are given by

$$\nabla \mathbf{H}_{\text{mag}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \hat{\mathbf{C}}_e^b [\mathbf{B}^e \times] & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (22.64)$$

$$\mathbf{D}_{\text{mag}} = \mathbf{I}_{3 \times 3} \quad (22.65)$$

When only the magnetic heading measurement is known, the magnetometer measurement is related to nominal states via

$$\mathbf{z}_{\text{mag}} = \psi_{\text{mag}} + v_{\text{mag}} \quad (22.66)$$

where v_{mag} is the error in the magnetic heading reading, and ψ_{mag} has been adjusted to true north before applying the measurement in the EKF. The observation model parameters used by the EKF are given by

$$\nabla \mathbf{H}_{\text{mag}} = \begin{bmatrix} \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & [0, 0, 1] & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (22.67)$$

$$\mathbf{D}_{\text{mag}} = 1 \quad (22.68)$$

22.4.4.3 Barometric Sensor Measurements

Barometric sensors derive measurements of the wind-relative airspeed and altitude of the UAV above sea level using measurement of static and dynamic air pressure typically measured using a pitot-static system. Altitude measurements are related to the ECEF position of the UAV via

$$\mathbf{z}_{\text{alt}} = \sqrt{(\mathbf{p}_x^e)^2 + (\mathbf{p}_y^e)^2 + (\mathbf{p}_z^e)^2} - R_{\text{Earth}} + v_{\text{alt}} \quad (22.69)$$

where v_{alt} is the barometric altitude measurement noise. The observation model parameters used by the EKF are given by

$$\nabla \mathbf{H}_{\text{alt}} = \begin{bmatrix} \frac{\mathbf{p}_x^e}{|\mathbf{p}^e|} & \frac{\mathbf{p}_y^e}{|\mathbf{p}^e|} & \frac{\mathbf{p}_z^e}{|\mathbf{p}^e|} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (22.70)$$

$$\mathbf{D}_{\text{alt}} = 1 \quad (22.71)$$

where $|\mathbf{p}^e|$ is the magnitude of the vector \mathbf{p}^e .

Assuming the airspeed measurement is made along the noise of the UAV (i.e., the x-axis of the UAV body frame), airspeed measurements are related to the ECEF velocity of the UAV via

$$\mathbf{z}_{\text{as}} = \mathbf{v}_x^b = (\hat{\mathbf{C}}_e^b \mathbf{v}^e)_x + v_{\text{as}} \quad (22.72)$$

where $(\hat{\mathbf{C}}_e^b \mathbf{v}^e)_x$ is the x-axis component of the vector $\mathbf{v}^b = \hat{\mathbf{C}}_e^b \mathbf{v}^e$ and v_{as} is the airspeed sensor noise. The observation model parameters used by the EKF are given by

$$\nabla \mathbf{H}_{\text{as}} = [\mathbf{0}_{1,3}, \cos \psi \cos \theta, \sin \psi \cos \theta, -\sin \theta, \dots \quad (22.73)$$

$$-v_d \sin \psi \cos \theta - v_e \sin \theta, v_d \cos \psi \cos \theta + v_n \sin \theta, \dots$$

$$-v_e \cos \psi \cos \theta + v_n \sin \psi \cos \theta, \mathbf{0}_{1 \times 3}, \mathbf{0}_{1 \times 3}]$$

$$\mathbf{D}_{\text{as}} = 1 \quad (22.74)$$

where $[v_n, v_e, v_d]$ are the components of the velocity vector $\mathbf{v}^n = \mathbf{C}_e^n \mathbf{v}^e$.

22.4.5 Kalman Filter Prediction Stage

The Kalman filter prediction stage is run at the frequency of received IMU data and is used to propagate forward in time a prediction of the ILS states and predicted error covariance. At the beginning of each prediction step, the error in the ILS state, $\delta \hat{\mathbf{x}}(k)$, is assumed to have a zero-mean value (as it has been previously corrected

at the last EKF update stage or during initialization). Therefore, the estimate of the error's mean value is propagated via Eq. 22.50 such that

$$\delta \hat{\mathbf{x}}(k) = \mathbf{F} \delta \hat{\mathbf{x}}(k-1) \quad (22.75)$$

$$= \mathbf{F} \mathbf{0} \quad (22.76)$$

$$= \mathbf{0} \quad (22.77)$$

Although the mean value of the error state $\delta \hat{\mathbf{x}}(k)$ is assumed to be zero, the estimated covariance of the error representing the uncertainty in the error state value is nonzero and grows at each prediction step based on the initial error uncertainty and expected noise in the IMU measurements and is propagated forward via

$$\mathbf{P}(k) = \mathbf{F} \mathbf{P}(k-1) \mathbf{F}^T + \mathbf{G} \mathbf{Q} \mathbf{G}^T \quad (22.78)$$

where $\mathbf{Q} = E[\mathbf{w} \mathbf{w}^T]$ is the covariance of the noise on the accelerometers, gyros, and IMU bias drift noises. The EKF prediction stage is calculated each time an IMU measurement is made.

22.4.6 Kalman Filter Update Stage

When an observation from any of the aiding sensors (such as the satellite positioning system, magnetometer, barometric sensors) is available, the EKF update stage is used to estimate the EKF error state and use this estimate to correct the ILS. The mean and covariance of the EKF error states after the update are computed:

$$\hat{\delta \mathbf{x}}(k) = \mathbf{W}(k) \delta \mathbf{z}(k) \quad (22.79)$$

$$\mathbf{W}(k) = \mathbf{P}(k) \nabla \mathbf{H}^T \mathbf{S}^{-1}(k) \quad (22.80)$$

$$\mathbf{S}(k) = \nabla \mathbf{H} \mathbf{P}(k) \nabla \mathbf{H}^T + \mathbf{D} \mathbf{R} \mathbf{D}^T \quad (22.81)$$

where $\mathbf{P}(k)$ is the current value of estimate covariance given by the prediction stage and $\mathbf{R} = E[\mathbf{v}(k) \mathbf{v}(k)^T]$ is the covariance of the observation noise \mathbf{v} associated with the updating sensor making the measurement. The estimate $\hat{\delta \mathbf{x}}$ is then used to correct the inertial localization system position, velocity, attitude, and IMU bias states via

$$\mathbf{p}^e = \mathbf{p}^e - \delta \hat{\mathbf{p}}^e \quad (22.82)$$

$$\mathbf{v}^e = \mathbf{v}^e - \delta \hat{\mathbf{v}}^e \quad (22.83)$$

$$\mathbf{C}_b^e = [\mathbf{I}_{3 \times 3} + [\delta \hat{\Psi} \times]] \hat{\mathbf{C}}_b^e \quad (22.84)$$

The Euler angles or quaternions representing the orientation of the body axes with respect to the local navigation frame axes are computed from \mathbf{C}_b^n , which is computed by

$$\mathbf{C}_b^n = (\mathbf{C}_n^e)^T \mathbf{C}_b^e \quad (22.85)$$

where \mathbf{C}_n^e is computed using \mathbf{p}^e and Eq. 22.14. The estimated IMU biases and scale factors and the bias and scale factor error estimates on the other localization sensors are also fed back to the ILS and used to correct the data coming from the IMU using Eqs. 22.34–22.35.

Finally, the estimated covariance of the error state is updated from the observation:

$$\mathbf{P}(k)^+ = \mathbf{P}(k) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}(k)^T \quad (22.86)$$

where the subscript $+$ indicates the covariance matrix after the update.

22.4.6.1 Detecting and Rejecting Measurement Outliers

In addition to sensor errors and uncertainty modeled by noise, in practice, sensor measurements may contain unmodeled error sources that occur too infrequently for incorporating into the models described in Sect. 22.4.4. The erroneous measurements can be treated as outliers and rejected from the EKF update stage during the EKF operation. When measurements are made by aiding sensors, they are tested through an “outlier gate” to test for their likelihood of being an outlier using the error observation $\delta\mathbf{z}(k)$ in the EKF update stage.

For each observation from an aiding sensor, the normalized innovation squared $\gamma(k)$ is computed:

$$\gamma(k) = \delta\mathbf{z}^T(k)\mathbf{S}^{-1}(k)\delta\mathbf{z}(k) \quad (22.87)$$

For a correctly tuned filter, $\delta\mathbf{z}(k)$ should follow a zero-mean distribution with covariance $\mathbf{S}(k)$ (i.e., the innovation covariance) which is computed using the assumed sensor uncertainty encoded in \mathbf{R} and the current time estimate of the error state covariance, encoded in $\mathbf{P}(k)$. Measurements that correspond to inliers will have normalized innovation squared $\gamma(k)$ values that average to $\gamma(k) = 1$ and follow a χ^2 distribution in the number of observation variables in $\delta\mathbf{z}(k)$ (i.e., the dimension of the observation vector). Based on confidence intervals in the χ^2 distribution, a threshold on $\gamma(k)$ is obtained such that observations whose normalized innovation squared values exceed the threshold are statistically unlikely and rejected as outliers.

22.5 Example: Inertial-GPS Localization on a UAV

In this section, results of the aided inertial localization algorithms described above when implemented onboard a small UAV are provided.



Fig. 22.4 UAV and localization sensor payload system: *left*, the J3 Cub, a small UAV used primarily for airborne photographic mapping; *right*, the sensor payload box carried onboard the UAV consisting of a triaxial IMU, GPS receiver, and PC104 computer stack for logging and processing data

22.5.1 Experimental Setup

Sensor data was collected during the flight of a small UAV, the J3 Cub (see Fig. 22.4) which carried a payload consisting of a low-cost IMU, differentially corrected GPS receiver, and a color camera system. The UAV is used primarily for photogrammetric mapping applications which require accurate estimates of the UAV's position and orientation throughout its flight path. The vehicle achieves speeds of 25 m/s and roll rates of up to 100°/s, necessitating the use of inertial sensors. The flight path of the vehicle is set by an onboard autonomous control system, with a preplanned flight pattern used to provide sensor coverage (using the onboard camera) with a fixed altitude over a region of interest for mapping.

Sensor data from the IMU and GPS was collected during a 65-min flight test of the UAV and processed using the ILS and EKF algorithms described in Sects. 22.3 and 22.4 to estimate the position, velocity, attitude, and IMU sensor biases throughout the flight. Specifications for the IMU and GPS receiver are shown in Table 22.1. Initial state estimates were computed using GPS and IMU measurements and an external heading measurement while stationary on the ground before takeoff.

22.5.2 Results and Discussion

Figure 22.5 illustrates the complete trajectory for the vehicle comparing position observations for the onboard GPS receiver, position estimates derived purely from

Table 22.1 Sensor payload specifications: the sensor payload consists of an low-cost IMU and GPS receiver

| GPS receiver | | IMU | |
|----------------|-----------------------|------------------------------|------------------------------------|
| Sampling rate | 5 Hz | Sampling rate | 100 Hz |
| Position error | 1 m (1σ) | Accelerometer noise | 0.05 m/s^2 (1σ) |
| Velocity error | 10 cm/s (1σ) | Gyro noise | 0.05 deg/s (1σ) |
| | | Accelerometer bias stability | $\pm 0.05 \text{ m/s}^2$ |
| | | Gyro bias stability | $\pm 0.05 \text{ deg/s}$ |

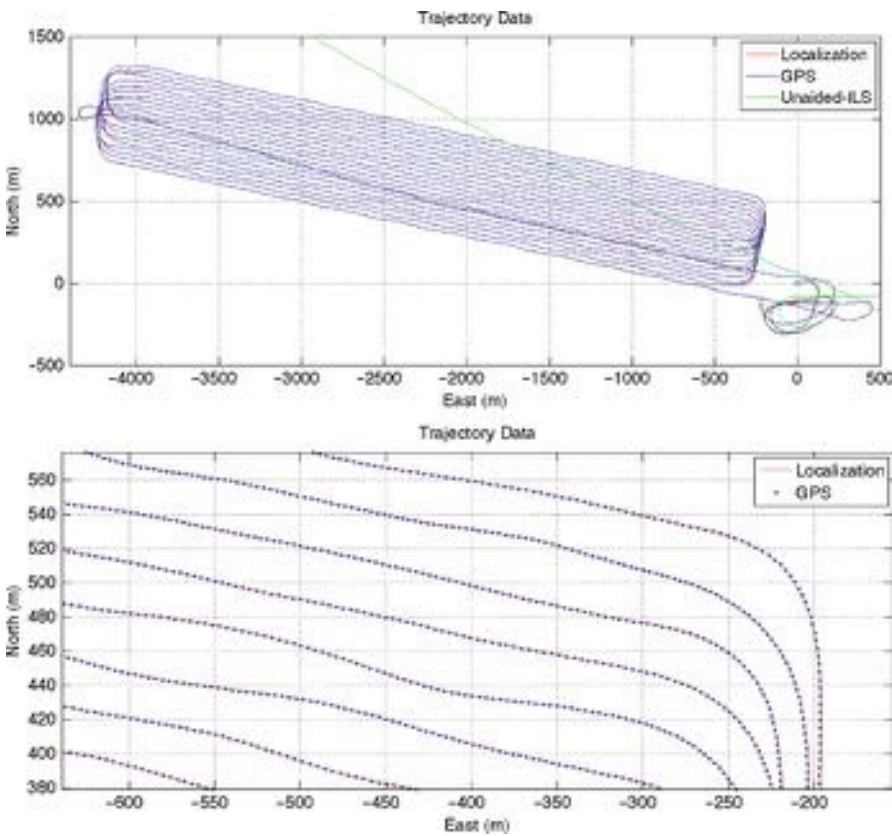


Fig. 22.5 Overhead view of UAV flight trajectory showing GPS-measured position, unaided ILS position, and the fused aided ILS EKF solution. Also shown is a zoomed-in section of the trajectory highlighting performance of the estimator during banking maneuvers

unaided ILS (using initial position and alignment from the IMU and GPS receiver) and the fused GPS-aided ILS solution. The origin of the displayed coordinate system is centered on the starting location of the vehicle before takeoff. The vehicle

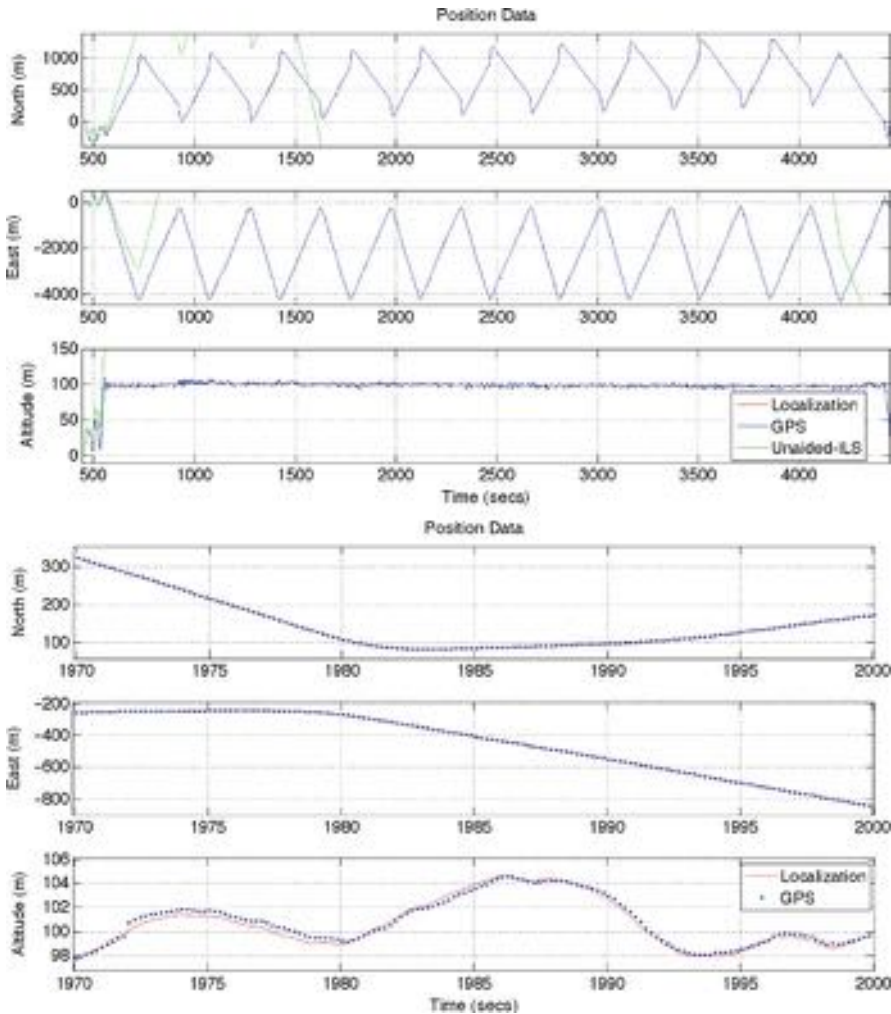


Fig. 22.6 ECEF UAV position trajectory showing GPS-measured position, unaided ILS position, and the fused aided-ILS EKF solution

takes off and performs a series of parallel transects at a fixed altitude before returning to the origin for landing. The GPS-aided ILS trajectory remains close to the GPS measurements during the flight, indicating a constraint in the position errors, whereas the unaided ILS position estimates drift significantly over time due to unbounded growth in the ILS dead-reckoning errors.

Figure 22.6 illustrates the components of the local navigation frame position of the vehicle (north, east, altitude) over the time of the flight, with a highlighted

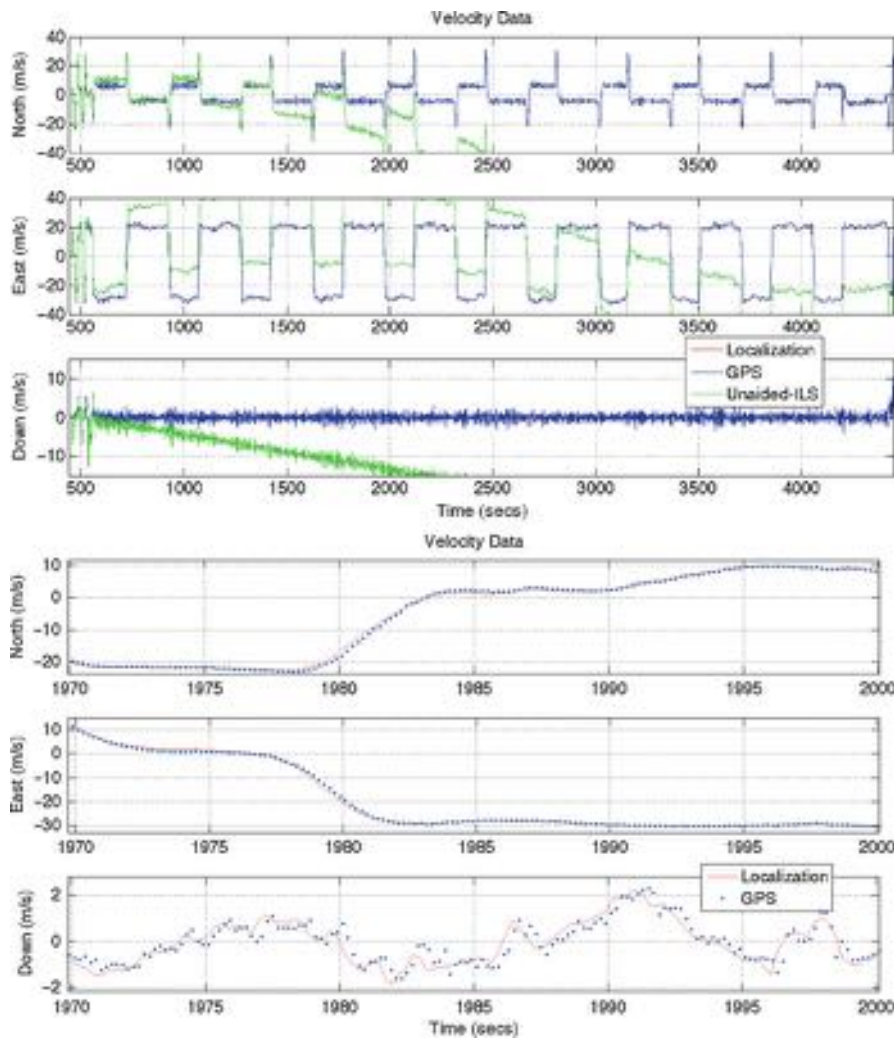


Fig. 22.7 ECEF UAV velocity showing GPS-measured velocity, unaided-ILS velocity and the fused aided ILS EKF solution

section of the flight shown in the second subfigure. The subfigure highlights an example of GPS outlier detection (see $t \approx 1972$ s) where a series of GPS measurements are seen to “jump” to a new position solution for a few seconds before returning to a stable position solution. Such outliers are typically observed in satellite positioning systems when, for example, satellites move in and out of view of the positioning system receiver. This series of outliers is detected by the outlier rejection process of the filter and is rejected.

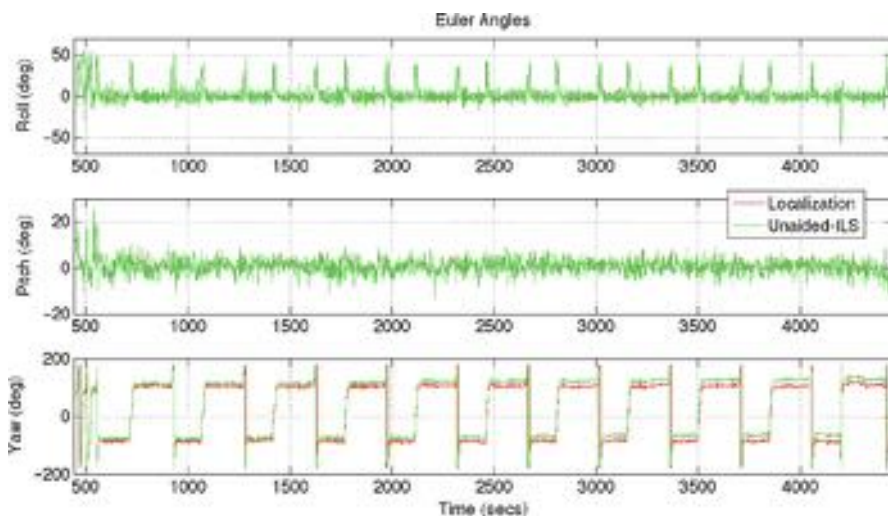


Fig. 22.8 UAV Euler angles showing unaided ILS attitude and the fused aided ILS EKF solution

Figure 22.7 shows a comparison of the GPS-measured platform velocity, unaided ILS velocity, and GPS-aided ILS velocity estimates. As in the case of position estimates, the unaided velocity estimate errors grow over time due to dead reckoning, whereas the GPS-aided velocity estimate errors are constrained by the filter corrections. The error growth in the position solution is observed to be much greater than the velocity error growth due the position information being derived from velocity (as a second-order derivative of acceleration).

Figure 22.8 illustrates the estimated Euler angles of the UAV transformed as to represent the rotations between the UAV body frame and the local navigation frame. The UAV undergoes banking turns and changes in heading as it completes the parallel trajectory transects, seen by the spikes in the roll angle data and changes in the yaw angle. The unaided ILS solution exhibits only a small degree of drift, mainly observed in the yaw angle, when compared to the error growth in both the position and velocity estimates. This is firstly due to the generally higher accuracy and bias stability of the gyro sensors with respect to accelerometers and secondly, and more importantly, to the fact that position and velocity dead-reckoning errors are also attributed to attitude errors themselves due to the transformation of IMU accelerometer measurements from body-fixed coordinates into navigation coordinates, proceeding numerical integration, which is performed using the Euler angles estimates.

Figure 22.9 shows the estimated accelerometer and gyro biases computed by the EKF and used to correct the IMU measurements in a feedback loop to the ILS. Estimation of these states relies on the motion of the UAV, thus resulting in a transient period of uncertainty in these estimates seen around takeoff, followed by a more stable estimate during the rest of the flight.

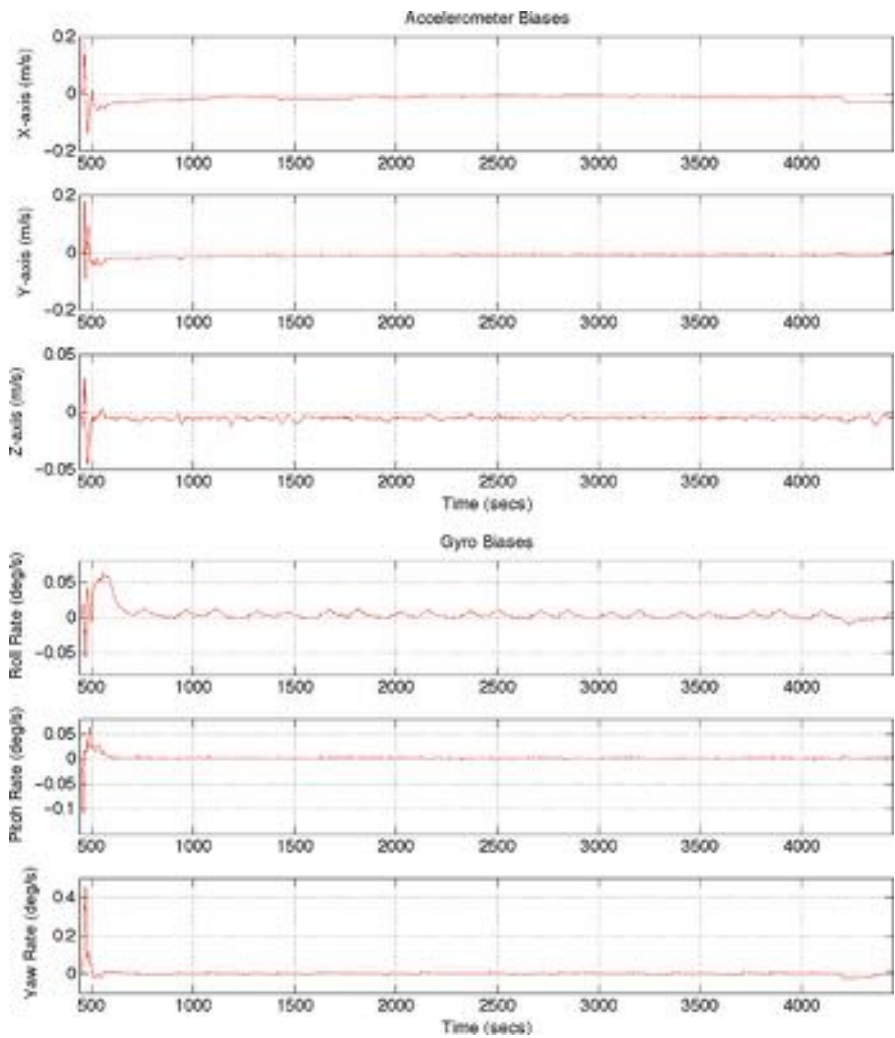


Fig. 22.9 IMU accelerometer and gyro bias estimates from the fused aided ILS EKF solution

Conclusion

This chapter has provided an overview of algorithms for aided inertial localization using the fusion of IMU measurements with measurements from sensors such as satellite positioning systems, magnetometers, and barometric sensors. The fusion of these sensors provides both accuracy and high-rate information about location including the position, velocity, and attitude of a UAV. The presentation in this chapter is based in the use of the EKF as the primary tool for data fusion and an “indirect” data fusion architecture based on an error state model, providing a simple and computationally efficient localization system.

References

- W. Baker, R. Clem, Terrain Contour Matching [TERCOM] primer. Technical Report ASP-TR-77-61, Aeronautical Systems Division, Wright-Patterson AFB, 1979
- I. Bar-Itzhack, D. Serfaty, Y. Vitek, Doppler-aided low-accuracy strapdown inertial navigation system. *J. Guid. Control* **5**(3), 236–242 (1982)
- Y. Bar-Shalom, T. Fortmann, *Tracking and Data Association* (Academic, Boston, 1998)
- D. Benson, A comparison of two approaches to pure-inertial and Doppler-inertial error analysis. *IEEE Trans. Aerosp. Electron. Syst.* **11**(4), 447–455 (1975)
- H. Carvalho, P.D. Morai, A. Monin, G. Salut, Optimal non-linear filtering in GPS/INS integration. *IEEE Trans. Aerosp. Electron. Syst.* **33**(3), 835–850 (1997)
- J. Fountain, Digital terrain systems, in *Airborne Navigation Systems Workshop, London* (The Institute of Electrical Engineers, London, 1997), pp. 1–6
- D. Gebre-Egziabher, R. Hayward, J. Powell, Low-cost GPS/inertial attitude heading reference system (AHRS) for general aviation applications, in *IEEE Position Location and Navigation Symposium*, Institute of Electrical Engineers, Piscataway, USA (1998)
- A. Gelb, *Applied Optimal Estimation* (MIT, Cambridge, 1974)
- C.S. Giovanni, Performance of a ring laser strapdown attitude and heading reference for aircraft. *J. Guid. Control* **2**(4), 320–327 (1979)
- A. Hiliuta, R. Landry, F. Gagnon, Fuzzy correction in a GPS/INS hybrid navigation system. *IEEE Trans. Aerosp. Electron. Syst.* **40**(2), 591–648 (2004)
- R. Kalman, A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Eng.* **82**(D), 35–45 (1960)
- J. Kim, S. Sukkarieh, Recasting SLAM – towards improving efficiency and platform independency, in *International Symposium on Robotics Research*, Springer-Verlag, New York (2003)
- J. Kim, S. Sukkarieh, SLAM aided GPS/INS navigation in GPS denied and unknown environments, in *International Symposium on GNSS/GPS* (2004). <http://www.gmat.unsw.edu.au/gnss2004unsw/toc.html>
- B. King, T. Kutta, *Impact: The History of Germany's V-Weapons in World War II* (Sarpedon Publishers, Rockville Center, New York, 1998)
- D. Knight, Rapid development of tightly-coupled GPS/INS systems, in *IEEE Position Location and Navigation Symposium* (1996)
- P. Maybeck, *Stochastic Models, Estimation and Control, Volume 1* (Academic, New York, 1979)
- J. Meyer-Hilberg, T. Jacob, High accuracy navigation and landing system using GPS/IMU system integration, in *IEEE Position, Location and Navigation Symposium*, Institute of Electrical Engineers, Piscataway, USA (1994)
- C.M. Roithmayr, Contributions of spherical harmonics to magnetic and gravitational fields, in *NASA Technical Report NASA/TM2004213007* (2004). http://nssdcftp.gsfc.nasa.gov/models/geomagnetic/igrf/old_matlab_igrf/Contributions.pdf
- M. Shuster, S. Oh, Three-axis attitude determination from vector observations. *J. Guid. Control* **4**(1), 7077 (1981)
- S. Sukkarieh, Aided inertial navigation systems for autonomous land vehicles. Ph.d, University of Sydney, 1999
- D. Titterton, J. Weston, *Strapdown Inertial Navigation Technology* (Peter Peregrinus Ltd., London, 1997)
- Y. Yang, J. Farrell, M. Barth, High-accuracy, high-frequency differential carrier phase GPS aided low-cost INS, in *IEEE Position, Location and Navigation Symposium*, Institute of Electrical Engineers, Piscataway, USA (2000)

Matthew F. Ridley, Ben Upcroft, and Salah Sukkarieh

Contents

| | | |
|--------|--|-----|
| 23.1 | Introduction | 462 |
| 23.1.1 | Related Work | 463 |
| 23.1.2 | Applications of DDF | 463 |
| 23.2 | Decentralized Data Fusion..... | 464 |
| 23.2.1 | The Decentralized Bayesian Estimator | 465 |
| 23.2.2 | Local Filter..... | 465 |
| 23.2.3 | Node-Node Fusion | 467 |
| 23.3 | Representations for Implementation of DDF | 469 |
| 23.3.1 | General Requirements..... | 469 |
| 23.3.2 | Approaches and Limitations | 469 |
| 23.3.3 | Alternatives for Probabilistic Representation | 470 |
| 23.3.4 | Storage Comparison | 471 |
| 23.4 | Gaussian Mixture-Based Decentralized Data Fusion | 472 |
| 23.4.1 | Introduction | 472 |
| 23.4.2 | Local State Estimation | 473 |
| 23.4.3 | Node-Node Fusion | 475 |
| 23.5 | Experimental Hardware and Algorithms..... | 477 |
| 23.5.1 | Flight Platform..... | 477 |
| 23.5.2 | Sensor Model and Transforms | 478 |
| 23.5.3 | Extraction of Features | 479 |

M.F. Ridley (✉)

Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW, Australia

e-mail: matthew@ridley.id.au

B. Upcroft

School of Electrical Engineering and Computer Science, Brisbane, QLD, Australia

e-mail: ben.upcroft@qut.edu.au

S. Sukkarieh

Australian Centre for Field Robotics (ACFR), School of Aerospace, Mechanical & Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW, Australia

e-mail: salah@acfr.usyd.edu.au

- 23.5.4 Observation Transformations 480
 - 23.5.5 Process Model 481
 - 23.5.6 Feature Maintenance 482
- 23.6 Example Implementation with Gaussian and GMM 483
 - 23.6.1 Data Set 483
 - 23.6.2 Platform and Component Topology 483
 - 23.6.3 Comparison to Gaussian Representation 483
 - 23.6.4 Sample Feature Results 484
 - 23.6.5 Discussion 485
- 23.7 Conclusion 487
- References 488

Abstract

This chapter describes decentralized data fusion algorithms for a team of multiple autonomous platforms. Decentralized data fusion (DDF) provides a useful basis with which to build upon for cooperative information gathering tasks for robotic teams operating in outdoor environments. Through the DDF algorithms, each platform can maintain a consistent global solution from which decisions may then be made. Comparisons will be made between the implementation of DDF using two probabilistic representations. The first, Gaussian estimates and the second Gaussian mixtures are compared using a common data set. The overall system design is detailed, providing insight into the overall complexity of implementing a robust DDF system for use in information gathering tasks in outdoor UAV applications.

23.1 Introduction

Cooperative information gathering tasks involving multiple autonomous platforms can be formulated for many types of hardware and software configurations. In all cases, teams of robots must efficiently communicate their observations of the world and together build a representative map of the environment that they operate in. This chapter will describe how a team of autonomous, mobile platforms in an outdoor environment can cooperatively build a map of the observed features.

This chapter concentrates on the implementation of non-Gaussian DDF for outdoor environments using a robust software framework for team robotics, specifically the *development* and *implementation* of non-Gaussian decentralized data fusion algorithms within complex, large-scale robotic systems.

The remainder of this section outlines related research and applications. Section 23.2 describes the DDF architecture. Section 23.3 details the DDF algorithms specific for a variety of probabilistic representations. Gaussian mixture-based DDF is then focused on in Sect. 23.4. The experimental hardware and related algorithms then follow in Sect. 23.5, and results are analyzed in Sect. 23.6.

23.1.1 Related Work

Applications that benefit from multisensor data fusion include environmental sensing, surveillance, and search-and-rescue (Paskin and Guestrin 2004; Rosencrantz et al. 2003a; Bourgalt and Durrant-Whyte 2004). In each of these problems, individual nodes of the network make local measurements or observations of the common environment and attempt to combine the measurements to produce a global estimate of the observed state. The fusion approach adopted here is motivated by the need to survey and map large outdoor natural environments in which distributed sensor networks are prone to node and/or communication failure. In contrast to hierarchical and centralized distributed methods (Hashemipour et al. 1988; Nii 1986), decentralized architectures ensure robustness to communication failures while allowing scalability and modularity (Manyika and Durrant-Whyte 1994). These properties arise as there are no central nodes to which global knowledge is communicated.

Previous approaches to robust decentralized data fusion have included tracking position features provided by range devices such as radar or laser (Rosencrantz et al. 2003a), monitoring temperature (Paskin and Guestrin 2004), people movement (Brooks et al. 2004) in an office environment, or tracking artificial visual features with known range (Nettleton et al. 2000). This chapter will expand upon visual features with known range to include those with a maximum range, i.e., a more realistic bearing only sensor model.

Non-Gaussian distributed algorithms for sensor calibration have previously been developed by Ihler et al. (2004), while Rosencrantz et al. (2003a) demonstrated non-Gaussian distributed state estimation in the context of robotic laser tag. In both cases, correlated estimation errors due to common information communicated between nodes in the past (also known as data incest) (Bar-Shalom 1990; Bar-Shalom and Li 1995) were not considered. Accounting for data incest in decentralized architectures is the key problem in ensuring mathematically consistent and convergent solutions. Implemented is an approximate node-node fusion algorithm (Upcroft et al. 2005) using Gaussian mixture models (GMMs) and a variant of the covariance intersect algorithm (Julier and Uhlmann 2001b). This algorithm provides consistent estimates in practice, but like Ihler and Rosencrantz's work, there is no guarantee of a convergent solution.

23.1.2 Applications of DDF

Decentralized data fusion provides a useful basis with which to build upon for cooperative information gathering tasks. Each platform obtains a consistent global solution upon which decisions can then be made. Any given platform can then operate with the confidence that all other platforms will make complimentary decisions, due to the consistency of information across all platforms. Further discussion of the suitability of the DDF algorithm for team problems has been made

by Grocholsky (2002). In a similar light the decentralized search and rescue problem has been investigated by Bourgault et al. (2003).

23.2 Decentralized Data Fusion

Decentralized data fusion (DDF) is the process of estimating states of features by combining information from multiple sensors connected to local processing nodes that are then interconnected in such a way that the system does not depend on a single centralized fusion center. Figure 23.1 depicts the major components of several decentralized sensing nodes.

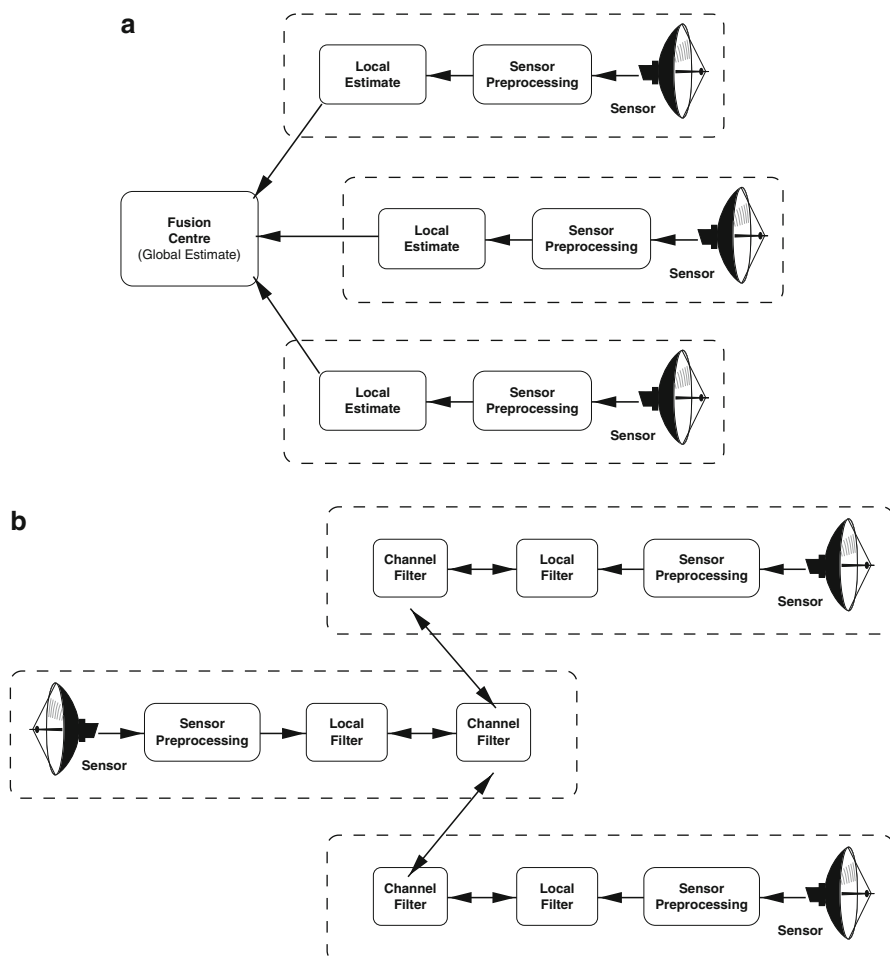


Fig. 23.1 Distributed, decentralized (with known and unknown hierarchy) data fusion architectures

Each decentralized sensor node typically has its own sensor, a local fusion center that contains global information, and a channel filter to facilitate the propagation of information throughout the network.

Decentralized systems are characterized by the following (Grime and Durrant-Whyte 1994):

- There is no single central decision center; no node should be central to the successful operation of the network.
- There is no common communication facility; nodes cannot broadcast results; and communication must be kept on a strictly node-to-node basis.
- Nodes do not have global knowledge of network topology; nodes should only know about connections in their own neighborhood.

These are considered as the three *axioms* of decentralized systems. A system cannot be considered truly decentralized if it does not adhere to these constraints.

The primary motivations to decentralize are:

Scalability The structure of communication within a decentralized system is such that communication bottlenecks are eliminated. The result is a system that scales independently of the number of nodes within the network, provided that independent communication links are available.

Robustness By removing the dependence on any centralized controller or fusion center, a decentralized system will tend toward a graceful degradation in performance rather than a system-wide failure in the case of a distributed system.

Modularity The modular nature of a decentralized system allows nodes to be added and removed at any time to suit the requirements at any given time.

Decentralized sensor nodes are required to have significantly more processing capability to provide for the fusion of both local and network information. Sensor nodes developed for distributed systems typically do not possess sufficient storage or processing capabilities to operate effectively within a decentralized sensor network.

23.2.1 The Decentralized Bayesian Estimator

This section describes the decentralized Bayesian estimation approach in the form developed by Grime (1992). The basic approach has changed little since then. However, it remains key to the successful operation of a decentralized sensor network. The following sections will describe the basic components of a decentralized sensor node and the interactions between them.

Figure 23.2 details the internal operations of a decentralized Bayesian estimator. To simplify the situation, only one channel filter is shown (there would be one for every node the node is connected to).

23.2.2 Local Filter

The local filter operates as a normal Bayesian estimator. The only difference is that there are now two sources from which new information may be gained:

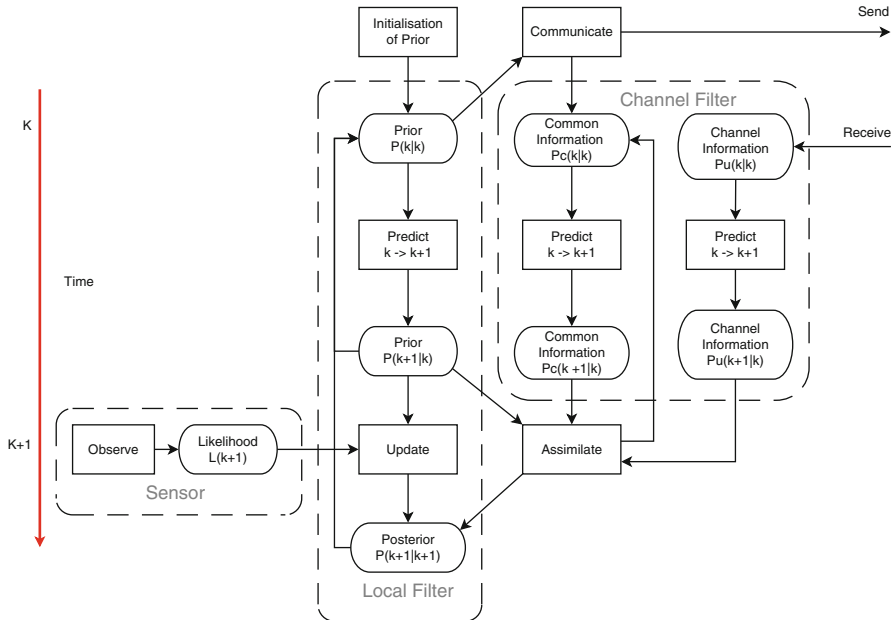


Fig. 23.2 General recursive decentralized Bayesian estimator

the local sensor and the remotely connected node. The local filter essentially operates in complete ignorance of all the decentralized aspects. The only difference being that a channel update/assimilation may occur, involving a full probability distribution as opposed to a sensor likelihood. So long as the local filter has a means to perform both likelihood updates and full state updates, no modification is required.

23.2.2.1 Local State Estimation

The recursive update after an observation is given by Bayes theorem:

$$p(x_k|Z^k) = \frac{p(z = z_k|x_k)p(x_k|Z^{k-1})}{p(z_k|Z^{k-1})} \quad (23.1)$$

where Z^k are observations of a state x_k at time t_k , $p(z|x_k)$ is the likelihood model, $p(x_k|Z^{k-1})$ is the prediction from the posterior over the previous state, and $Z^k = \{z_k, Z^{k-1}\}$ is the set of observations from all nodes in the DDF network.

The local prediction step is given by the Chapman-Kolmogorov equation:

$$p(x_k|Z^{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Z^{k-1}, x_0)dx_{k-1} \quad (23.2)$$

where $p(x_k|x_{k-1})$ is the transition probability density and $p(x_{k-1}|Z^{k-1}, x_0)$ is the updated estimate from the previous time step.

23.2.3 Node-Node Fusion

All local node information is transmitted to neighboring platforms as a posterior probability distribution. The net result is that each platform locally maintains a complete map (or belief) of all features observed by all nodes in the network. Multiple observations of the same feature, possibly by different platforms, result in an increasingly accurate estimate of the feature location for *all* nodes.

Use of the covariance intersect algorithm (for Gaussian distributions) as described by Julier (Julier and Uhlmann 2001b) and Sect. 23.4.3.1 provides a means to generate conservative updates between multiple decentralized nodes. This process eliminates the need for a channel filter but is typically more conservative than otherwise desirable.

Communicating increments of information updates eliminates the need for a channel filter. However, this approach is not robust to communication failure and is seldom used for this very reason. In the event of communication failure, the channel filter on a given node may become out of sync with the actual amount of common information a node contains. While this is not disastrous, the result is less than optimal as excessive information will be removed from incoming transmissions, until the channel filters realign.

23.2.3.1 The Channel Filter

The purpose of the channel filter is to maintain the information about a particular feature that a node has in common with another that it is connected to within a network. Without this, assimilation of information from the network may result in double counting of information previously sent, resulting in overconfident estimates. This is the area of primary concern, as removal of common information from communicated distributions other than Gaussians is nontrivial.

It can be shown (Bar-Shalom 1990; Bar-Shalom and Li 1995) that fusion of the raw correlated information between nodes i and j is

$$p(x|Z_i \cup Z_j) = \frac{1}{c} \frac{p(x|Z_i)p(x|Z_j)}{p(x|Z_i \cap Z_j)} \quad (23.3)$$

where $Z_{i(j)}$ are all the observations available to node i (j), $p(x|Z_i \cup Z_j)$ is the posterior probability over the unknown state given information from both nodes, $p(x|Z_{i(j)})$ are the posteriors based only on locally available information, $p(x|Z_i \cap Z_j)$ is the information the two nodes have in common, and c is a normalizing constant.

The immediate consequence of this is that channel update operations require both a multiplication and a division operation to assimilate channel information. When operating with log probabilities or an information filter, this can be accomplished with addition and subtraction operations. However, this form is seldom used (apart from the information filter) as addition and subtraction is not closed in form for most representations. Thus, the problem of constructing the union $Z_i \cup Z_j$ reduces to finding the common information $Z_i \cap Z_j$ and is the *key* to the decentralized communication problem. The incorporation of redundant information in DDF systems may lead to bias, overconfidence, and divergence in estimates.

23.2.3.2 Send Operation

As local observations are made at a sensor node, the local filter accrues information. At some point, this information will be transmitted to a remote node. This may occur periodically (as in the most simple case), or additional channel management components will determine this according to its own rules.

When a feature is communicated to a connected node, the contents of the local filter are directly copied into the corresponding channel filter, and the data is then transmitted down the link. Assuming no communication failures, the contents of the channel filter will contain all that is common between the two nodes.

The full state is always communicated, as per Nettleton (2003), as this ensures that in the event of communication failure, all sensor nodes may recover, and no information will be lost.

23.2.3.3 Receive Operation

Once a message is received, temporal alignment of any information received must be performed. Either the received information is propagated forward in time until it reaches that of the local filters, or the local filters are propagated forward to the corresponding time of the received data.

Any incoming data must first be matched against all local filters, to determine which feature the data should be associated with, i.e., data association. This incoming data will contain information that is common between the two nodes. This must now be removed prior to fusion within the local filter.

In state space this is achieved by dividing the incoming probability distribution by the distribution forming the contents of the channel filter. In Fisher space, this would correspond to subtracting the equivalent forms. The result is termed the “information gain” or “new information.”

This new information may then be fused into the local filter. At the same time, an update must be performed on the channel filter with this new information that was previously calculated. This corresponds to a multiplication in state space or an addition in Fisher space. The channel filter must again contain all information that is common between the two nodes, so the contents of the local filter are copied to the channel filter.

23.3 Representations for Implementation of DDF

This section will outline a variety of means to communicate multimodal probability distributions within a sensor network and then to assimilate this information without overconfidence into the destination nodes. Most non-Gaussian representations will require an operation that will result in a less than optimal solution. The operations of primary concern are prediction, Bayesian updates, and assimilation of channel information.

The objective of this section is to identify a *specific* representation and then develop algorithms with which to *efficiently* manipulate these representations, yet manage the expected deviation from an optimal solution.

23.3.1 General Requirements

Assimilation of information communicated within particular channels requires both a product and a division. In log space this may be achieved by addition and subtraction. As for the information filter, operating in Fisher space provides similar such benefits. Local updates from locally attached sensors require only a product or addition depending on the representation chosen.

The time evolution (or prediction) of a probability distribution must also be performed at regular intervals. This is required for both local filters and channel filters. As such it occurs more frequently than that of channel and local update operations. Prediction operations therefore need to be the most efficient of the two.

23.3.2 Approaches and Limitations

There are a variety of techniques for alternate representations of probability distributions to improve the performance of tracking algorithms for nonlinear and/or non-Gaussian sensors and/or systems. Such approaches include the particle filter by Gordon et al. (1993); Gaussian mixtures by Alspach and Sorenson (1972), Ito and Xiong (2000), and Robert (1996); support vector machines by Challa et al. (2002); or even discretized distributions by Stone et al. (1999).

The desire for richer representations in decentralized systems (Ridley et al. 2002) has been present for some time. However, decentralized data fusion techniques have generally remained exclusively tied to Gaussian representations. A method to decentralize a particle filter is described by Rosencrantz et al. (2003a,b). However, this method did not have a means to directly calculate the common information shared between the nodes, and its operation did not guarantee a consistent result at each node. The approach communicated information increments, hence not being robust to communication failure and opened the possibility of overconfidence.

23.3.3 Alternatives for Probabilistic Representation

This section will explore several possible avenues for a suitable representation and provide a selection based on algorithmic efficiency, storage requirements, and use of approximations. An assessment will be made of exponential functions, kernel methods, and Monte Carlo methods to determine what prospects exist for the use of these representations in decentralized data fusion applications.

Any number of representations may be chosen to perform DDF. However, the majority of operations required to perform DDF with these representations are not closed in form, and many require inefficient numerical approximations. This renders these representations impractical for real-world applications, with real-time constraints. With the advent of GPU-based numerical processing, this is becoming less of a concern.

Monte Carlo methods may be employed as described by Ong (2007) and Ong et al. (2005) who provide a workable solution by converting the particles to a continuous distribution using a variety of reparameterization methods. After channel filter operations are complete, the distribution is then resampled as particles.

23.3.3.1 Gaussian Representation

The baseline implementation of DDF would imply Gaussian distributions and Kalman filter-based techniques.

This is often in the form of the information filter form of the Kalman filter as described by Nettleton and Durrant-Whyte (2001). This simplifies channel updates and local updates at the expense of prediction cycles.

When the information filter is applied to a decentralized data fusion problem, the result is the decentralized information filter, defined as follows:

$$\hat{\mathbf{y}}(k|k) = \hat{\mathbf{y}}(k|k-1) + \sum_{j=1}^n \Delta \mathbf{y}_j(k) \quad (23.4)$$

$$\mathbf{Y}(k|k) = \mathbf{Y}(k|k-1) + \sum_{j=1}^n \Delta \mathbf{Y}_j(k) \quad (23.5)$$

where j is the number of nodes providing information at time k .

When considering channel updates, information held within the channel filters may be conveniently and efficiently subtracted from the information received from another node.

When considering the choice of using the Kalman filter or its information form, one would need to determine the frequency of channel updates (a function of the number of connected nodes) versus frequency of local state predictions to determine the suitability for a particular purpose.

23.3.3.2 Monte Carlo Methods

The use of particle filters in robotic applications has been successfully employed for navigation by Gustafsson et al. (2002) and many other applications. However,

the multiagent distribution of a particle sets has remained relatively limited. The concept of the assimilation of multiple independent particle sets is described by Thrun et al. (1999). However, it quickly becomes apparent that costly numerical approximations are required to perform a Bayesian update with two independent particle sets. This is due to the fact that no two samples from each set are identical, thus making it impossible to directly obtain a product.

Ong (2007) and Ong et al. (2005) provide a workable solution by converting the particles to a continuous distribution using a variety of reparameterization methods. After channel filter operations are complete, the distribution is then resampled as particles.

23.3.3.3 Gaussian Mixtures

Gaussian mixtures provide a convenient means to represent a multimodal probability distribution. The product of two Gaussian mixtures is indeed another Gaussian mixture. However, the combinatorial expansion results in a dramatic increase of the quantity of Gaussians. Use of Gaussian mixtures for Bayesian filtering is closed in form, but not closed in its dimensionality. Practical application of Gaussian mixtures is demonstrated in both recent and past works of Alspach and Sorenson (1972), Ito and Xiong (2000), and Robert (1996). The resampling of the ever-growing Gaussian mixtures may be achieved by a variety of methods such as EM (expectation maximization), importance sampling (as used by particle filters), and clustering using distance measures between Gaussians.

The assimilation of Gaussian mixtures within a DDF network can be achieved using an extension of the Gaussian covariance intersect method described by Upcroft et al. (2005). However, there is again a multiplicative increase in the number of components of the resulting distribution. The number of components must be reduced with one of the aforementioned methods. The covariance intersect method is also somewhat conservative in nature and provides a far from optimal Bayesian update.

23.3.3.4 Parzen Density Estimates

An attractive alternative for approximating a probability distribution is a Gaussian mixture consisting of the same-shaped Gaussian kernel.

Probability distributions may be represented using a variety of kernel-based methods. One such method is the Parzen density estimate (Parzen 1962). Since the kernel is Gaussian, it is a most useful implementation due to the fact that a significant number of operations are closed in form and are computationally less complicated than full Gaussian mixtures (Ridley 2005).

23.3.4 Storage Comparison

Consider the communication storage requirements of a probabilistic representation in N dimensions, with a possible M components. For the case of Gaussian distributions, Gaussian mixtures, and the Gaussian kernel of Parzen density estimates, the covariance matrix is symmetric positive definite (SPD), and as such communication

Table 23.1 Storage requirements of various probabilistic representations in N dimensions with M possible components

| Name | Samples | Kernels | Weights | Total storage |
|--------------------|---------|---------|---------|---------------------------|
| Gaussian | 1 | 1 | 0 | $(N^2 + N)/2 + N$ |
| Parzen estimate | M | 1 | M | $(N^2 + N)/2 + M(N + 1)$ |
| Weighted particles | M | 0 | M | $M(N + 1)$ |
| Particles | M | 0 | 0 | $N \times M$ |
| Gaussian mixture | M | M | M | $(N^2 + N)M/2 + M(N + 1)$ |

does not require the storage of all $N \times N$ terms. Each covariance matrix only requires the storage of $(N^2 + N)/2$ values. Table 23.1 shows the required storage in units of the floating-point values (typically 64-bit floating-point numbers).

Both Gaussian mixtures and Parzen estimates perform well. In terms of storage, Gaussian mixtures would be an obvious choice. However, when the overall computational complexity is taken into consideration, the benefits of a Parzen density estimate are significant and may be a better choice when computational resources are at a premium.

23.4 Gaussian Mixture-Based Decentralized Data Fusion

Gaussian mixture model-based decentralized data fusion is chosen for the later example due to its compact storage requirements and ability to provide a good approximation to a variety of distributions. The desire to provide capability for the characterization of natural features and the common use of GMM in machine learning problems made it difficult to overlook.

23.4.1 Introduction

Mixtures of Gaussians, otherwise known as “sums of Gaussians” (SOG) or “Gaussian mixture models” (GMM), are often used in order to represent a probability distribution that has multiple modes. A mixture of Gaussians i , each with weights γ_i , is defined as:

$$P(\mathbf{x}) = \sum_{i=1}^n \gamma_i G_i(\mathbf{x}) \quad (23.6)$$

where

$$G_i(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2} [\mathbf{x} - \bar{\mathbf{x}}]^T \Sigma_i^{-1} [\mathbf{x} - \bar{\mathbf{x}}]} \quad (23.7)$$

and

$$\sum_{i=1}^n \gamma_i = 1 \quad (23.8)$$

Motivation for the use of GMMs:

- A compact representation
- Good approximation to a variety of distributions
- Well-established methods for learning models

Bayesian estimation with Gaussian mixtures has been well established with early work by Sorenson and Alspach (1971) and Alspach and Sorenson (1972); the development and implementation of DDF with such distributions was therefore a novel contribution to this field.

23.4.2 Local State Estimation

Maintaining a local estimate with Gaussian mixtures requires substantial infrastructure for sensor likelihood models, data association, and component reduction. This subsection describes the more challenging of these aspects.

23.4.2.1 Observations

A Gaussian mixture model (GMM) was used to fit the observations, and thus an approximate transformation of the likelihood from the bearing-only measurement space to a sensor-centric Cartesian coordinate space was required (Upcroft et al. 2005). In the experiments, the approximation for the GMM was learned off-line using trust region optimization (Fig. 23.3). The initial parameters for the optimization were equally weighted Gaussians spread evenly over a range of 300 m on the x-axis. Approximately 20 iterations resulted in a good fit to the distribution. The learned model was then rotated and translated appropriately for each specific observations.

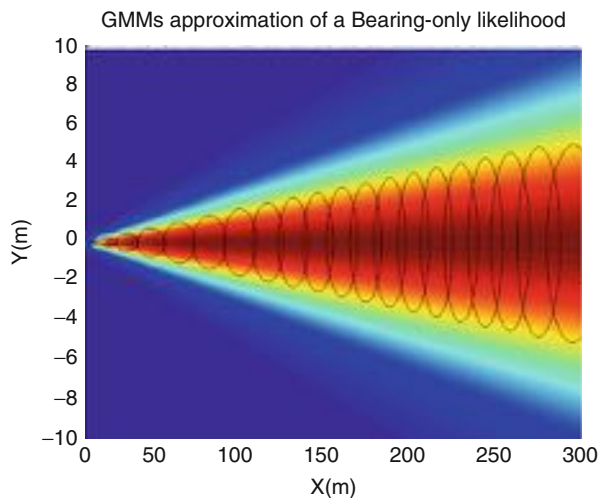


Fig. 23.3 Bearing-only likelihood, learned off-line using 20 components

23.4.2.2 Local Filter

Local state estimation proceeded using a Bayesian filter (Sorenson and Alspach 1971; Alspach and Sorenson 1972; Alspach 1975; Upcroft et al. 2005). Substitution of GMMs into Bayesian theorem (Eq. 23.1) gives

$$p(\mathbf{x}_k | \mathbf{z}_k) = A \sum_{i=1}^M \pi_{zi} \mathcal{N}_{zi} \sum_{j=1}^N \pi_{xj} \mathcal{N}_{xj} \quad (23.9)$$

where $A = 1/p(\mathbf{z}_k | \mathbf{z}_{k-1})$ is a normalizing constant, the \mathcal{N}_z 's represent the likelihood distribution $p(\mathbf{z}_k | \mathbf{x}_k)$, and the \mathcal{N}_x 's represent the prediction $p(\mathbf{x}_k | \mathbf{z}_{k-1})$, similarly for π_z and π_x .

Expanding Eq. 23.9 results in $M \times N$ terms, each which involves a multiplication of two weighted Gaussians. Thus, the posterior distribution is represented by $M \times N$ weighted Gaussians.

GMMs also allow an analytical solution to the Chapman-Kolmogorov equation. Substituting GMMs into Eq. 23.2 results in a convolution between $M \times N$ weighted Gaussians with each term a Gaussian of the form (Muirhead 1982)

$$\pi \mathcal{N}(\mu_a + \mu_b, \Sigma_a^2 + \Sigma_b^2) \quad (23.10)$$

where the subscripts denote the variables for the two Gaussians and π is a constant weighting term.

23.4.2.3 Component Reduction

Figure 23.4 shows that when an update is performed, it will result in a multiplicative increase in parameters. A component-merging technique is then required to keep the update operations tractable. As many of the parameters are similar or bear little contribution to the overall distribution, it is possible to reduce the number of components without modifying the probability distribution function significantly.

While the number of components should be kept to a minimum, the overall shape of the distribution should remain the same. To achieve this, a joining algorithm is applied repeatedly until the desired level of component reduction has been achieved (West 1993; Salmond et al. 1998; Salmond 1989). The distance measure for gauging the similarity between component i and component j of the GMM is a Mahalanobis-type distance measure:

$$d_{ij}^2 = \frac{\gamma_i \gamma_j}{\gamma_i + \gamma_j} (\mu_i - \mu_j)^T \Sigma^{-1} (\mu_i - \mu_j) \quad (23.11)$$

where μ is the component state vector, Σ is the mixture covariance matrix, and γ is the component weight.

The parameters of the component that results from combining component i with the component most similar to it (component j) are:

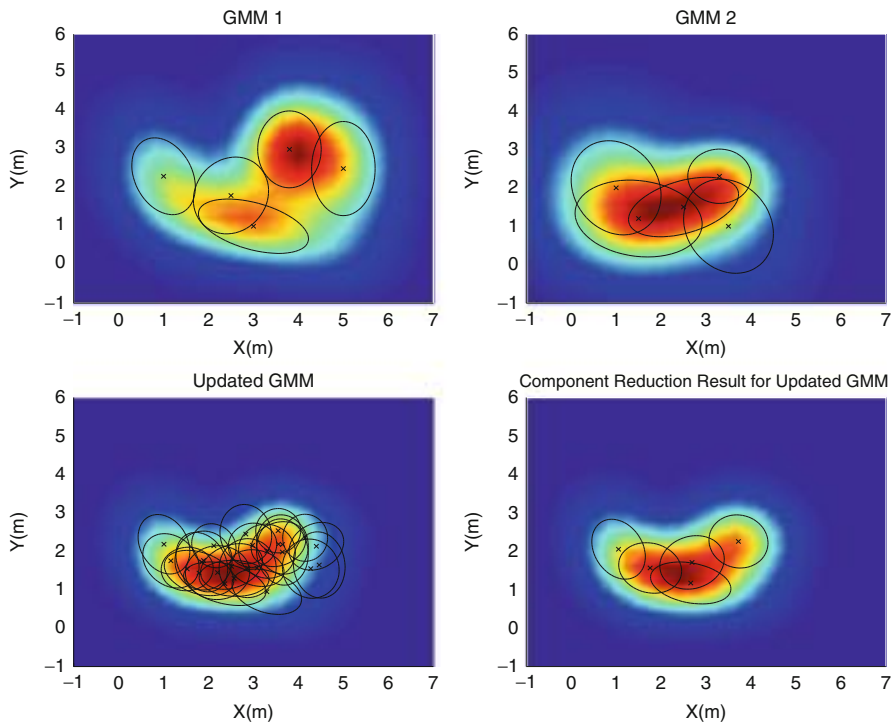


Fig. 23.4 The first Gaussian mixture (GMM 1) consists of five components. If updated with the second Gaussian mixture (GMM 2) which also consists of five components, the multiplicative expansion of parameters occurs. The updated Gaussian mixture (*bottom left*) consists of 25 components. As many of the components are similar, the Gaussian mixture may be reduced to a lesser number of components without modifying the probability distribution significantly. The figure on the *bottom right* shows that the distribution is preserved when using West's joining algorithm to merge the components to the original number of five

$$\text{Weight} : \gamma_{ij} = \gamma_i + \gamma_j \quad (23.12)$$

$$\text{Mean} : \mu_{ij} = \frac{1}{\gamma_i + \gamma_j} \{ \gamma_i \mu_i + \gamma_j \mu_j \} \quad (23.13)$$

$$\begin{aligned} \text{Covariance} : \Sigma_{ij} = \frac{1}{\gamma_i + \gamma_j} \{ \gamma_i \Sigma_i + \gamma_j \Sigma_j + \frac{\gamma_i \gamma_j}{\gamma_i + \gamma_j} \\ (\mu_i - \mu_j)(\mu_i - \mu_j)^T \} \end{aligned} \quad (23.14)$$

23.4.3 Node-Node Fusion

Analytical solutions to the division required for consistent channel filter-based fusion using GMMs are unknown. As will be described in Sect. 23.4.3.1, a

nonoptimal solution for node-to-node fusion of Gaussian representations is the covariance intersect (CI) filter which conservatively combines the information in two incoming channels, assuming that the correlation is unknown. In addition, a variant of the covariance intersect algorithm (Upcroft et al. 2005) is applied to Gaussian mixtures. However, as in the work of Ihler et al. (2004) and Rosencrantz et al. (2003a), divergent solutions are still possible.

23.4.3.1 Covariance Intersect

Primary use of covariance intersect algorithm is for the initialization of channel filters for situations where the local filter and channel contain prior information of unknown correlation.

The covariance intersection (CI), developed by Julier and Uhlmann (2001b), is a method of data fusion that provides a robust method for fusing data from two sources with unknown correlation.

During implementation, the conditional independence of subsequent observations from a single platform resulted in overconfident and biased solutions. The solution was to employ covariance intersect on local observations and full channel filter updates between platforms as independence of observations was assured.

Consider two estimates μ_a and μ_b with covariances Σ_a and Σ_b , respectively. The CI algorithm computes an updated covariance matrix as a convex combination of the two initial covariance matrices in the form

$$\Sigma_c^{-1} = \omega \Sigma_a^{-1} + (1 - \omega) \Sigma_b^{-1} \quad (23.15)$$

$$\Sigma_c^{-1} \mu_c = \omega \Sigma_a^{-1} \mu_a + (1 - \omega) \Sigma_b^{-1} \mu_b \quad (23.16)$$

where $0 \leq \omega \leq 1$ with ω computed so as to minimize a chosen measure for the size of the covariance matrix.

The resultant estimate is based on all possible correlations and thus removes the need for the division in Eq. 23.3.

While this approach to data fusion is conservative, Nettleton (2003) demonstrated its use for the initialization of decentralized filters when a single node previously in contact with a group of other nodes reestablishes contact. The common information between the nodes is unknown, and immediate use of the information filter update could potentially result in the double counting of this information, resulting in overconfidence. The CI algorithm makes it possible to make use of some of this information while still ensuring no double counting of information.

23.4.3.2 Pairwise Component Covariance Intersect

This variant of the CI algorithm involves a pairwise CI update between each of the Gaussian components in the two mixtures that are to be fused. The weight update for each component is given by

$$\pi_c = \alpha \pi_a \pi_b \quad (23.17)$$

where

$$\alpha = \frac{1}{(2\pi)^{D/2} |\Sigma_\omega|^{1/2}} e^{-1/2(\mu_a - \mu_b)^T \Sigma_\omega^{-1} (\mu_a - \mu_b)} \quad (23.18)$$

is the scaling constant resulting from the multiplication of two Gaussians, D is the dimension of the space, and $\Sigma_\omega = \Sigma_a/\omega + \Sigma_b/(1 - \omega)$.

The update has been found to remain non-divergent for all practical scenarios encountered and is always better than a straight multiplication in which the common information is not accounted for at all (Upcroft et al. 2005). Although this conservative behavior is not guaranteed, there is hope that future work will obtain convergence bounds for this method.

Recently, Julier (2006) has proposed a similar fusion update which behaves more consistently but again does not guarantee that consistency is maintained.

23.5 Experimental Hardware and Algorithms

In this chapter, the DDF algorithms are demonstrated through playback of data obtained from multiple UAVs and online between heterogeneous platforms including a ground vehicle, a UAV, and human operators. The following section describes the physical hardware used to gather the data, as well as some specific algorithm detail employed to process it.

23.5.1 Flight Platform

Data acquired during the ANSER project (University of Sydney) was gathered using the Brumby MKIII air vehicles as shown in Fig. 23.5. Figure 23.6 shows the vision sensors used in the experiments.

Both platforms were fitted with an INS/GPS subsystem that provides a position in a global coordinate frame as well as the corresponding Euler angles: roll ϕ ,



Fig. 23.5 Brumby Mk III flight vehicle

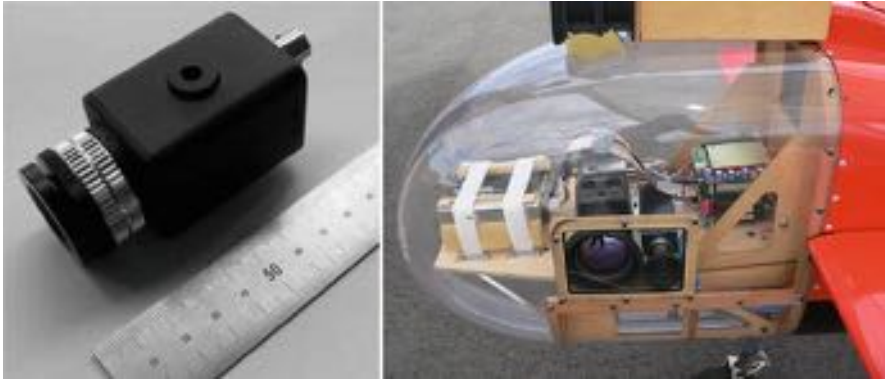


Fig. 23.6 Brumby sensors and payloads: the visual sensors used to gather data experiment

pitch θ , and yaw ψ . The GPS receiver was used to synchronize the local clocks of all onboard payloads, thus assuring accurate temporal registration of the data sets.

23.5.2 Sensor Model and Transforms

The camera was calibrated with well-established methods described in Heikkilä and Silvén (1997) with a readily available toolbox. The calibration procedure provides the principal point (u_0, v_0) and focal lengths f_u, f_v (in pixels) for each axis and the expected error in the pinhole model. Application of this model via the camera matrix (Hartley and Zisserman 2000) projects the feature coordinates (in the sensor reference frame) to homogenous coordinates:

$$\begin{bmatrix} u' \\ v' \\ s \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ z \\ x \end{bmatrix} \quad (23.19)$$

From this, a direct mapping from sensor coordinates (x, y, z) to image coordinates (u, v) can be defined:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{y f_u}{x} + u_0 \\ \frac{z f_v}{x} + v_0 \end{bmatrix} \quad (23.20)$$

The inverse mapping cannot recover the loss of range information. However, a direction to the feature can be recovered. It may be more convenient to define this information in terms of the Euler angles ψ_f^s, θ_f^s , similar to the current vehicle state:

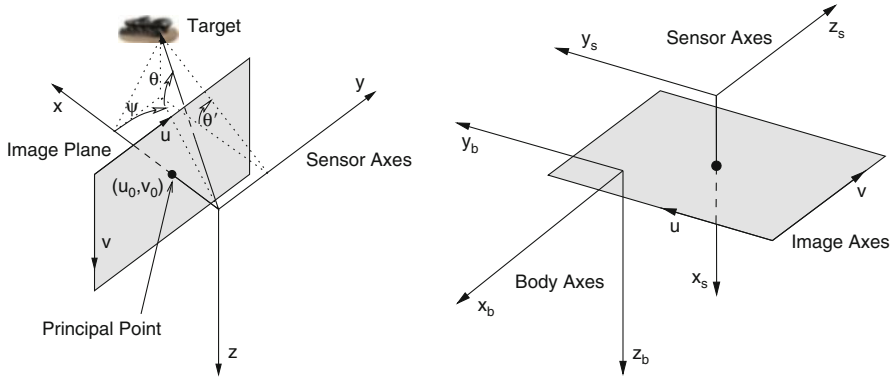


Fig. 23.7 Principal point, image, and sensor axes (*left*). Image, sensor, and body axes (*right*)

$$\psi_f^s = \arctan((u - u_0)/f_u) \quad (23.21)$$

$$\theta' = -\arctan((v - v_0)/f_v) \quad (23.22)$$

$$\theta_f^s = \arctan(\tan(\theta') \cos(\psi)) \quad (23.23)$$

Figure 23.7 provides a summary of the image, sensor, and platform geometry.

23.5.3 Extraction of Features

The flight test facility conveniently had artificial features placed under the vehicle flight path. The features were 0.9×0.9 m square plastic sheets that appeared as distinctive white points in the image. These features were able to be reliably extracted by a multi-pass algorithm progressively reducing possible candidates within the image. Objects that satisfied the criteria of intensity, size, and shape of the artificial features were reliably extracted.

Some range information can be recovered due to the fact that the features were of a known size. Utilizing the bearing equations and assuming the feature is parallel to the image plane, the height, width, and corresponding area (in pixels), A_i of an object can be defined, where A_f is the feature area in meters:

$$\begin{aligned} h_u &= \frac{h_y f_u}{x} \\ w_v &= \frac{h_z f_v}{x} \\ A_i &\approx \frac{f_u f_v A_f}{x^2} \end{aligned} \quad (23.24)$$

Rearranging Eq. 23.24 provides an expression for an estimate of the perpendicular height (x) above the feature. The area being expressed in terms of effective side length of the feature in pixels (p):

$$x = \sqrt{\frac{f_u f_v A_f}{p^2}} \quad (23.25)$$

The range estimate is then obtained by correcting the height estimate with the direction of the target:

$$r_f^s = \frac{x}{\cos(\theta) \cos(\psi)} \quad (23.26)$$

Differentiating x with respect to p provides an expression for the error in the range estimate, where Σ_p^2 is the variance in the pixels:

$$\begin{aligned} \Sigma_x &= \frac{f_u f_v A_f}{x p^3} \Sigma_p \\ \Sigma_r &= \frac{\Sigma_x}{\cos(\theta) \cos(\psi)} \end{aligned} \quad (23.27)$$

23.5.4 Observation Transformations

The previous sections only describe the extraction of a feature from an image, which provides an observation in the sensor coordinate frame. The sensor is mounted on a platform, with its own reference frame. The platform is localized within the global reference frame provided by the INS/GPS subsystem. This requires both rotation and translation of the observation multiple times to provide observation information in the global reference frame.

The cosine rotation matrix \mathbf{C}_i^j relating frame i to frame j with yaw ψ , pitch θ , and roll ϕ is used frequently when transforming between reference frames.

Three positions are required:

- \mathbf{X}_f^s : Position of feature relative to sensor coordinate frame
- \mathbf{X}_s^p : Position of sensor relative to platform coordinate frame
- \mathbf{X}_b^g : Position of body relative to global coordinate frame

Also, three different rotations are required:

- \mathbf{C}_f^s : Cosine matrix of feature relative to sensor coordinate frame
- \mathbf{C}_s^p : Cosine matrix of sensor relative to platform coordinate frame
- \mathbf{C}_b^g : Cosine matrix of body relative to global coordinate frame

The position of the feature relative to the sensor is

$$\mathbf{X}_f^s = \begin{bmatrix} r_f^s \cos \theta_f^s \cos \psi_f^s \\ r_f^s \cos \theta_f^s \sin \psi_f^s \\ r_f^s \sin \theta_f^s \end{bmatrix} \quad (23.28)$$

with covariance matrix

$$\mathbf{R}_f^s = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & r^2 \sigma_\psi^2 & 0 \\ 0 & 0 & r^2 \sigma_\theta^2 \end{bmatrix} \quad (23.29)$$

The full transformation of the feature position to the global coordinate frame is

$$\mathbf{X}_f^g = \mathbf{X}_b^g + \mathbf{C}_b^g \mathbf{X}_s^b + \mathbf{C}_b^g \mathbf{C}_s^b \mathbf{X}_f^s \quad (23.30)$$

The feature covariance in the global frame is

$$\mathbf{R}_f^p = [\mathbf{C}_p^g \mathbf{C}_s^p \mathbf{C}_f^s] \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & r^2 \sigma_\psi^2 & 0 \\ 0 & 0 & r^2 \sigma_\theta^2 \end{bmatrix} [\mathbf{C}_p^g \mathbf{C}_s^p \mathbf{C}_f^s]^T \quad (23.31)$$

However, platform localization will normally contain uncertainties in roll, pitch, and yaw. To properly compensate for this, a more elaborate transformation is required. Either the angular uncertainties may be factored into the sensor observations, or the Jacobian of the observation function with respect to all platform states $\nabla g(\mathbf{P}_f^b, \psi_p^g, \theta_p^g, \phi_p^g)$ will have to be evaluated. The result being

$$\mathbf{R}_f^p = \nabla g(\mathbf{P}_f^b, \psi_p^g, \theta_p^g, \phi_p^g) \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\psi^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\phi^2 \end{bmatrix} \nabla g(\mathbf{P}_f^b, \psi_p^g, \theta_p^g, \phi_p^g)^T \quad (23.32)$$

When using Gaussian mixtures, this method can be applied in turn to each component of the distribution. Alternatively, the unscented filter (Julier and Uhlmann 2001a) is an equally effective method to propagate the uncertainties associated with the platform localization system and can equally be applied in a Gaussian or Gaussian mixture-based system. The complexity in obtaining and evaluating the Jacobian may be bypassed.

23.5.5 Process Model

As the features are identical to that in the results obtained by Nettleton (2003) for the ANSER project, the process model remains unchanged as the integrated Ornstein-Uhlenbeck (IOU) process. This process has a bounded velocity by appropriate choice of the model parameter γ . The discrete-time system model is

$$\mathbf{x}(k) = \mathbf{F}_k \hat{\mathbf{x}}(k-1 | k-1) + \mathbf{w}_k \quad (23.33)$$

where the state vector is

$$\mathbf{x}(k) = [x, \dot{x}, y, \dot{y}, z]^T \quad (23.34)$$

The vertical dimension is modeled with position only, as the feature is assumed to remain on the ground at all times, and local flatness of the environment is assumed. The state transition matrix for this system is

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 \\ 0 & F_v & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 \\ 0 & 0 & 0 & F_v & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (23.35)$$

where

$$F_v = e^{-\Delta T \gamma} \quad (23.36)$$

The process noise is $\mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T$ where

$$\mathbf{Q}_{(k,k-1)} = \begin{bmatrix} q_x & 0 & 0 \\ 0 & q_y & 0 \\ 0 & 0 & q_z \end{bmatrix} \quad (23.37)$$

and

$$\mathbf{G}_k = \begin{bmatrix} \Delta T^2 & 0 & 0 \\ \frac{1}{\gamma}(1 - F_v) & 0 & 0 \\ 0 & \Delta T^2 & 0 \\ 0 & \frac{1}{\gamma}(1 - F_v) & 0 \\ 0 & 0 & \Delta T^2/2 \end{bmatrix} \quad (23.38)$$

While the features are stationary in this experiment, the model is appropriate for tracking moving features as well. The model is also an excellent choice for situations where features remain unobserved for large periods. A constant velocity model is inappropriate, as the target estimate will continue to drift in the last observed direction, until reobserved. This will make successful data association difficult to achieve. By bounding the velocity, the feature uncertainty will better encapsulate the variety of maneuvers a feature may make.

23.5.6 Feature Maintenance

The tracking of multiple targets suffers from data association problems caused by spurious observations and features that have too great an uncertainty. This implementation utilizes the Bhattacharyya coefficient as defined by Kailath (1967) for data association. It was chosen as its use for distributions other than Gaussian is straightforward and will allow for future comparisons. The feature was known to be well separated; thus, features that are associated with other features were culled.

Features with entropies greater than an empirical threshold were also culled. This kept the number of tracked features quite stable and ensured all features updated regularly due to reliable data association. More elaborate multi-hypothesis tracking management may be employed where the implementor deems necessary, but for the purposes of this experiment, this approach was sufficient.

23.6 Example Implementation with Gaussian and GMM

This section shows example results from a real-data simulation of two sets of two DDF nodes, one set operating with the Gaussian representation, the other with the Gaussian mixture representation.

The aim of this example is to show with actual data the feasibility of performing decentralized data fusion utilizing GMM density estimates. The quality and constancy of the solution will be compared at all nodes to a reference node that has a centralized solution.

Both simulations were run concurrently; both GMM and Gaussian likelihoods are generated from the same sensor observation. This makes a direct comparison more effective as each observation has the same time stamp, and both filters on each node run with the same time-base. This reduces any inconsistencies arising from variations between separate simulations.

23.6.1 Data Set

Data sets from two previous flights were used. Each set contains about 40,000 gray-scale video frames logged at 50 Hz at a resolution of 384×288 pixels, along with a corresponding GPS/INS navigation solution. These two data sets were played back concurrently to represent two flight vehicles searching an area at the same time. Figures 23.8 and 23.9 show the flight paths, artificial feature locations, and the raw observations allocated to each simulated platform. Sample video frames are shown in Fig. 23.10.

23.6.2 Platform and Component Topology

The component topology includes a parallel set of nodes operating with the same sensor components as shown in Fig. 23.11. There are two nodes operating with a DDF link and a separate node receiving all observations to provide a centralized solution with which to provide a comparison.

23.6.3 Comparison to Gaussian Representation

As both the Gaussian and GMM DDF nodes were run in parallel, it is straightforward to compare the difference in performance of the algorithms. Identical

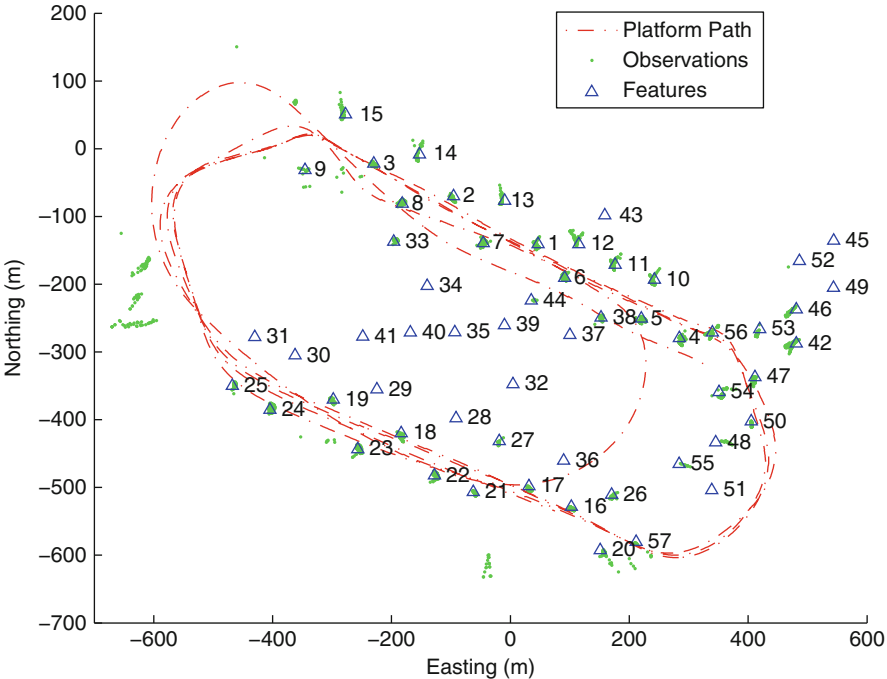


Fig. 23.8 Flight path and raw observations of flight vehicle 0

camera observations were fed into two separate likelihood generators. The Gaussian likelihood generator makes use of a range estimate, whereas the GMM likelihood generator utilized 20 components numerically optimized off-line. A single feature was chosen to compare the different methods.

23.6.4 Sample Feature Results

Figure 23.12 shows the absolute errors for both Gaussian and GMM nodes. The different variants arrive at similar solutions. As would be expected, the GMM versions, not having range information, take longer to converge to their solution. The individual nodes are also less coherent, as the pairwise covariance intersect method is more conservative in its approach.

Figure 23.13 shows the entropy of the Gaussian and GMM variants. The entropy provides a measure of the certainty of a particular estimate. The GMM variants tend to have greater entropy. This would be expected as no explicit range information is used, and multimodal GMM distributions would always tend to have a greater entropy than that of a unimodal Gaussian (except in cases when the distribution collapses to a single Gaussian).

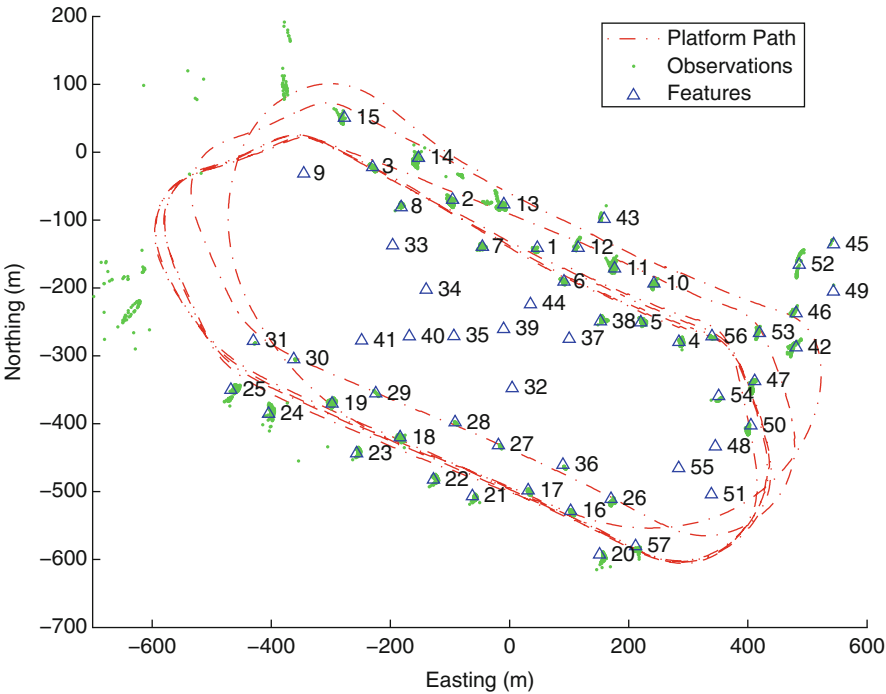


Fig. 23.9 Flight path and raw observations of flight vehicle 1

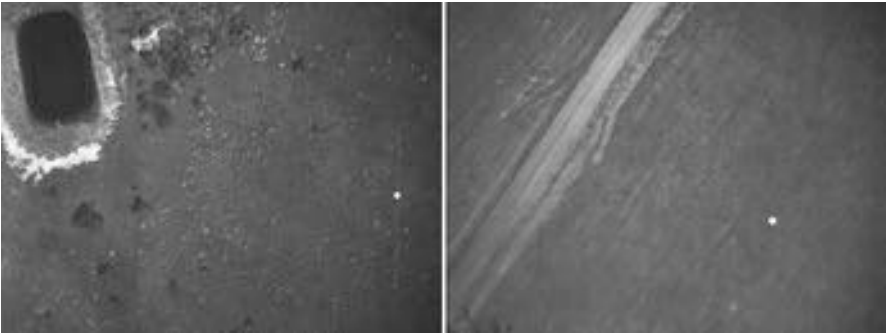


Fig. 23.10 Sample logged video data, showing features such as roads, dams, and artificial features (*bright white*)

23.6.5 Discussion

The error of the GMM solution is of similar magnitude to that of Gaussian version and is less variable. The Gaussian implementation was able to make use of the range estimate during image processing. However, this range estimate may be biased

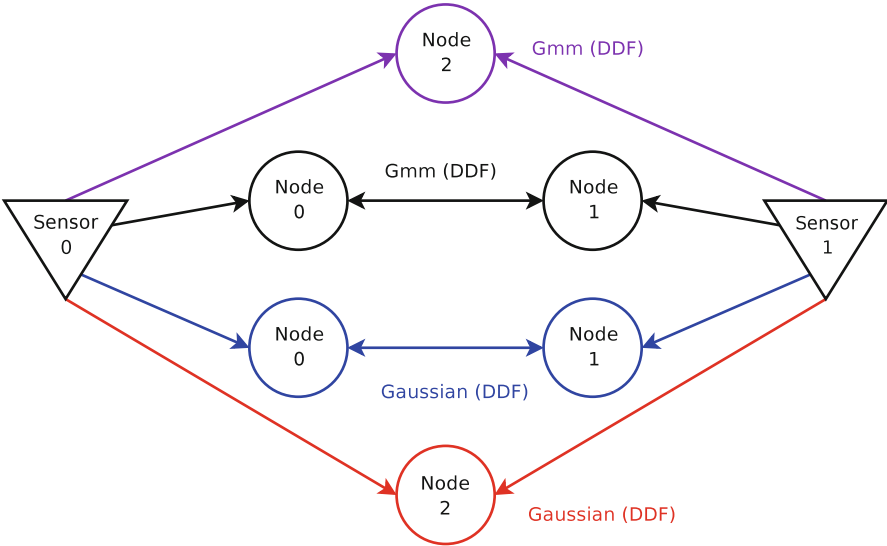


Fig. 23.11 Communication topology of sensors and parallel nodes

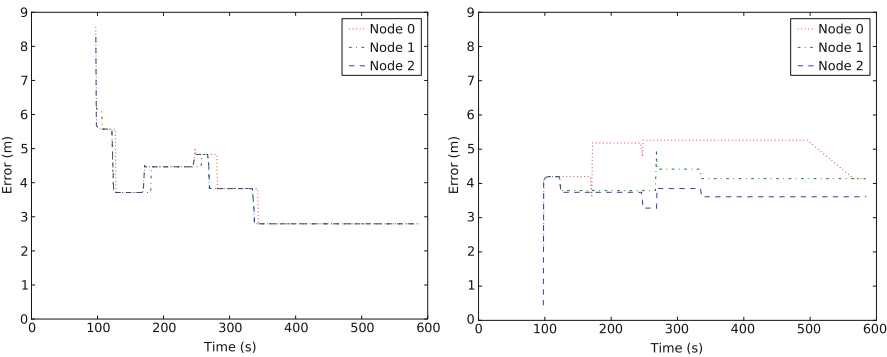


Fig. 23.12 GMM and Gaussian DDF comparison. Absolute sample feature error at all nodes. Gaussian (*left*) and GMM (*right*)

due to variability in any given target. The GMM version is a more representative example of a true bearing-only sensor and made no use of no range information. It therefore had to depend on the accuracy of the fitted likelihood function and performed well.

The GMM DDF nodes tended to have differing solutions to each other; this is due to the heuristic applied to prevent a covariance intersect update from occurring when the information gain is minimal from channel data. Performing repeated update would only serve to increase overconfidence in the solution as the reduction process following the covariance intersect operation tends to cause a collapse of the distribution. Conversely the Gaussian nodes have more consistent

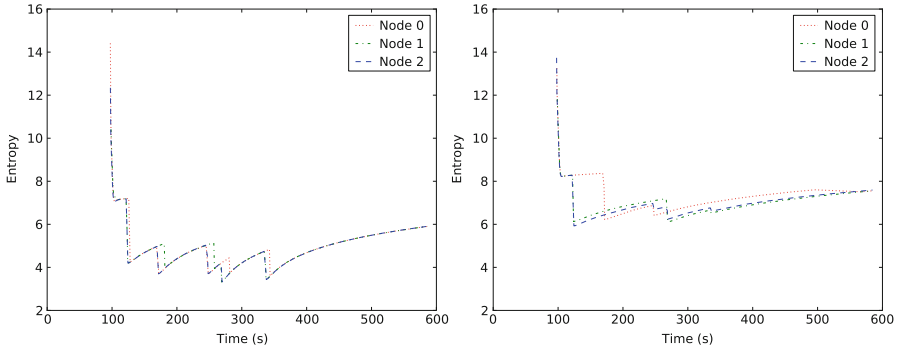


Fig. 23.13 GMM and Gaussian DDF comparison. Sample feature entropy of all nodes Gaussian (*left*), GMM (*Right*)

solutions; they only differ for the periods where information from new sensor observations have yet to propagate to the other node not directly connected to the sensor.

As expected, the entropy of the GMM solutions is equal to or greater than the Gaussian variants. This is to be expected due to the nature of multimodal distributions and the more conservative PCCI fusion algorithm.

23.7 Conclusion

The design of robust multi-platform UAV-based search and rescue or location/tracking systems would benefit from a decentralized approach due to the higher failure likelihood of UAV-based systems. The benefits of a decentralized data fusion approach for such a problem have been highlighted, as well as a high-level functional description of the system components.

The various options available to a systems designer for implementing a non-Gaussian UAV-based tracking/search and rescue system are numerous. This chapter has provided a summary of the available options and the benefits and difficulties that may arise from each option.

Kalman filter-based approaches to decentralized data fusion have been long established. More specific methods for the implementation of Gaussian mixture model-based DDF system have been outlined. Details are provided for the implementation of a system that employs a vision-based sensor to track an infrequently observed erratically moving object.

It has been shown, through example, that it is possible to implement a decentralized data fusion system using single Gaussian or mixtures of Gaussians. The use of Gaussian mixtures introduces many challenges, but none that are computationally intractable or otherwise infeasible to implement.

References

- D. Alspach, A Gaussian sum approach to the multi-target identification-tracking problem. *Automatica* **11**, 285 (1975)
- D.L. Alspach, H.W. Sorenson, Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. Autom. Control* **17**(4), 439–448 (1972)
- Y. Bar-Shalom (ed.), *Multitarget-Multisensor Tracking: Advanced Applications* (Artech House, Boston, 1990)
- Y. Bar-Shalom, X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques* (Yaakov Bar-Shalom, Storrs, 1995)
- F. Bourgalt, H. Durrant-Whyte, Communication in general decentralized filters and the coordinated search strategy, in *Proceedings of The 7th International Conference on Information Fusion*, Stockholm, Sweden, 2004
- F. Bourgault, T. Furukawa, H.F. Durrant-Whyte, Coordinated decentralized search for a lost target in a Bayesian world, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, Las Vegas, NV, USA, 2003, pp. 48–53
- A. Brooks, A. Makarenko, T. Kaupp, S. Williams, H. Durrant-Whyte, Implementation of an indoor active sensor network, in *Proceedings of the 9th International Symposium on Experimental Robotics*, Singapore, 2004
- S. Challa, M. Palaniswami, A. Shilton, Distributed data fusion using support vector machines, in *Proceedings of the 5th International Conference on Information Fusion*, Annapolis, MD, USA 2002, pp. 881–885
- N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F* **140**, 107–113 (1993)
- S.H. Grime, Communication in decentralised sensing architectures. Ph.D. thesis, The University of Oxford, 1992
- S. Grime, H.F. Durrant-Whyte, Data fusion in decentralized sensor networks, in *Control Engineering Practice*, vol. 2(5) (Pergamon Press Ltd, Oxford, 1994), pp. 849–863
- B. Grocholsky, Information-theoretic control of multiple sensor platforms. Ph.D. thesis, Australian Centre for Field Robotics, The University of Sydney, 2002
- F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P. Nordlund, Particle filters for positioning, navigation and tracking. *IEEE Trans. Signal Process.* **50**, 425–437 (2002)
- R.I. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision* (Cambridge University Press, Cambridge/New York, 2000)
- H. Hashemipour, S. Roy, A. Laub, Decentralized structures for parallel Kalman filtering. *IEEE Trans. Autom. Control* **33**, 88–93 (1988)
- J. Heikkilä, O. Silvén, A four-step camera calibration procedure with implicit image correction, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, San Juan, Puerto Rico, 1997, pp. 1106–1112
- A. Ihler, J. Fisher III, R. Moses, A. Willsky, Nonparametric belief propagation for self-calibration in sensor networks, in *Information Processing in Sensor Networks*, Berkeley, California, USA, 2004
- K. Ito, K. Xiong, Gaussian filters for nonlinear filtering problems. *IEEE Trans. Autom. Control* **45**(5), 910–927 (2000)
- S.J. Julier, An empirical study into the use of chernoff information for robust, distributed fusion of Gaussian mixture models, in *Proceedings of the 2006 FUSION Conference*, Florence, Italy, 2006
- S.J. Julier, J.K. Uhlmann, Data fusion in nonlinear systems, in *Handbook of Multisensor Data Fusion* (CRC, Boca Raton, 2001a). Chap. 13
- S.J. Julier, J.K. Uhlmann, General decentralized data fusion with covariance intersection (ci), in *Handbook of Multisensor Data Fusion* (CRC, Boca Raton, 2001b). Chap. 12

- T. Kailath, The divergence and bhattacharyya distance measures in signal selection. *IEEE Trans. Commun. Technol.* **15**(1), 52–60 (1967)
- J. Manyika, H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach* (Ellis Horwood, New York, 1994)
- R. Muirhead, *Aspects of Multivariate Statistical Theory* (Wiley, New York, 1982)
- E. Nettleton, Decentralised architectures for tracking and navigation with multiple flight vehicles. Ph.D. thesis, Department of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, 2003
- E. Nettleton, H. Durrant-Whyte, Delayed and asequent data in decentralised sensing networks, in *Sensor Fusion and Decentralised Control in Robotic Systems IV*, vol. 4571, ed. by G.T. McKee, P.S. Schenker (Society of Photo-Optical Instrumentation Engineers, Bellingham, 2001), pp. 1–9
- E. Nettleton, P. Gibbens, H. Durrant-Whyte, Closed form solutions to the multiple platform simultaneous localisation and map building (slam) problem, in *Sensor Fusion: Architectures, Algorithms, and Applications IV*, (Society of Photo-Optical Instrumentation Engineers, Bellingham, 2000), pp. 428–437
- H Nii, “Blackboard Systems” *AI Magazine*, vol. 7, no. 2, 1986, pp. 38–53 and no. 3, pp. 82–106.
- L.L. Ong, Non-Gaussian representations for decentralised Bayesian estimation. Ph.D. thesis, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, 2007
- L.L. Ong, M. Ridley, B. Upcroft, S. Kumar, T. Bailey, S. Sukkarieh, H. Durrant-Whyte, A comparison of probabilistic representations for decentralised data fusion, in *Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference*, Melbourne, Australia, 2005, pp. 187–192
- E. Parzen, On estimation of a probability density function and mode. *Comput. Stat. Data Anal.* **3**, 1065 (1962)
- M.A. Paskin, C.E. Guestrin, Robust probabilistic inference in distributed systems, in *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI-04)*, Banff, Canada, 2004
- M.F. Ridley, Towards generalised decentralised Bayesian estimation. Ph.D. thesis, Department of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, 2005
- M.F. Ridley, E. Nettleton, S. Sukkarieh, H. Durrant-Whyte, Tracking in decentralised air-ground sensing networks, in *Proceedings of the 5th International Conference on Information Fusion*, Annapolis, MD, USA, 2002, pp. 616–623
- C.P. Robert, Mixtures of distributions: inference and estimation, in *Markov Chain Monte Carlo in Practice*, ed. by W. Gilks, S. Richardson, D. Spiegelhalter. Interdisciplinary Statistics (Chapman & Hall, London, 1996), pp. 441–464. Chap. 24
- M. Rosencrantz, G. Gordon, S. Thrun, Decentralized sensor fusion with distributed particle filters, in *Proceedings of the Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003a
- M. Rosencrantz, G. Gordon, S. Thrun, Locating moving entities in indoor environments with teams of mobile robots, in *AAMAS '03: Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems* (ACM, New York, 2003b), pp. 233–240
- D.J. Salmond, Mixture reduction algorithms for target tracking, in *IEE Colloquium on State Estimation in Aerospace and Tracking Applications* (IEE, London, 1989), pp. 7/1–7/4
- D.J. Salmond, D. Fisher, N.J. Gordon, Tracking in the presence of intermittent spurious objects and clutter, in *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, Orlando, Florida, 1998, pp. 460–474
- H.W. Sorenson, D.L. Alspach, Recursive Bayesian estimation using Gaussian sums. *Automatica* **7**, 465–479 (1971)
- L.D. Stone, C.A. Barlow, T.L. Corwin, *Bayesian Multiple Target Tracking* (Artech House, Boston, 1999)

- S. Thrun, J.C. Langford, D. Fox, Monte carlo hidden markov models: learning non-parametric models of partially observable stochastic processes, in *Proceedings of the 16th International Conference on Machine Learning* (Morgan Kaufmann, San Francisco, 1999), pp. 415–424
- B. Upcroft, L.L. Ong, S. Kumar, M. Ridley, T. Bailey, S. Sukkarieh, H. Durrant-Whyte, Rich probabilistic representations for bearing-only decentralised data fusion, in *IEEE Conference Proceedings on Information Fusion 2005*, Philadelphia, PA, USA, 2005
- M. West, Approximating posterior distributions by mixtures. *J. R. Stat. Soc. Ser. B* **55**(2), 409–442 (1993)

Section V

UAV Propulsion

***George J. Vachtsevanos and
Kimon P. Valavanis***

Kimón P. Valavanis and George J. Vachtsevanos

Propulsion is a critical and indispensable module of the UAV structure. Propulsion provides the necessary thrust to maintain a flight. The UAV performance, effectiveness, and utility depend strongly on the onboard propulsion capabilities. Jointly with other UAV design considerations, the propulsion system determines the endurance, size, weight, and application domain of all UAV classes.

► **Power Managements of a Hybrid Electric Propulsion System Powered by Solar Cells, Fuel Cells, and Batteries for UAVs** by Lee, Park, and Kim provides a very comprehensive description of the power management of a UAV hybrid electric propulsion system. It considers three electric propulsion systems with solar cells, fuel cells, and battery power sources, which are designed and constructed to share the same operation voltage range and connect to the power bus without the need for additional converters or controllers. MATLAB-/Simulink-based modeling is followed for each power source, and the component models are verified with published data from the manufacturers as well as flight test data. The component models are integrated into one power system model that is used to simulate the UAV electric propulsion system. For fuel cells and batteries, the simulation process is verified via comparison between simulation results and available UAV flight test results. Two types of power control logic are investigated, passive and active. The passive power management simulation shows that the behavior of each power source

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

and its integrated system are adequate for the overall UAV flight envelope. The active power management simulation yields more efficient power distribution and better system safety than passive power management. Additional simulations of a hybrid electric power system allow for estimation of the output behavior of the power source.

Power Managements of a Hybrid Electric Propulsion System Powered by Solar Cells, Fuel Cells, and Batteries for UAVs

25

Bohwa Lee, Poomin Park, and Chuntaek Kim

Contents

| | | |
|--------|--|-----|
| 25.1 | Introduction | 496 |
| 25.2 | Hybrid Electric Propulsion System | 498 |
| 25.3 | Electric Power Sources Modeling | 500 |
| 25.3.1 | Solar Cell | 500 |
| 25.3.2 | Fuel Cell | 503 |
| 25.3.3 | Battery | 506 |
| 25.4 | Simulation Modeling Verification with Flight Test Data | 506 |
| 25.5 | EAV-2 Electric Propulsion System Simulation | 512 |
| 25.5.1 | Passive Simulation | 512 |
| 25.5.2 | Active Simulation | 516 |
| 25.6 | Conclusion | 519 |
| | Nomenclature | 521 |
| | Subscripts | 523 |
| | References | 523 |

Abstract

This chapter describes the power management of a UAV hybrid electric propulsion system. Three electric propulsion systems with different power sources, i.e., solar cells, fuel cells, and batteries, are considered. The power sources are designed and constructed to share the same operation voltage range and connect to the power bus without the need for additional converters or controllers. Each power source is modeled in MATLAB/Simulink, and the component models are verified with published data from the manufacturers as well as flight test data. Furthermore, the component models are integrated into one power system. The integrated power system model is used to simulate the

B. Lee (✉) • P. Park • C. Kim

Aero Propulsion Division, Korea Aerospace Research Institute, Daejeon, South Korea

e-mail: bhlee@kari.re.kr; ppm@kari.re.kr; ctkim@kari.re.kr

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

495

DOI 10.1007/978-90-481-9707-1_115,

© Springer Science+Business Media Dordrecht 2015

electric propulsion system. For fuel cells and batteries, the simulation process is verified via comparison between the simulation results and available UAV flight test results. Subsequently, two types of power control logic are investigated: passive and active. The passive power management simulation shows that the behavior of each power source and its integrated system are adequate for the overall UAV flight envelope. In addition, the active power management simulation demonstrates that active power management yields more efficient power distribution and better system safety than passive power management. Also, power simulation of a hybrid electric power system allows for estimation of the output behavior of the power source. Therefore, it can be a valuable tool for development of a power control logic that ensures efficient power distribution.

25.1 Introduction

The mainstay of aircraft propulsion is the internal combustion engine, which takes such forms as gas turbines and reciprocating engines. However, because of soaring fuel costs and tightened control on environmental issues such as carbon dioxide emissions and noise, the search for an environment-friendly propulsion device has been an imperative research topic. Thanks to advances in electric power sources such as fuel cells, solar cells, batteries, and associated components, which yield higher performance with lighter bodies, electric propulsion is very close to achieving practical application in vehicles and, thus, in UAVs.

UAVs that employ fuel cells as their primary power source are listed in Table 25.1 (Scheppat 2004; Kellogg 2005; Crumm 2006; Anon 2007; Herwerth et al. 2007; McConnell 2007; Velez 2007; Bradley et al. 2007, 2009). The Georgia Institute of Technology demonstrated the feasibility of a small fuel-cell-powered UAV through fabrication, flight testing, and ground testing of a 500 W PEM fuel-cell-powered aircraft (Bradley et al. 2007). KAIST investigated the current and voltage behaviors of a NaBH_4 fuel cell through a flight test of a UAV, which contained a 100 W fuel cell combined with a battery. Michigan State University and Adaptive Materials recorded 10 h of flight with a SOFC fuel cell that uses propane, a hydrocarbon fuel (Crumm 2006). Table 25.2 shows some representative UAVs that utilize solar cells as their primary power source (Boucher 1984; Herwitz et al. 2004; Noth 2006, 2008). Evidently, recent applications of solar cells in both manned and unmanned vehicles show remarkable progress in endurance. Solar Impulse, a manned aircraft, recorded 27 h of flight endurance, while Zephyr, a UAV from QinetiQ, recorded a flight time of 2 weeks in the air. To accomplish missions involving such long-endurance flights, the aircraft must be equipped with secondary high-capacity batteries for overnight operation.

This chapter describes the modeling of a hybrid system that employs all three major electric power sources, i.e., solar cell, fuel cell, and battery, and an associated power management system (PMS). The PMS is necessary to efficiently provide the UAV propulsion system with the required power from multiple power sources. A PMS can be passive or active. A passive PMS is the simplest form of power

Table 25.1 Fuel cell-powered UAV (Bradley et al. 2009)

| Organization (date) | Fuel cell type | Reactant storage type | Endurance (estimated) |
|--|----------------|-----------------------|-----------------------|
| AeroVironment, Inc. (2003) | PEM | H2 sodium borohydride | 0.2 h |
| AeroVironment, Inc. (2005) | PEM | H2 cryogenic | 24 h |
| Fachhochschule Wiesbaden (2005) | PEM | H2 gaseous | 90 s |
| Naval Research Laboratory (2006) | PEM | H2 gaseous | 3.3 h |
| Adaptive Materials, Inc. (2006) | SOFC | Propane | 4 h |
| Georgia Institute of Technology (2006) | PEM | H2 gaseous | 0.75 h |
| California State University, Los Angeles (2006) | PEM | H2 gaseous | 0.75 h |
| DLR, German Aerospace Research Center, HyFish (2006) | PEM | H2 gaseous | 0.25 h |
| California State University, Los Angeles, and Oklahoma State University (2007) | PEM | H2 gaseous | 12 h |
| Korea Advanced Institute of Science and Technology (2007) | PEM | H2 sodium borohydride | 10 h |
| AeroVironment, Inc. (2007) | PEM | H2 sodium borohydride | 9 h |

Table 25.2 Solar cell-powered UAV (Noth 2008)

| Organization, product (date) | Type | Endurance (estimated) |
|-----------------------------------|-------------|-----------------------|
| AstroFlight, Inc., Sunrise (1974) | Early stage | 20 min |
| NASA, Pathfinder-Plus (2002) | | 15 h |
| NASA, Helios (2003) | | 16 min |
| AcPropulsion, Solong (2005) | | 48 h, 16 min |
| Sky-Sailor (2008) | HALE | 27 h, 5 min |
| QinetiQ, Zephyr (2010) | | 14 days, 21 min |

management, and it only protects the power sources; the power output is manually controlled according to the behavior of the UAV propulsion system. On the other hand, an active PMS has the full authority to control both the power sources and output. In a passive management system, the efficiency of power usage and the safety of the power system are not as optimized as they can be in an active management system. However, as passive management does not require a power converter or a power controller, it makes the PMS simpler, yielding benefits in weight reduction. In addition, it can avoid the collateral power loss associated with the power conversion. Therefore, passive management is more widely used

in small UAVs, while active management is used in most hybrid electric vehicles (HEVs), which demand higher efficiency at the expense of weight reduction. The power simulation conducted in this chapter demonstrates both the passive and active managements.



Although the operational behavior and proper capacity of each power source in the hybrid propulsion system should be determined, it is too costly and time-consuming to evaluate it through tests of the onboard system. Therefore, it is necessary to formulate simulation models of the power sources and integrate the models to estimate the behavior of each power source in the prescribed flight profile before deployment of the system on the actual UAV. In cases involving hybrid and electric vehicles, there have been extensive studies in modeling and simulation (Powell et al. 1998; Butler et al. 1999; Rizzoni et al. 1999; Lin et al. 2001; He and Hodgson 2002a, b; Gao et al. 2005), and various mathematical models and simulators are available for HEV systems, e.g., ADvanced VehIcle SimulatOR (Wipke et al. 1999; Markel et al. 2002), Virtual Test Bed, and PSIM.

Each power source investigated in this chapter is modeled in MATLAB/Simulink, and the component models are verified with published data from the manufacturers as well as flight test data. The component models are integrated into one power system, and the integrated power system model is used in the simulation of the electric propulsion system. The target platform is the Electric Aerial Vehicle-2 (EAV-2), which was developed by Korea Aerospace Research Institute (KARI). Prior to implementing it in the target platform, the integrated system was constructed for EAV-1 to verify the simulation method. EAV-1, which was also developed by KARI, employs a fuel cell and batteries, making it an appropriate UAV to verify the system in terms of its functionality involving multiple power sources. The simulation results of EAV-1 are compared with flight test data, and the feasibility of the simulation method is verified. Using the verified simulation method, this chapter investigates the capacity determination and the behavior of the power output in terms of passive and active power managements.

25.2 Hybrid Electric Propulsion System

EAV-1 employs a fuel cell and batteries as its power sources, while EAV-2 has solar cells, fuel cells, and batteries. The main features and dimensions of these aircraft are shown in Table 25.3. The EAV-2 is a low-speed, long-endurance UAV with the potential to conduct ground observation and surveillance using its payload camera. Using its onboard flight control computer (FCC), it can perform autonomous flights according to mission commands transmitted from the Ground Control Station (GCS). Its maximum gross weight is 18.0 kg, and the wingspan reaches 6.9 m. It has a fuselage with a diameter of 0.19 m, which is equipped with 0.003 m² NACA intakes on both the left and right sides for supplying air to fuel cells. The heated air and fuel cell's exhaust are vented through air outlets on the top, left, and right sides at the rear of the fuselage. The cross-sectional shape of the

Table 25.3 Specifications of EAV-1 and EAV-2

| Parameter | EAV-1 | EAV-2 |
|---------------------------|---|--|
| Picture |  |  |
| Weight | | |
| Structure, kg | 2.8 | 7.0 |
| Propulsion, kg | 2.5 | 8.5 |
| FCC, payload, kg | 1.2 | 1.5 |
| Power management system | – | 1.0 |
| Power range/manufacturor | | |
| Solar cell, W | – | 0 ~ 240/SunPower |
| Fuel cell, W | 0 ~ 200/Horizon | 0 ~ 400/Horizon |
| Battery, W | 0 ~ 400/Thunder Power RC | 0 ~ 1,300/Enerland |
| Wing/airfoil | | |
| Wing area, m ² | 0.68 | 1.92 |
| Wingspan, m | 2.4 | 6.9 |
| Aspect ratio | 10 | 20.0 |
| C _{L,max} | 1.17 | 1.7 |
| Airfoil shape | SD7032 modified | SG6043 |
| Motor/propeller | | |
| Motor, outrunner type | AXI 4120/20 | AXI 5320/34 |
| Propeller type | Fixed, pusher | Folding, tractor |
| Propeller size, in. | 14 × 9.5 | 21 × 13, 5° tilt hub |
| Operating condition | | |
| Endurance, h | 4.5 | 4.5 |
| Cruise speed, m/s | 12.9 | 12.1 |

intake is adjusted in accordance with the fuselage cross section. The required cruise power determined by aerodynamic analysis is 400 W, with a maximum power rating of 1.3 kW. Taking into account the 25 W of additional power needed for avionics and payload, the total cruise power required is 425 W. This amount of power is generated by the hybrid electric propulsion system, as it combines the use of the three power sources: solar cells, fuel cells, and batteries. This chapter determines the optimized analytical capacity of each power source based on the performance requirements of the platform aircraft, the output behavior of the power sources, wing area, and specifications of available commercial components.

Each power source is summarized as follows: The solar cells generate electricity from light energy. Because they use an external energy source, their activity is not limited by the quantity of any onboard item, such as fuel. However, because their

energy source is external, their output is heavily dependent on external conditions such as solar irradiation. In addition, they require large areas to generate sufficient power for flight. In contrast, fuel cells provide stable electricity output and are minimally affected by external conditions. Their stacks also have relatively high power density. However, their operation is limited by the amount of onboard fuel, and, furthermore, the fuel influences the total weight of the aircraft. The fuel reservoir, generally hydrogen, is very heavy. Fuel cells also have limitations in their inherent characteristics, such as a long start-up time and poor responses to instantaneous power demands. Batteries, like fuel cells, have finite capacities. Although the energy density of a battery is low, the power density is very high. Therefore, the combination of batteries and fuel cells provides a hybrid power generation system with benefits of both the high power density of batteries and the high energy density of fuel cells (Jiang and Dougal 2004).

EAV-2's passive-controlled and active-controlled electric propulsion systems, each with their three power sources, are illustrated in Figs. 25.1 and 25.2, respectively. In the passive PMS, each power source is directly connected to the power bus without the use of a converter or controller. Therefore, the output terminal voltage range of every power source is the same as the bus voltage range. This leads to the fact that the operational voltage ranges of the solar cells, fuel cells, and batteries should be matched to the bus voltage range. Furthermore, the bus voltage is determined by the battery, which functions as a reservoir for the electric power. The battery used is a Li-Polymer battery with a nominal voltage of unit cell of 3.7 V. Seven unit cells are connected in series, yielding a nominal voltage of 25.9 V. The solar cells are constructed in the same method, with the proper number of cells, and the fuel cells are selected from commercially available items.

25.3 Electric Power Sources Modeling

The power sources are modeled in MATLAB/Simulink, and the models are verified using the manufacturer's data and test data.

25.3.1 Solar Cell

This study employs the empirical model that represents the equivalent electrical circuit of a solar cell as a single module or an array of units. The empirical model is a curve fit of the measured current-voltage (I-V) relationship and requires fewer parameters than the one-diode and the two-diode models. The parameters of this model are usually available from the manufacturer's data sheet. Therefore, an empirical model can be easily constructed with an acceptable accuracy that the manufacturer verifies. The empirical model for C60 cells from SunPower is used; C60 cells are the cells used by EAV-2.

Table 25.4 shows the manufacturer's data for 50 serial C60 cells under an ambient solar irradiation of $1,000 \text{ W/m}^2$ and a cell temperature of 25°C . Under the specified

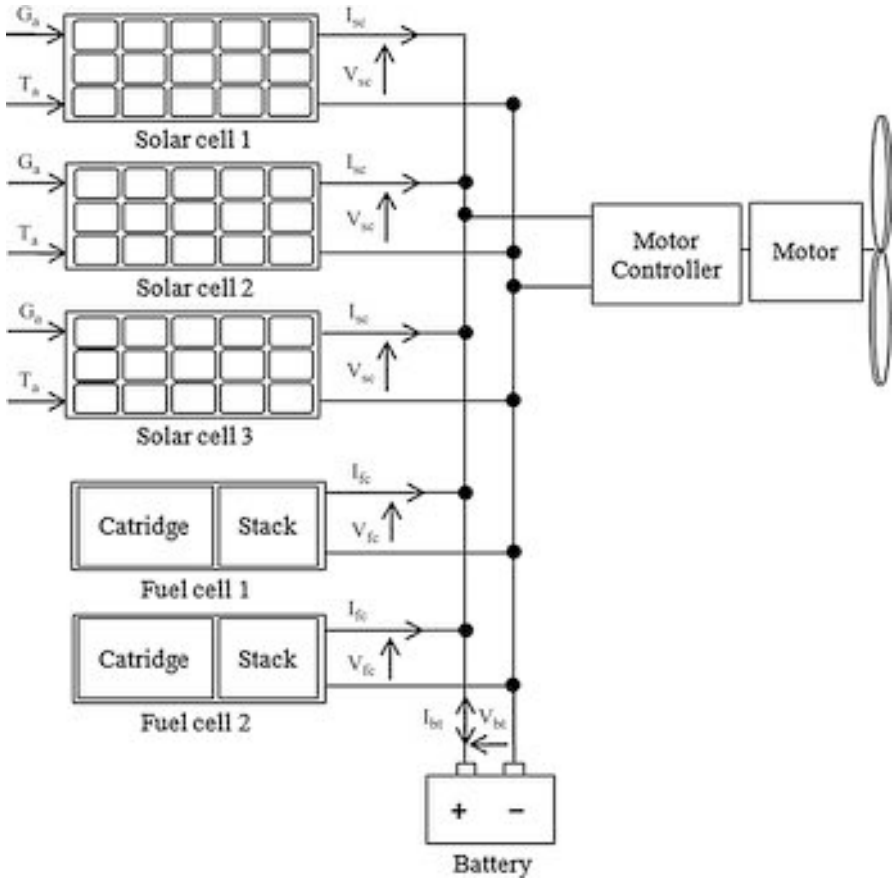


Fig. 25.1 EAV-2 passive electric control schematic

conditions, the net current from the solar cell module, which is the difference between photo current I_{ph} and diode current I_d , is given as

$$I = I_{ph} - I_d = I_{ph} - I_0 \left(\exp \frac{e(V + IR_s)}{akT_c} - 1 \right) \quad (25.1)$$

where I_0 is dark saturation current, R_s is serial resistance, k is Boltzmann's constant, and T_c is cell temperature. Idealizing factor a is applied.

Open-circuit voltage V_{oc} is the voltage when I is equal to 0, and it is related to thermal voltage V_t as follows:

$$V_{oc} = \frac{akT_c}{e} \ln \left(\frac{I_{ph}}{I_0} \right) = V_t \ln \left(\frac{I_{ph}}{I_0} \right) \quad (25.2)$$

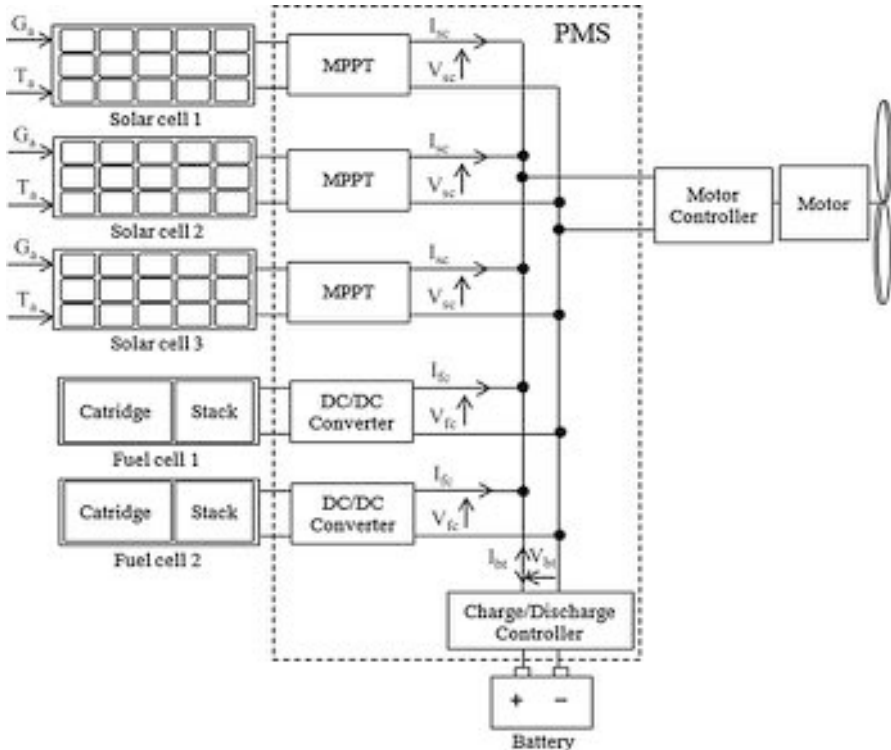


Fig. 25.2 EAV-2 active electric control schematic

Table 25.4 Solar cell data

| Parameter | Value |
|--------------------------|-------|
| Peak power, W | 238 |
| Rated voltage, V | 28.1 |
| Rated current, A | 5.88 |
| Open-circuit voltage, V | 33.7 |
| Short-circuit current, A | 6.25 |

Using the superscript C for a unit cell and M for a module of multiple cells, the electric current I^M under arbitrary operating conditions (V^M , ambient temperature T_a , solar irradiation G_a) can be calculated as shown in Eq. (25.3) (Hansen et al. 2000):

$$I^M = I_{sc}^M \left[1 - \exp \left(\frac{V^M - V_{oc}^M + R_s^M I^M}{N_{SM} V_t^C} \right) \right] \tag{25.3}$$

where N_{SM} is series module.

The simulated current-voltage (I-V) curves of the MATLAB/Simulink model for 50 serial solar cells are shown in Fig. 25.3 with the manufacturer’s data for

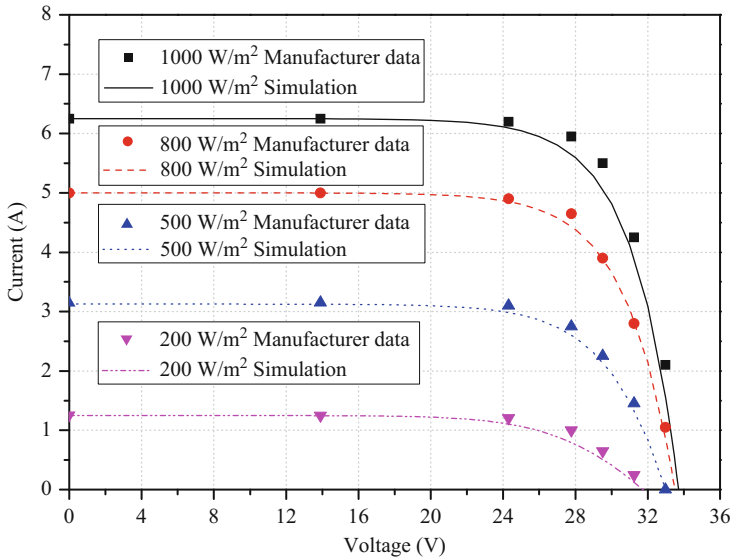


Fig. 25.3 Solar cell (one-cell 50 series) I-V curves

verification. The figure depicts the current behavior that is in accordance to the solar irradiation at the ambient temperature of 25 °C. The current remains constant until the output voltage reaches a threshold, and it drops if the voltage increases beyond the threshold. At this point, because power is the product of voltage and current, the power varies depending on the load voltage even under a constant light intensity. The point that generates maximum power is called the maximum power point (MPP). The power, according to the voltage under different sunlight intensities, is shown in Fig. 25.4. The MPP voltage decreases slightly as the light intensity decreases.

25.3.2 Fuel Cell

The estimation of the dynamic behavior of the fuel cell stack follows the mathematical models presented in Buasri and Salameh (2006) and Lee and Wang (2007). The theoretical optimum voltage of a unit cell is 1.2 V for any operational current. However, the actual voltage accounts for voltage lost due to activation V_{act} , voltage lost due to ohmic loss V_{ohmic} , and voltage lost due to concentration loss V_{con} as follows:

$$V_{fc} = E_{Nernst} - V_{act} - V_{ohmic} - V_{con} \quad (25.4)$$

where Nernst potential E_{Nernst} and the voltage losses are determined using Eqs. (25.5)–(25.8):

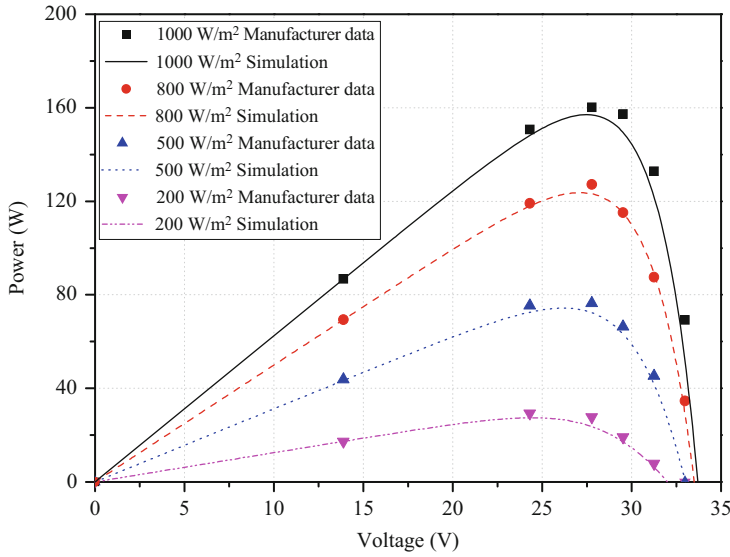


Fig. 25.4 Solar cell (one-cell 50 series) P-V curves

$$E_{\text{Nernst}} = 1.229 - 0.85 \times 10^{-3}(T - 298.15) + 4.3085 \times 10^{-5}T[\ln(P_{\text{H}_2}) + 0.5 \ln(P_{\text{O}_2})] \quad (25.5)$$

$$V_{\text{act}} = -[\xi_1 + \xi_2 T + \xi_3 T \ln(C_{\text{O}_2}) + \xi_4 T \ln(I_{\text{fc}})] \quad (25.6)$$

$$V_{\text{ohmic}} = I_{\text{fc}}(R_{\text{m}} + R_{\text{c}}) \quad (25.7)$$

$$V_{\text{con}} = -B_{\text{f}} \ln(1 - J/J_{\text{max}}) \quad (25.8)$$

where P_{H_2} and P_{O_2} are partial pressure of hydrogen and oxygen, respectively. $\xi_{1,2,3,4}$ is empirical coefficients of activation over voltage, and C_{O_2} is concentration of oxygen in the catalytic interface of the cathode. R_{m} and R_{c} are equivalent membrane resistance to proton conduction and equivalent contact resistance to electron conduction, respectively. B_{f} is constant dependent on the cell type and its operation state. J and J_{max} are actual cell current density and maximum cell current density, respectively.

A fuel cell stack voltage V_{stack} is a serial connection of unit cells, and its voltage is the unit voltage multiplied by the number of cells n , such that

$$V_{\text{stack}} = n \cdot V_{\text{fc}} \quad (25.9)$$

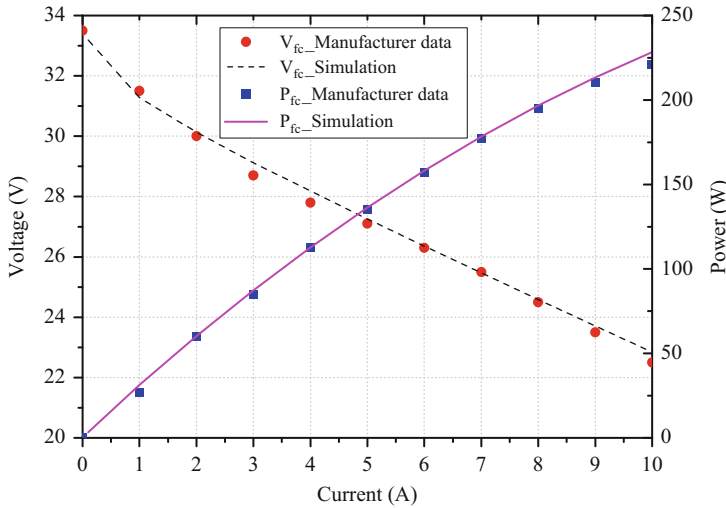


Fig. 25.5 Fuel cell I-V and P curves

The dynamic voltage of the fuel cell v_d is determined as follows:

$$\frac{dv_d}{dt} = \frac{1}{C} I_{fc} + \frac{1}{\tau} v_d \quad (25.10)$$

where fuel cell electrical time constant τ is given as

$$\tau = C \left(\frac{V_{act} + V_{con}}{I_{fc}} \right) \quad (25.11)$$

The dynamic behavior of the fuel cell voltage reflects the dynamic voltage as follows (Qingshan et al. 2008):

$$V_{fc} = E_{Nernst} - V_{ohmic} - v_d \quad (25.12)$$

A 200 W PEM-type fuel cell from Horizon Energy Systems is employed. It has a stack combined with a fuel cartridge that chemically generates hydrogen. The generated hydrogen is first stored in a small tank that maintains 4–6 bar, and then it is sent to the stack. Therefore, ideally, the generated hydrogen behaves identically to the hydrogen provided by a pressurized reservoir. The molar fractions of hydrogen and oxygen are assumed to be constant for all flight envelopes.

The fuel cell model is implemented in MATLAB/Simulink. The results of the simulation and the performance data from the manufacturer are shown together in Fig. 25.5.

Table 25.5 Battery data

| Parameter | Value |
|------------------------------|-------|
| Battery type | Li-Po |
| Nominal voltage, V | 25.9 |
| Rated capacity, Ah | 4.3 |
| Initial SOC, % | 100 |
| Nominal discharge current, A | 2.3 |

25.3.3 Battery

The dynamic model of a battery’s charge and discharge follows Tremblay and Dessaint (2009), as shown in Eqs. (25.13) and (25.14):

Charge: $V_{bt} = E_0 - R \cdot I - K \cdot \frac{Q}{it - 0.1 \cdot Q} \cdot i^* - K \cdot \frac{Q}{Q - it} \cdot it + A \cdot \exp(-B \cdot it)$

(25.13)

Discharge: $V_{bt} = E_0 - R \cdot I - K \cdot \frac{Q}{Q - it} \cdot (it + i^*) + A \cdot \exp(-B \cdot it)$

(25.14)

where A and B are exponential zone amplitude and exponential zone time constant inverse, respectively. E_0 is battery constant voltage; K is polarization constant. it and i^* are actual battery charge and filtered current, respectively. Q is battery capacity.

The parameters of this dynamic model are easily available from the manufacturer’s data sheet. Another advantage of this model is that it can be directly implemented in MATLAB/Simulink using library blocks. The battery model is verified with the battery block parameters shown in Table 25.5 for seven serially connected PQ4550XQ batteries from Enerland Co., Ltd. Figure 25.6 shows the simulation and test results of voltage and capacity of the battery pack in discharge at 2.3 A (0.5 C) and 4.5 A (1.0 C). The simulation generally agrees with the test results, with little deviation in capacity at roughly 1,000 mAh.

25.4 Simulation Modeling Verification with Flight Test Data

EAV-1 can fly in autonomous mode and path tracking for prescribed trajectories. In addition, it returns to the base and makes an emergency landing when it loses communication with the controller. It is equipped with an onboard camera and transmits video images and power source information to the ground station. EAV-1’s range reaches 3 km. The EAV-1 flight tests were conducted three times during October 18–19, 2010, at the Goheung Aerospace Center, Korea. The aircraft was programmed to loiter in autonomous flight mode, forming a specified trajectory within a radius of 500 m. It recorded an endurance of 4.5 h at an altitude of 450 m on the second day. The specifications of EAV-1’s fuel cell are determined according to the cruise condition so that it reaches a power output of 300 W and a capacity

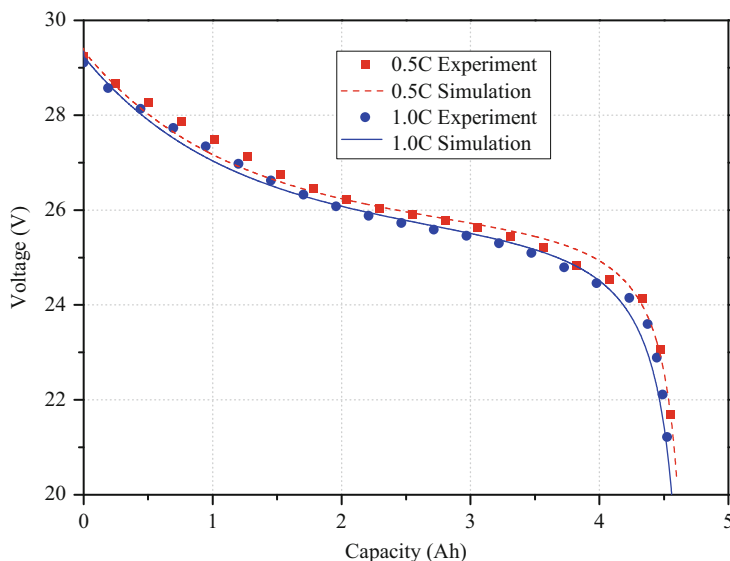


Fig. 25.6 Battery voltage-capacity curves

of 950 Wh, while the capacity of the battery pack is determined to be 30 Wh considering its weight.

The simulation method is verified with a hybrid power system of a fuel cell and batteries applied to EAV-1, whose configuration is shown in Fig. 25.7. Before verification, the results of the flight test were investigated as follows: The average fuel cell voltage, given a 95 % confidence level, is shown in Fig. 25.8 as a function of current over a flight time of 4.5 h. It is noticeable that the fuel cell that was installed and tested in EAV-1 reaches higher operational current than indicated levels of the manufacturer's data shown in Fig. 25.5, yielding a maximum power of 300 W. The dispersion of the data is ± 1 V on average. The main factors that affect the dispersion are dynamic responses of the fuel cell due to the internal hardware, fluctuation in hydrogen supply pressure (insufficient hydrogen), and vapor purge control of the stack. The fuel cell simulation model is adjusted to reflect the I-V curve from the test data.

EAV-1's battery pack consists of six Li-Polymer batteries in series, which are grouped into two serial packs of three each. The individual batteries are Pro Lite V2s from Thunder Power RC. Output from the battery pack passes through a converter and combines with output from the fuel cell to provide the power required for EAV-1. The converter, which comes with the fuel cell, breaks the circuit for a current input of 230 mA or higher. The battery pack also supplies the additional power that cannot be provided by the fuel cell during takeoff and transient flights. Figure 25.9 shows the results of a simulation and the tests on the battery and converter while discharging at 700 mA (0.5 C based on the battery capacity) and charging at 230 mA.

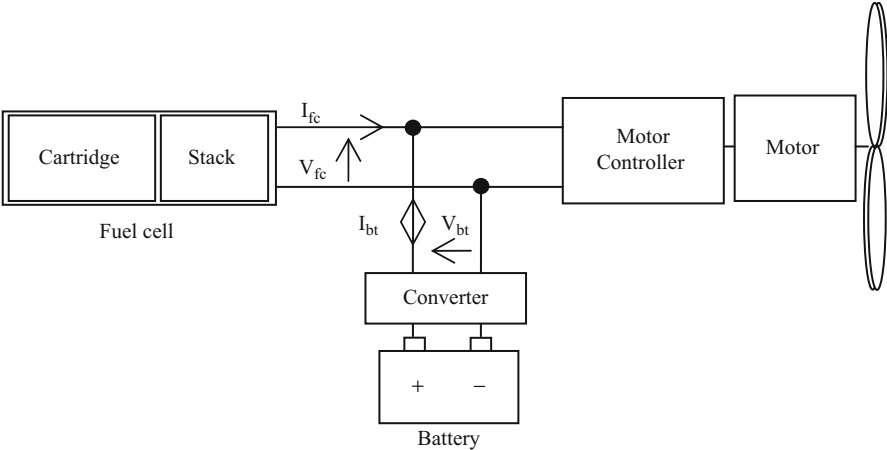


Fig. 25.7 EAV-I passive electric control schematic

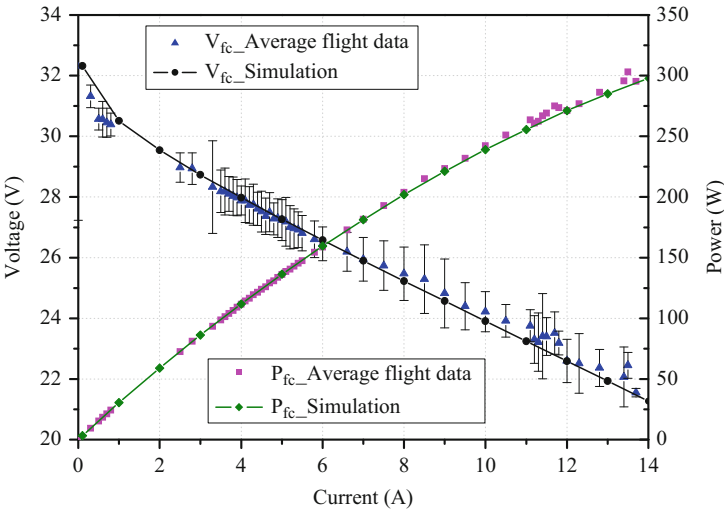


Fig. 25.8 Fuel cell I-V and P curves

The discharge test indicates that the capacity of the Pro Lite V2 battery is 1.26 Ah, which is 93 % of the manufacturer’s data of 1.36 Ah.

When the fuel cell, the primary power source, provides the electricity to meet the power required of the system, the voltage temporarily drops along the I-V curve and the fuel cell takes several seconds to begin producing the required power. In the meantime, the fuel supply may not be sufficient and the electrocatalyst’s life may be reduced (Thounthong and Sethakul 2007). Therefore, the power sources should

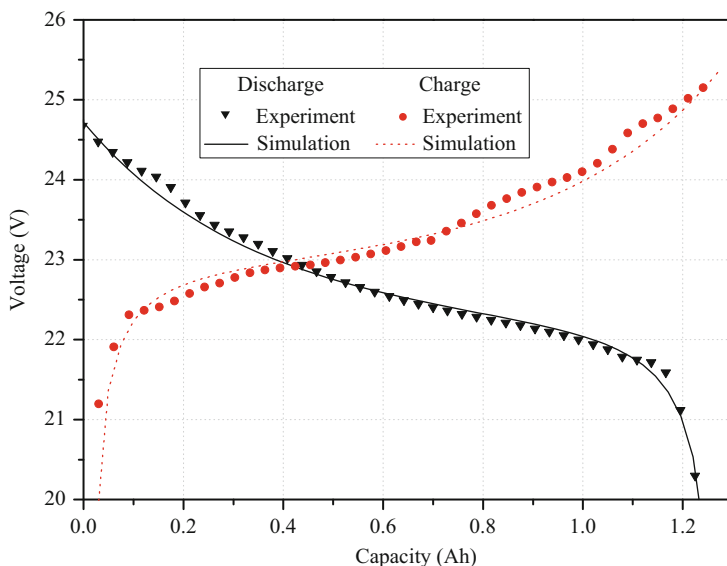


Fig. 25.9 Battery and converter characteristic curves

be controlled so that the fuel cell is maintained at a steady state, as the battery compensates for the shortage of power. Once the battery's voltage is matched to the fuel cell's voltage, the current is adjusted along the I-V curve of the battery. This yields a power balance between the power required and the summation of outputs from the fuel cell and the battery.

The measured and simulated time histories of the voltage and residual current of the fuel cell in the fuel cell-battery-integrated system are shown in Fig. 25.10. The severe fluctuations in the flight test results before the 2.5-h mark result from the built-in system controller of the fuel cell, which maintains the hydrogen tank pressure at 4–5.5 bar and breaks the current to prevent further pressure drop if the pressure stays under 4 bar for 6 s or more. In this case, the voltage of the system becomes the open-circuit voltage. As the simulation assumes a steady supply of hydrogen, the results do not reflect the fluctuations that are caused by circuit breaks. However, the general trend of the simulated results closely resembles the test data, and good agreement is clearly seen after 2.5 h, at which point the controller no longer breaks the current.

Figures 25.11 and 25.12 indicate that the fuel cell is the primary power source in the regime that requires cruise power, and extra power is provided by the battery for takeoff and transient regimes. Meanwhile, the excess power from the fuel cell charges the battery. The simulation underestimates power output from the battery during the takeoff and transient flight because the fuel cell model provides power faster than the actual fuel cell. The response time of the model differs from that of the actual fuel cell because the fuel cell model employed by this study is

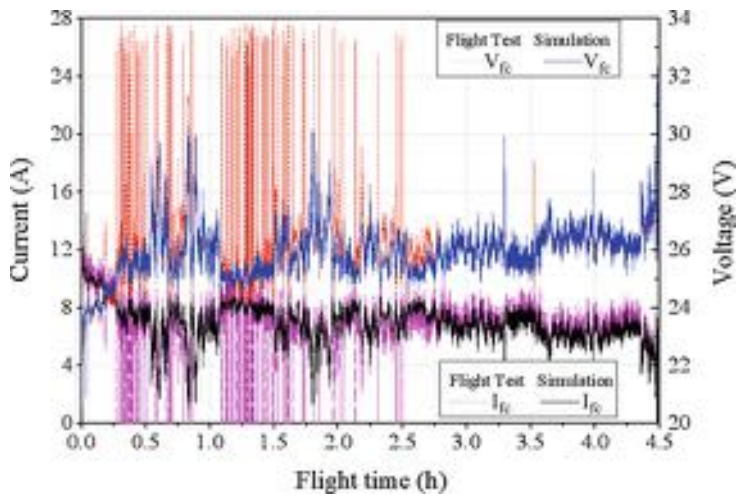


Fig. 25.10 Fuel cell current and voltage variation during the whole flight

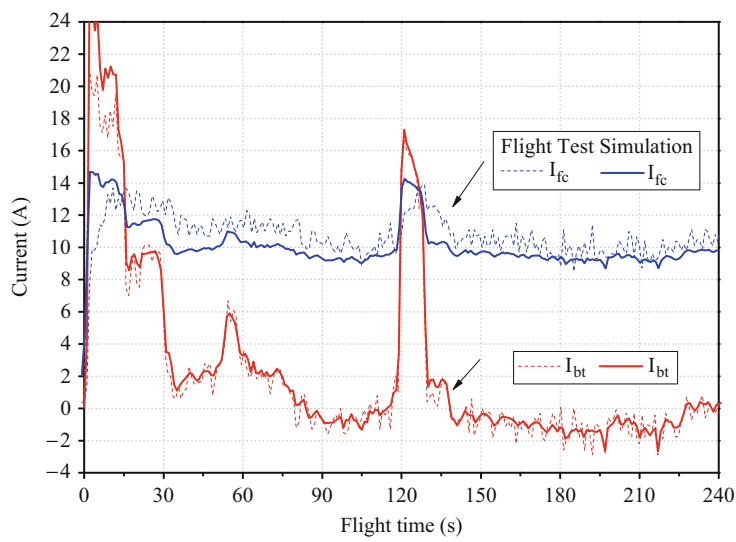


Fig. 25.11 Current variations at takeoff

a simplified one, and it lacks dynamic response of detailed components for the particular Aeropak fuel cell. These detailed components include the anode model and cathode model, the stack controller that controls hydrogen pressure and vapor purge, and the associated balance of plant (BOP). However, these omissions are acceptable because the primary interests of this study are the overall behavior of the output during the flight and the remaining capacity of each power source of

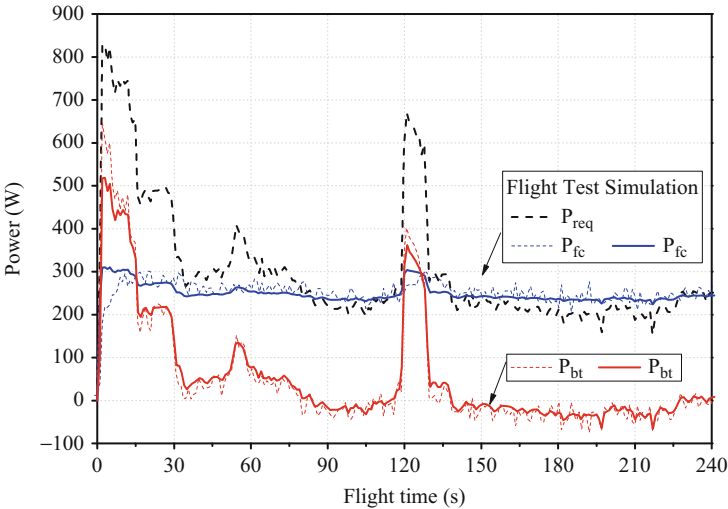


Fig. 25.12 Power variations at takeoff

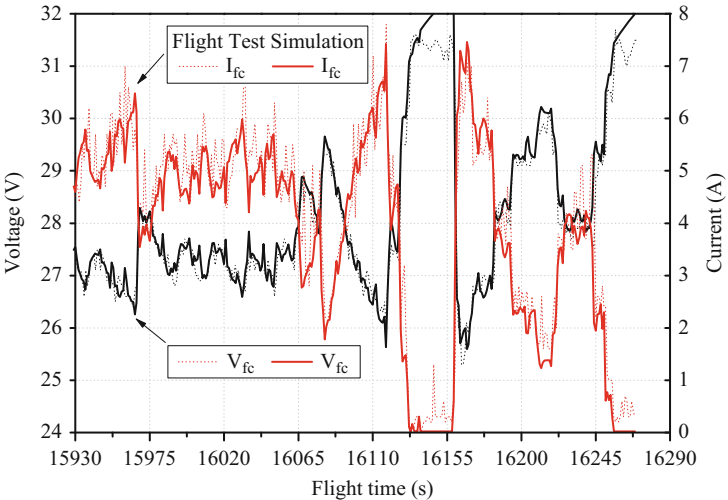


Fig. 25.13 Fuel cell voltage and current variations at landing

the hybrid power system. Figure 25.13 shows the time histories of the voltage and current of the fuel cell during landing, and Fig. 25.14 shows the power variation under the same conditions. In the simulation, because the aircraft manages flight conditions using the power available from the fuel cell, the battery output seldom varies. However, the actual battery is charged and discharged as the output of the fuel

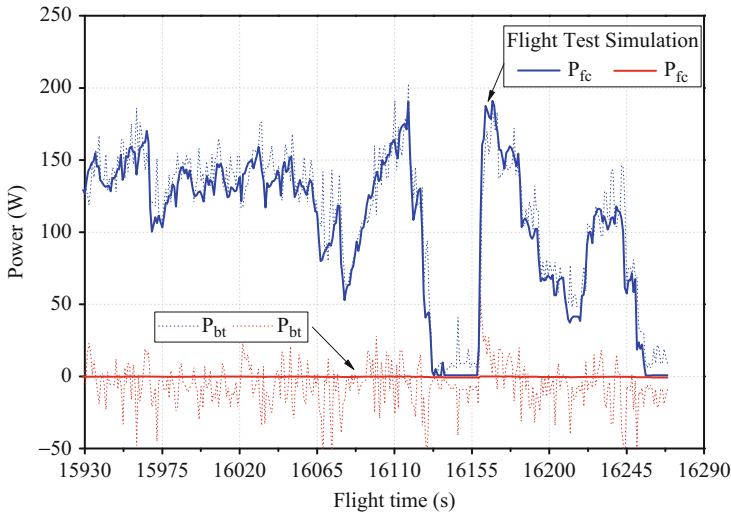


Fig. 25.14 Power variations at landing

cell fluctuates. This fluctuation is caused by many reasons that are not reflected in the model, such as varying hydrogen pressure, vapor purge, and external disturbances.

The simulation results of the component models and the integrated model closely reflect the test results. This means that the simulation method provides a feasible estimation tool for the power behavior of an electric propulsion system that employs multiple power sources in types and classes. Based on this conclusion, the same simulation method is used for EAV-2, which employs solar cells as well.

25.5 EAV-2 Electric Propulsion System Simulation

25.5.1 Passive Simulation

It is necessary to yield a power balance between the power required and the summation of outputs from the solar cells, fuel cells, and batteries during the simulation. The current and voltage of each power source is determined by its characteristic curve for the supply of the required power. Because the integrated propulsion system model of EAV-2 consists of power sources directly connected to the power bus without a converter, the operational bus voltage range becomes the output voltage range of each power source. Once the solar cell's output voltage is determined by the real-time battery voltage, the current from the solar cell is determined via Eq. (25.3). It is apparent from the solar cell performance charts that for a moderate output voltage, the current is practically determined by the solar irradiation G_a and ambient temperature T_a . In the same way, the output voltage of the fuel cell stack is matched to the battery pack voltage range by adjusting its

current along the I-V curve. Consequently, the power of the fuel cell stack falls out. If the fuel cell deviates from the manufacturer's I-V curve, the load may not be evenly distributed in a passive power management system. The sum of the currents from each power source is supposed to be the same as the current that the motor consumes depending on the power required at the moment. Therefore, the current of the battery I_{bt} is determined as follows:

$$I_{bt} = I_{req} - I_{sc} - I_{fc} \quad (25.15)$$

If the available power varies, the required current I_{req} will also vary, resulting in a different real-time battery voltage along the I-V curve.

The simulation is conducted on a power-required profile of EAV-2, which is scaled from that of EAV-1. In order to evaluate the effects of sunlight conditions G_a and T_a on the solar cell's output P_{sc} , the simulation conditions are set to approximate the winter solstice and summer solstice. Assuming that EAV-2 maintains a level flight for the whole flight envelope, the solar irradiation projected onto a level plane at the end of the atmosphere is calculated in Eq. (25.16). On the Earth's surface, the clearness index, 0.6 for c is multiplied by Eq. (25.16), yielding the solar irradiation G_a as Eq. (25.17) (Pefititsis et al. 1994):

$$G_{ref} = G_{SC} \left[1 + 0.033 \cos \left(\frac{360n}{365} \right) \right] \cdot (\cos \phi \cos \delta \cos \omega + \sin \phi \sin \delta) \quad (25.16)$$

$$G_a = c \cdot G_{ref} \quad (25.17)$$

where ϕ of the simulation site is 36.18° , δ varies between -23.5° and $+23.5^\circ$, and ω varies from -180° to 180° in 15° increments. The variation of the solar irradiation at this site is shown in Fig. 25.15. The figure shows a difference of 400 W/m^2 between the summer and winter solstices at around noon. The temperature variation data from the National Weather Service is shown in Fig. 25.16.

The time histories of the simulation at the winter solstice are shown in Figs. 25.17 and 25.18. During periods that require high power, i.e., 10–30 s for takeoff and 200–210 s for transient flight, the high-power-density battery provides most of the power required, causing the SOC of the battery to drop. For the period spanning 12–200 s, which corresponds to the cruise, the solar cells and fuel cells take charge of the power required. During this period, the battery's output drops to under 100 W, as it recovers its voltage, and the SOC decrease slows down.

Figure 25.19 shows data for EAV-2 performing a cruise for 4.5 h, from 10:00 am to 2:30 pm. As demonstrated in the figure, the maximum output from the solar cells reaches only 86 W at the winter solstice. Therefore, more power output is demanded from the fuel cells and batteries. The maximum power of the fuel cell stack with the support of the batteries may be sufficient for the mission at the winter solstice. However, in the case of passive management, the fuel cells operate under a partial

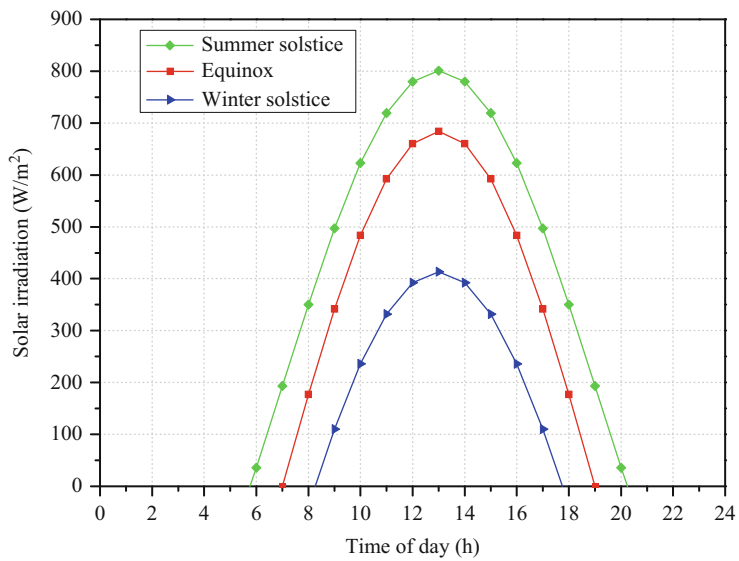


Fig. 25.15 Solar irradiation

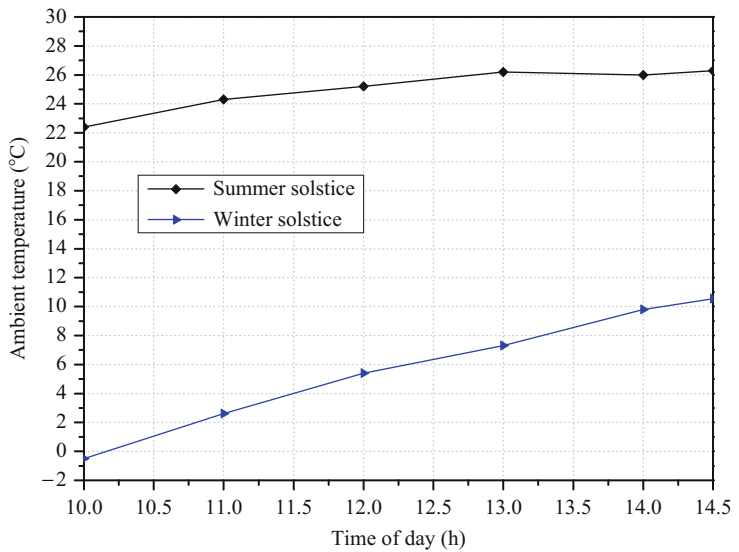


Fig. 25.16 Ambient temperature variations

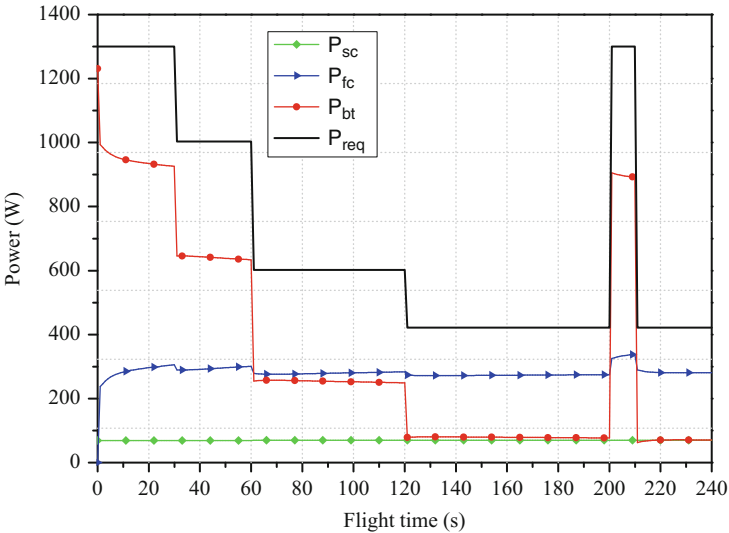


Fig. 25.17 Power variations at the winter solstice

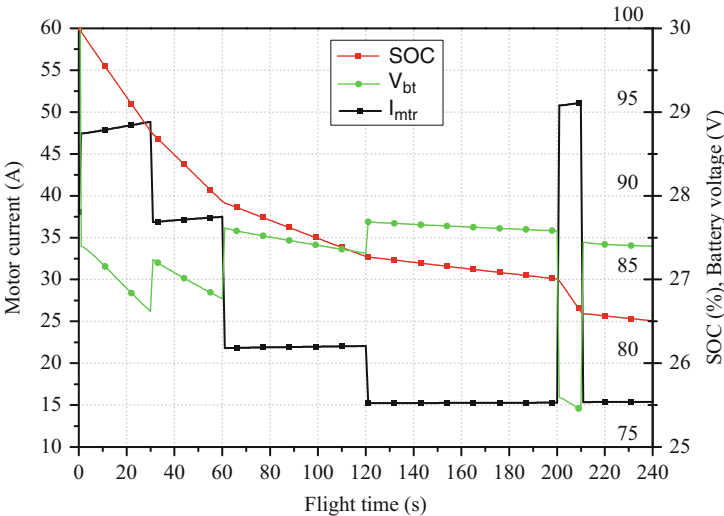


Fig. 25.18 Motor current and battery property variation at the winter solstice

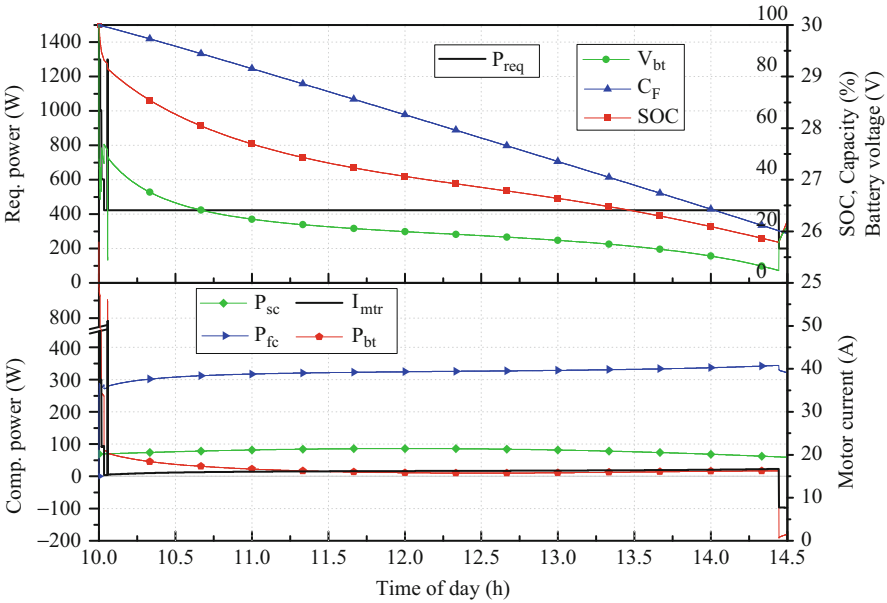


Fig. 25.19 Simulation results at the winter solstice

load with continuous discharge of the batteries because the maximum power of the fuel cell occurs when the battery’s voltage, i.e., the bus voltage, is close to complete discharge. In the case of the summer solstice shown in Fig. 25.20, the solar cells generate up to 178 W. Therefore, with the output from the fuel cell stack added, the aircraft can charge the batteries while it cruises. During both the winter and summer solstices, the batteries are charged during the landing, which requires only 200 W.

The energy fractions at the winter solstice and summer solstice are shown in Table 25.6. The solar cells provide 46.9 % higher energy at the summer solstice than at the winter solstice.

25.5.2 Active Simulation

In an emergency situation in which the solar cells are not fully functional due to external factors, e.g., clouds, and at the same time the fuel cell stack undergoes a power outage, the aircraft should attempt to land only with the power left in the batteries. Therefore, a certain level of SOC of the battery pack should be maintained during the flight as a precaution against such cases. The aircraft consumes 33 Wh over 10 min during landing. Assuming that the descent requires 200 W, this corresponds to 30 % of the SOC. Table 25.6 indicates that in the case of passive power management, the remaining SOC of the battery pack after 4.5 h of flight is only 24.2 % at the winter solstice, meaning that the system

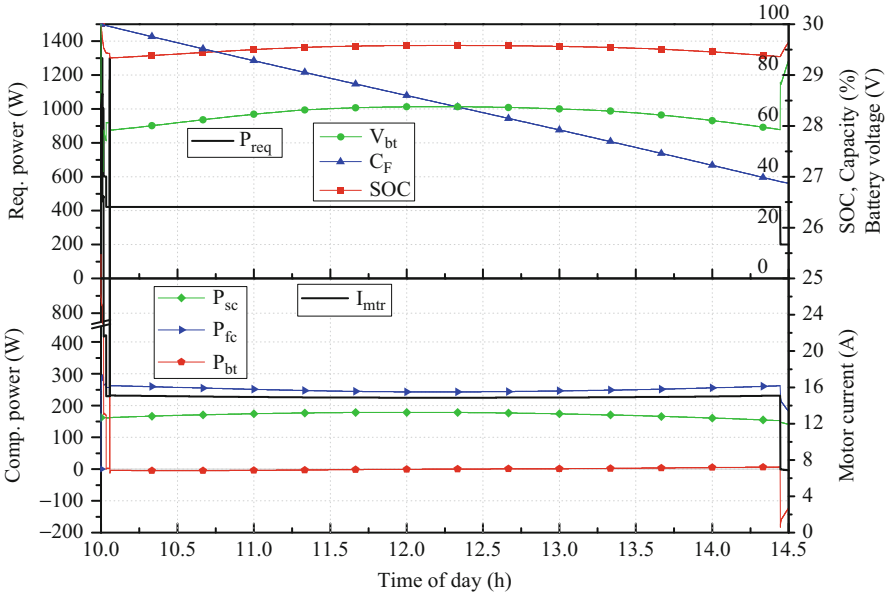


Fig. 25.20 Simulation results at the summer solstice

Table 25.6 Used energy portion from passive simulation

| Energy portion | Winter solstice | Summer solstice |
|----------------|-----------------|-----------------|
| Solar cell, % | 18.6 | 39.6 |
| Fuel cell, % | 76.4 | 59.9 |
| Capacity | 19.2 | 36.7 |
| Battery, % | 5.0 | 0.5 |
| Min. SOC | 16.0 | 86.4 |
| Final SOC | 24.2 | 93.0 |

requires adjustments to meet safety parameters. On the other hand, during the summer solstice, the remaining SOC reaches 93 %. This shows that the power is not efficiently distributed. To resolve this issue, the active power management takes control of the individual power sources. The active power management has a complex PMS and introduces more weight. However, it is beneficial to the safety because it actively handles any failure of power source and power distribution efficiency of the system because each power source can operate at its optimal condition.

Because the active power management system controls the output of individual power sources, the output voltage of each component is not necessarily the same as the system voltage. Therefore, DC/DC converters are required to match the output voltage of each component to the system voltage. A DC/DC is a passive element that matches input and output voltages, ideally without change in power. Therefore, it does not affect the results of the power simulations. This study does not account

for DC/DC converters in the power model and focuses primarily on the behavior of the power output with active management versus with passive simulations.

The active power management schemes widely used for the HEVs can be divided into the maximum SOC-of-PPS control strategy, the thermostat control strategy, and the constrained thermostat control strategy (Ehsani et al. 2009). The maximum SOC-of-PPS control strategy extracts power from each power source to supply the propulsion power required. At the same time, it charges the battery in real time to maintain a high SOC of the battery. For example, an automobile traveling in an urban area frequently makes quick starts and stops because of pedestrians and traffic signals. Therefore, the battery should be able to provide sufficient power for the entire duration. In such a case, the other power source must charge the battery to maintain a high SOC level. Therefore, the capacity of all the propulsion system, including the battery, has to be increased. However, for a UAV whose primary objective is long-endurance in the air, the weight of the system plays a crucial role in its mission capability. Furthermore, UAVs do not experience frequent high accelerations and transient flights. As a result, a power management to maintain a high SOC level is not a practical option for a UAV.

On the other hand, the thermostat control strategy turns off the fuel cell, the primary power source, and uses the other power sources for power supply when the power required is lower than 50 W preserving the fuel cell to be operated only at high efficiency conditions. It utilizes the fuel cell only when the power required is greater than 50 W. In this sense, automobiles usually employ this strategy to improve efficiency in highway operations. However, as the passive simulation in the previous section predicts that EAV-2's fuel cell stack operates under conditions with efficiency over 45 %, the on-off control for the fuel cell is unnecessary during the flight.

The constrained thermostat control strategy, which is employed in EAV-2, blends the two strategies above and controls output from each power source, maintaining a certain level of SOC of the battery. The active power management logic is illustrated in Fig. 25.21. The power from the solar cells is consumed with the highest priority because the solar cells do not consume the onboard fuel. Power from the solar cells that exceeds the power required is charged to the battery. Once the battery is fully charged, the controller reduces the output from the solar cells by increasing the system voltage and lowering the current from the solar cells. However, because of the high power demands made by an aircraft during the flight, the solar cells operate at their maximum capacity. For fuel cell control during the winter solstice, in order to maintain the battery's SOC at 30 %, the active power management controller assigns a higher output to the fuel cell stack than the passive management does. On the other hand, the controller assigns a lower output during the summer solstice to reduce the SOC. This logic is implemented using a constrained thermostat control strategy with an adding factor, as shown in Eq. (25.18):

$$P_{fc} = P_{req} - P_{sc} + P_{chg,m} \times (SOC_L - SOC) \quad (25.18)$$

where $P_{chg,m}$ is maximum charge power and SOC_L is low level of SOC.

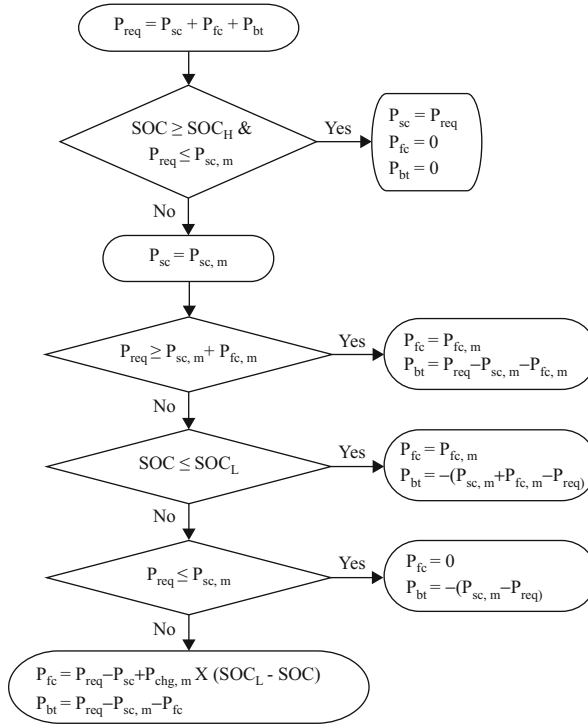


Fig. 25.21 Active power management logic

The controller demands its peak power from the battery for takeoff and transient flight. In addition, if the battery has a remaining capacity greater than SOC_L , the controller takes output from the battery to fill the power shortage of the solar cells and fuel cells. The results of the simulation that employs this management logic are shown in Figs. 25.22 and 25.23 for the winter and summer solstices, respectively. It is worth mentioning that the battery maintains an SOC of greater than 30 % after completion of 4.5 h of flight, as shown in Table 25.7.

25.6 Conclusion

In this chapter, the behavior of the power sources in a hybrid electric propulsion system for a UAV was simulated. The simulation was useful in confirming the feasibility of the capacities of the power sources and was helpful in predicting their characteristics in the UAV propulsion system. The propulsion system of the target UAV (EAV-2) consists of solar cells, fuel cells, and batteries. Each power source was modeled in MATLAB/Simulink and was verified with the manufacturer's data.

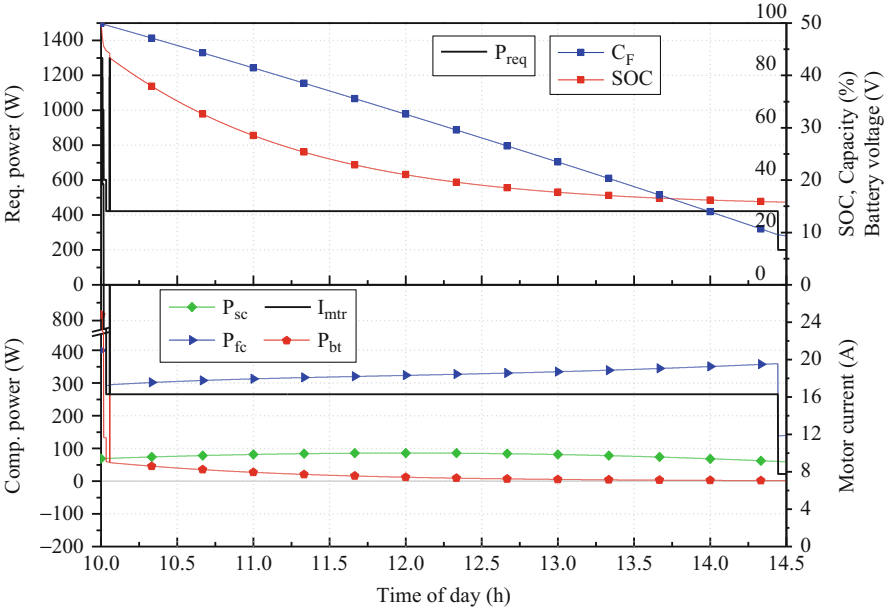


Fig. 25.22 Simulation results of applied active power management logic at the winter solstice

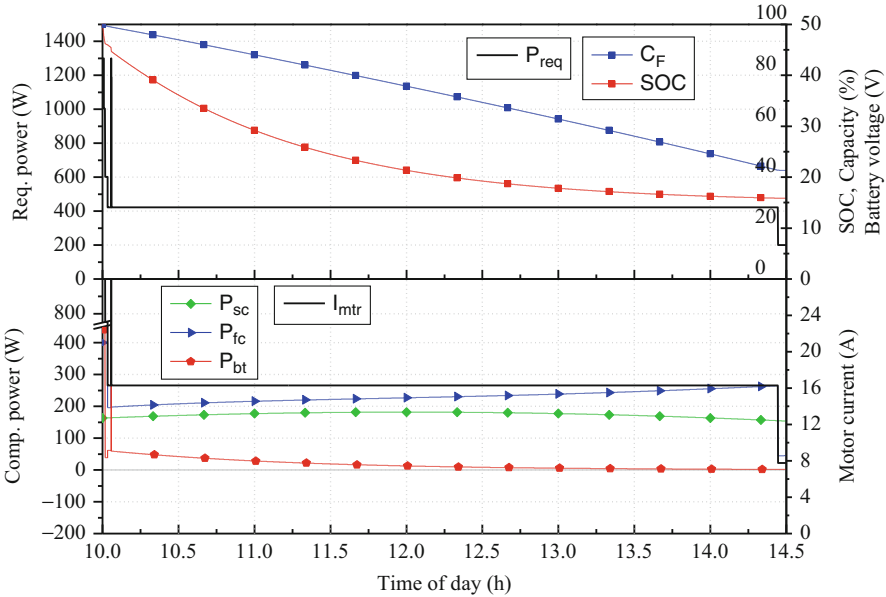


Fig. 25.23 Simulation results of applied active power management logic at the summer solstice

Table 25.7 Used energy portion from active simulation

| Energy portion | Winter solstice | Summer solstice |
|----------------|-----------------|-----------------|
| Solar cell, % | 18.5 | 39.6 |
| Fuel cell, % | 77.0 | 55.9 |
| Capacity | 18.7 | 42.6 |
| Battery, % | 4.5 | 4.5 |
| Final SOC | 31.6 | 31.6 |

In addition, the simulation of the power behavior of the system model was verified with the flight test data of EAV-1, which has a hybrid propulsion system consisting of a fuel cell and batteries.

The passive and active PMS were simulated and compared for the EAV-2 propulsion system. The simulation was conducted for duration of 4.5 h, from 10:00 am to 2:30 pm, at the winter and summer solstices. The components were selected for the operational conditions of passive management, in which the voltages of the components were matched. The simulation results of the passive management indicated that even without an active control applied to the system, the system operates reasonably and according to the characteristics of each power source: the solar cells and fuel cells operate as the primary power sources, and the batteries supplement them when peak power is required, e.g., during takeoff and transient flight. At the summer solstice, the solar cell generated 46.9 % more power than at the winter solstice, allowing the aircraft to fly only with the power from the solar cells and fuel cells, and as a result, it charged the batteries during the flight. The active management system, which controls the individual power sources, employed the constrained thermostat control strategy. The simulation results of active management indicated that it could maintain a minimum level of SOC for the battery pack, resulting in efficient power distribution and greater system safety.

In summary, the power simulation of a hybrid electric power system allows for prediction of the dynamic behavior of the system before the costly flight test, and it can be a useful tool in determining power distribution logic.

Nomenclature

A = exponential zone amplitude, V

a = idealizing factor

B = exponential zone time constant inverse, A/h

B_f = constant dependent on the cell type and its operation state, V

b = span, m

C = capacitance, F

C_F = capacity of fuel cell, %

C_{O_2} = concentration of oxygen in the catalytic interface of the cathode, mol/cm³

c = clearness index

E = energy, Wh

E_0 = battery constant voltage, V

E_{Nernst} = Nernst potential, V

e = electron charge, C

G_a = solar irradiation, W/m²

I = current, A

I_0 = dark saturation current, A

I_d = diode current, A

I_{ph} = photo current, A

it = actual battery charge, Ah

i^* = filtered current, A

J = actual cell current density, A/cm²

J_{max} = maximum cell current density, A/cm²

K = polarization constant, V/(Ah) or polarization resistance, Ω

k = Boltzmann's constant, J/K

N_{SM} = series module

n = number of cells

P = power, W

P_{H_2} = partial pressure of hydrogen, atm

P_{O_2} = partial pressure of oxygen, atm

Q = battery capacity, Ah

R = internal resistance, Ω

R_c = equivalent contact resistance to electron conduction, Ω

R_m = equivalent membrane resistance to proton conduction, Ω

R_s = serial resistance, Ω

s = wing area, m²

T = temperature, K

T_a = ambient temperature, °C

T_c = cell temperature, °C

V = voltage, V

V_{act} = voltage lost due to activation, V

V_{con} = Voltage lost concentration loss, V

V_{oc} = open-circuit voltage, V

V_{ohmic} = overvoltage due to ohmic loss, V

V_t = thermal voltage, V

W = weight, kg

v_d = dynamic voltage, V

τ = fuel cell electrical time constant

$\xi_1, \xi_2, \xi_3, \xi_4$ = empirical coefficients of activation over voltage

ϕ = latitude °

δ = solar declination °

ω = solar time, h

Subscripts

bt = battery
 chg = charge
 fc = fuel cell
 L = low limit
 m = maximum
 mtr = motor
 req = required
 sc = solar cell
 tot = total

References

- AeroVironment, Inc., *Aerovironment Flies World's First Liquid Hydrogen-Powered UAV* (AeroVironment, Inc., Monrovia, 2005)
- Anon, *Erfolgreicher Erstflug des Hyfish* (DLR, German Aerospace Center, Stuttgart, 2007)
- R.J. Boucher, History of solar flight, in *Proceeding of the 20th Joint Propulsion Conference*, AIAA-84-1429, Cincinnati, June 1984
- T.H. Bradley, B. Moffitt, D. Mavris, D.E. Parekh, Development and experimental characterization of a fuel cell powered aircraft. *J. Power Sources* **171**(2), 793–801 (2007). doi:10.1016/j.jpowsour.2007.06.215
- T.H. Bradley, B.A. Moffitt, T.F. Fuller, D.N. Mavris, D.E. Parekh, Comparison of design methods for fuel-cell-powered unmanned aerial vehicles. *J. Aircr.* **46**(6), 1945–1956 (2009)
- P. Buasri, Z.M. Salameh, An electrical circuit model for a proton exchange membrane fuel cell. IEEE, 1-4244-0493-2, Power Engineering Society General Meeting, (2006)
- K.L. Butler, M. Ehsani, P. Kamath, A Matlab-based modeling and simulation package for electric and hybrid electric vehicle design. *IEEE Trans. Veh. Technol.* **48**(6), 1770–1778 (1999)
- California State University, Los Angeles, *Cal State L.A.'s Fuel-Cell Plane Passes Key Flight Test* (California State University, Los Angeles, 2006)
- A. Crumm, Solid oxide fuel cell systems, in *Proceedings of the Fuel Cell Seminar*, Honolulu, Nov 2006
- M. Ehsani, Y. Gau, A. Emadi, Parallel hybrid electric drive train design, in *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles Fundamentals, Theory, and Design*, 2nd edn. (CRC Press, Boca Raton, 2009)
- W. Gao, S. Neema, J. Gray, J. Picone, S. Porandla, S. Musunuri, J. Mathews, Hybrid powertrain design using a domain-specific modeling environment, in *Proceedings of the IEEE Vehicle Power Propulsion Conference*, Chicago, 2005, pp. 6–12
- A.D. Hansen, P. Sørensen, L.H. Hansen, H. Bindner, *Models for Stand-Alone Statement System* (National Risø Laboratory, Roskilde, 2000)
- X. He, J.W. Hodgson, Modeling and simulation for hybrid electric vehicles-part I: modeling. *IEEE Trans. Intell. Transp. Syst.* **3**(4), 235–243 (2002a)
- X. He, J.W. Hodgson, Modeling and simulation for hybrid electric vehicles-part II: simulation. *IEEE Trans. Intell. Transp. Syst.* **3**(4), 244–251 (2002b)
- C. Herwerth, C. Chiang, A. Ko, S. Matsuyama, S.B. Choi, M. Mirmirani, D. Gamble, A. Arena, A. Koschany, G. Gu, T. Wankewycz, Development of a small long endurance hybrid PEM fuel cell powered UAV. Society of Automotive Engineers paper 2007-01-3930, Sept 2007

- S.R. Herwitz, L.F. Johnson, S.E. Dunagan, R.G. Higgins, D.V. Sullivan, J. Zheng, B.M. Lobitz, J.G. Leung, B.A. Gallmeyer, M. Aoyagi, R.E. Slye, J.A. Brass, Imaging from an unmanned aerial vehicle: agricultural surveillance and decision support. *Comput. Electron. Agric.* **44**(1), 49–61 (2004)
- Z. Jiang, R.A. Dougal, Control strategies for active power sharing in a fuel-cell-powered battery-charging station. *IEEE Trans. Ind. Appl.* **40**(3), 917–924 (2004)
- J. Kellogg, Fuel cells for micro air vehicles, in *Joint Service Power Exposition*, Tampa, May 2005
- D.J. Lee, L. Wang, Dynamic and steady-state performance of PEM fuel cells under various loading conditions. *IEEE, Power Engineering Society General Meeting*, 1-4244-1298-6 (2007)
- C.C. Lin, Z. Filipi, Y. Wang, L. Louca, H. Peng, D. Assanis, J. Stein, Integrated, feed-forward hybrid electric vehicle simulation in SIMULINK and its use for power management studies, in *Proceeding of the Society of Automotive Engineers, Inc.*, 2001-01-1334, Michigan, 2001
- T. Markel, A. Brooker, T. Hendricks, V. Johnson, K. Kelly, B. Kramer, M. O’Keefe, S. Sprik, K. Wipke, ADVISOR: a systems analysis tool for advanced vehicle modeling. *J. Power Sources* **110**(2), 255–266 (2002)
- V.P. McConnell, Military UAVs claiming the skies with fuel cell power. *Fuel Cells Bull.* **2007**(12), 12–15 (2007). doi:10.1016/S1464-2859(07)70438-8
- A. Noth, Design of a lightweight & high efficiency MPPT for Sky-Sailor airplane. Internal technical report, Autonomous Systems Laboratory, ETH Zürich, Feb 2006
- A. Noth, Design of solar powered airplanes for continuous flight, Dissertation, ETH, No. 18010, degree of Doctor of Technical Sciences, Zürich, Sept 2008
- D. Peftitsis, G. Adamidis, A. Balouktsis, An investigation of new control method for MPPT in PV array using DC-DC buck-boost converter, in *Proceeding 4th Annual Allerton Conference on Circuits and Systems Theory*, New York, 1994
- B.K. Powell, K.E. Bailey, S.R. Cikanek, Dynamic modeling and control of hybrid electric vehicle powertrain systems. *IEEE Control Syst. Mag.* **18**(5), 17–33 (1998)
- X. Qingshan, W. Nianchu, K. Ichianagi, K. Yukita, PEM fuel cell modeling and parameter influences of performance evaluation, in *DRPT2008*, Nanjing
- G. Rizzoni, L. Guzzella, B.M. Baumann, Unified modeling of hybrid electric vehicle drivetrains. *IEEE/ASME Trans. Mechatron.* **4**(3), 246–257 (1999)
- B. Scheppat, *Betriebsanleitung für das Brennstoff Zellenbetriebene Modellflugzeug* (Fachhochschule Wiesbaden, Wiesbaden, 2004)
- P. Thounthong, P. Sethakul, Analysis of fuel starvation phenomenon of a PEM fuel cell, in *IEEE 4th Power Conversion Conference*, Nagoya, 2007. doi:10.1109/PCCON.2007.373048
- O. Tremblay, L.A. Dessaint, Experimental validation of a battery dynamic model for EV applications. *World Electr. Veh. J.* **3**, 1–10 (2009). EVS24 Stavanger, Norway
- O. Velev, Summary of fuel cell projects: AeroVironment 1997–2007, in *National Hydrogen Association Fall 2007 Topical Forum*, Columbia, Oct 2007
- K.B. Wipke, M.R. Cuddy, S.D. Burch, ADVISOR 2.1: a user-friendly advanced powertrain simulation using a combined backward/forward approach. *IEEE Trans. Veh. Technol.* **48**(6), 1751–1761 (1999)

Section VI

UAV Control

Jonathan P. How

Kimon P. Valavanis and George J. Vachtsevanos

UAV Control presents a very comprehensive treatment of UAV control and related technologies. The inherently unstable nature of typical UAV configurations necessitates a rigorous approach to the analysis and design of UAV control technologies, as well as a thorough understanding of stability issues.

► **Linear Flight Control Techniques for Unmanned Aerial Vehicles** by How, Frazzoli, and Chowdhary presents an overview of linear flight control and guidance methods for UAVs, starting with a discussion of rotation matrices and UAV kinematic equations and followed by derivation of the 6 DOF UAV equations of motion. Equations of motion are then linearized, and several linear multi-loop closure techniques for UAV guidance and control are discussed.

► **Nonlinear Flight Control Techniques for Unmanned Aerial Vehicles** by Chowdhary, Frazzoli, How, and Liu discusses nonlinear and adaptive control techniques that are often used to improve UAV performance and reliability. Such techniques are actively being studied to handle nonlinear aerodynamic and kinematic effects, actuator saturations and rate limitations, modeling uncertainty, and time-varying dynamics. An overview of tools and techniques used for designing nonlinear flight controllers is presented along with a summary of Lyapunov stability theory. In detail, the nonlinear control techniques that are considered include gain scheduling, model predictive control, backstepping, dynamic inversion-based control, model reference adaptive control, and model-based fault-tolerant control.

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

► **Adaptive Control of Unmanned Aerial Vehicles: Theory and Flight Tests** by Kannan, Chowdhary, and Johnson focuses on adaptive control of unmanned aircraft that are underactuated systems. A 6 DOF flight control algorithm is derived that can track both position and attitude trajectories. Approximate inverse models for vehicle attitude and position dynamics are used for feedback linearization leading to an inner loop that tracks attitude and angular rate and an outer loop that tracks position and velocity commands. A single adaptive element is used to compensate for inversion errors (uncertainty) in both loops. A key challenge in realizing an adaptive control design on real aircraft is dealing with actuator magnitude and rate saturation. Such saturation elements cannot be easily captured in inverse models and lead to incorrect learning in the adaptive element during periods of saturation. A mechanism to exactly remove such incorrect learning is provided. Additionally, nonlinear reference models are introduced to mitigate the risks of the closed-loop system entering regions of the flight envelope that result in loss of controllability. The resulting adaptive controller accepts trajectory commands comprising of desired position, velocity, attitude, and angular velocity and produces normalized actuator signals required for flight control. A modification to the baseline adaptive control system is also provided that enables long-term retention of the uncertainty approximation within the adaptive element. This architecture is validated through flight tests on several fixed wing and rotorcraft UAVs.

► **Robust and Adaptive Control Methods for Aerial Vehicles** by Lavretsky includes a collection of reliable, efficient, robust, and adaptive control methods for aerial vehicles. It begins with a brief overview of flight dynamics models suitable for flight control design. The first control design method represents the well-understood and now classical linear quadratic regulator (LQR) command tracker, with proportional-integral (PI) feedback connections, serving as the backbone of all other subsequent methods. The main intent is to demonstrate the design of predictable, formally justified, yet numerically efficient flight controllers, with an LQR PI baseline and with a direct model reference adaptive control (MRAC), as an augmentation to the baseline. Through extensive simulation, analysis, and actual flight testing, it is shown that (LQR PI + adaptive) – controllers provide robust stability and maintain tracking performance, when operated in the presence of “unknown unknowns” in the vehicle dynamics and in often “unfriendly” operational environment. All presented control methods are flight tested and validated on a wide range of aerial vehicles.

In summary, this section presents a variety of classical and modern control techniques for unmanned aircraft, which have been successfully tested and implemented.

Linear Flight Control Techniques for Unmanned Aerial Vehicles

27

Jonathan P. How, Emilio Frazzoli, and Girish Vinayak Chowdhary

Contents

| | | |
|--------|---|-----|
| 27.1 | Introduction | 530 |
| 27.2 | Equations of Motion | 531 |
| 27.2.1 | Dynamics | 532 |
| 27.2.2 | Kinematics | 533 |
| 27.2.3 | Forces | 538 |
| 27.2.4 | Linearization of the Equations of Motion | 539 |
| 27.2.5 | Wind Disturbances | 546 |
| 27.2.6 | Simplified Aircraft Equations of Motion for Path Planning | 547 |
| 27.3 | Flight Vehicle Control Using Linear Techniques | 548 |
| 27.3.1 | Standard Performance Specifications | 549 |
| 27.3.2 | Classical Techniques | 551 |
| 27.3.3 | Successive Loop-Closure Examples | 555 |
| 27.3.4 | Multi-input–Multi-output Control | 559 |
| 27.4 | Path Following | 570 |
| 27.5 | Conclusion and Future Directions | 573 |
| | References | 574 |

J.P. How (✉)

Department of Aeronautics and Astronautics, Aerospace Controls Laboratory
Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: jhow@mit.edu

E. Frazzoli

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge,
MA, USA
e-mail: frazzoli@mit.edu

G.V. Chowdhary

Department of Aeronautics and Astronautics Aerospace Controls Laboratory, Massachusetts
Institute of Technology Laboratory for Information and Decision Systems, Cambridge, MA, USA
e-mail: girishc@MIT.EDU

Abstract

This chapter presents an overview of linear flight control and guidance methods for unmanned aerial vehicles (UAVs). The chapter begins with a discussion of rotation matrices and kinematic equations of a UAV. The six degree of freedom UAV equations of motion are also derived using rigid-body dynamics principles. The equations of motion are then linearized, and several linear multi-loop closure techniques for UAV guidance and control are discussed.

27.1 Introduction

A flight control system is required to ensure that the UAV exhibits stable behavior and delivers desired performance such as following a desired trajectory with sufficient accuracy in the presence of external disturbances. Thus, a UAV flight control system is typically safety/mission critical, as an incorrect design, lack of robustness to model variations, or a component failure could result in poor performance (impacting, e.g., payload pointing capability) or even loss of the vehicle.

Figure 27.1 depicts a typical UAV control system. The goal of the control system is to track the desired reference command in presence of external disturbances. The control system achieves this by attempting to eliminate the tracking error between the reference commands and the measured response of the UAV. The control is typically achieved by using estimates of the UAV states, which include position, velocity, angular rate, and attitude of the UAV. These estimates are obtained by the navigation system that fuses noisy measurements from several sensors. The purpose of this chapter, and its companion chapter on nonlinear control (chapter [Nonlinear Flight Control Techniques for Unmanned Aerial Vehicles](#)), is to outline the various methods for designing the control system G_c (and some aspects of the navigation system) and in the process highlight the benefits and challenges of using each approach.

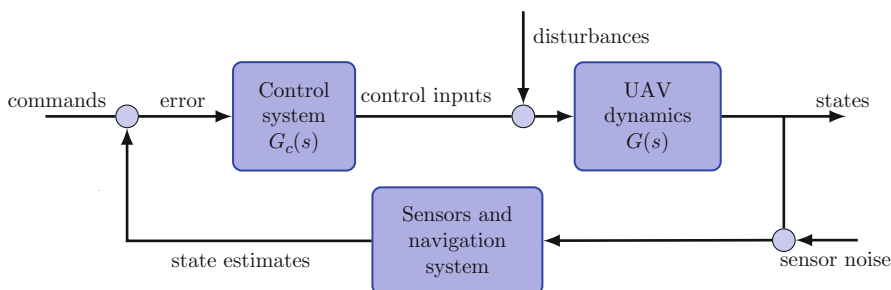


Fig. 27.1 A typical UAV control system architecture. A control system attempts to minimize the tracking error between the desired and estimated states in presence of disturbances. A navigation system provides estimates of the states by fusing together measurements from several complementary sensors

This chapter presents a detailed overview of linear techniques for UAV control. Linear control techniques for manned and unmanned aircraft are well understood and are widely utilized due to their simplicity, ease of implementation, and associated metrics of stability and performance. The chapter begins with a description of UAV dynamics in Sect. 27.2. Linear control techniques are then developed in Sect. 27.3. A brief discussion of path following guidance controllers is presented in Sect. 27.4.

27.2 Equations of Motion

For most problems of interest to UAVs, the Earth can be taken to be an inertial reference frame, and the aircraft can be modeled as a rigid body. In order to write equations of motions for an aircraft, it is customary to define a reference frame (often called a body frame) that is rigidly attached to it (see Fig. 27.2).

A standard choice (often referred to as NED) for the inertial reference frame (X, Y, Z) is to choose the X -axis pointing north, the Y -axis pointing east, and the Z -axis pointing down. Several choices of body frames are common, depending on the task at hand; examples include principal axes and stability axes, and will be discussed in the following. There are advantages to each choice, but the stability axes are very commonly used. In general though, the appropriate body frame (x, y, z) is chosen in such a way that the y -axis is orthogonal to the aircraft's plane of symmetry, pointing to the right. The x -axis is chosen pointing "forward" (the exact direction depends on the definition of the body frame), and the z -axis points "down," to complete the right-handed triad (see Fig. 27.3). Note that for this set, in different *flight equilibrium conditions*, the axes will be oriented differently with respect to the aircraft principal axes, so one must transform (rotate) the principal

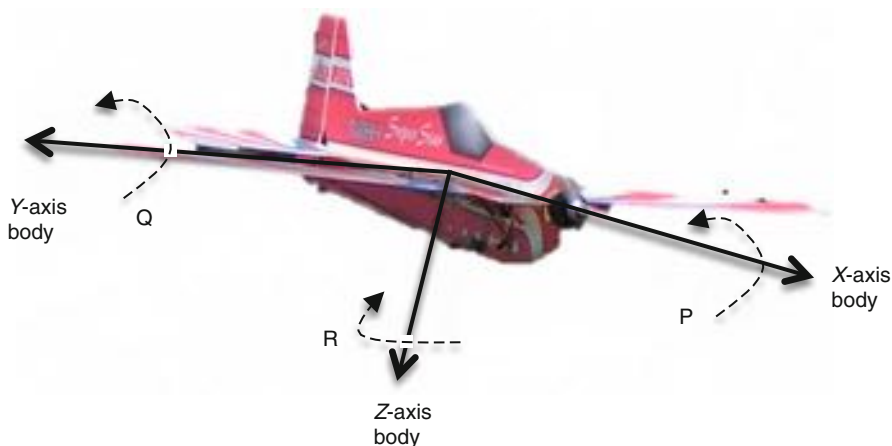


Fig. 27.2 Aircraft body frame

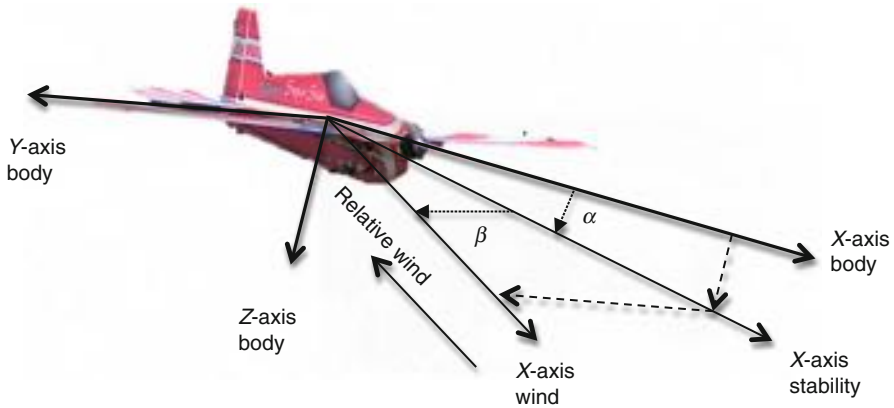


Fig. 27.3 Various body frames used for aircraft analysis

inertia components between the frames. However, when vehicle undergoes motion with respect to the equilibrium, stability axes remain fixed to airplane as if *painted on* (see Fig. 27.3).

27.2.1 Dynamics

Newton's laws of mechanics imply that in a body frame whose origin coincides with the aircraft's center of mass,

$$F_B = m\dot{v}_B + \omega_B \times mv_B, \quad (27.1)$$

$$T_B = J_B\dot{\omega}_B + \omega_B \times J_B\omega_B. \quad (27.2)$$

In the above equations, the subscript $(\cdot)_B$ indicates vectors or tensors expressed in the body frame, the dot indicates differentiation with respect to time, v_B is the velocity of the aircraft's center of mass with respect to the inertial frame, ω_B is the angular velocity of the body frame with respect to the inertial frame, m is the aircraft's mass, and J_B is its inertia tensor, which is constant in the body frame. The operator \times denotes the cross product that arises due to the vector derivative transport theorem (see, e.g., Etkin 1982; Stevens and Lewis 2003; Nelson and Smith 1989). On the left-hand side, F_B denotes the sum of the forces acting on the vehicle (including aerodynamic, gravity, thrust, and buoyancy), and T_B denotes the sum of the moments of these forces about its center of mass.

In coordinates, Eqs. (27.1) and (27.2) are written as follows. Let $F_B = (X, Y, Z)$, $T_B = (\bar{L}, M, N)$, $v_B = (U, V, W)$, and $\omega_B = (P, Q, R)$. The inertia tensor is defined as

$$J_B = \int_{\text{Aircraft}} \rho(s) [(s \cdot s)I - s \otimes s] ds,$$

where $\rho(s)$ indicates the density of the aircraft at the point s in the body frame, I is the identity matrix, \cdot indicates the dot product, and \otimes is the Kronecker product, that is, if $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, then $w = u \otimes v$ is an $m \times n$ matrix with entries $w_{ij} = u_i v_j$. In most cases of interest, aircraft has a plane of symmetry, typically separating the left and right halves. Assuming that the y -axis is orthogonal to the aircraft's plane of symmetry, the inertia tensor will have the following structure:

$$J_B = \begin{bmatrix} J_{xx} & 0 & -J_{xz} \\ 0 & J_{yy} & 0 \\ -J_{xz} & 0 & J_{zz} \end{bmatrix}.$$

(The cross moment of inertia $J_{xz} = \int_{\text{Aircraft}} \rho(x, y, z) xz dx dy dz$ will in general depend on the choice of the body axes and will be zero in case principal axes are chosen.) Substituting into Eqs. (27.1) and (27.2) (see Etkin and Reid 1996; Blakelock 1965; Etkin 1982; Stevens and Lewis 2003; Nelson and Smith 1989; Stengel 2004 for details),

$$\frac{1}{m} F_B \equiv \frac{1}{m} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \dot{U} + QW - RV \\ \dot{V} + RU - PW \\ \dot{W} + PV - QU \end{bmatrix}, \quad (27.3)$$

$$T_B \equiv \begin{bmatrix} \bar{L} \\ M \\ N \end{bmatrix} = \begin{bmatrix} J_{xx}\dot{P} - J_{xz}\dot{R} + QR(J_{zz} - J_{yy}) - PQJ_{xz} \\ J_{yy}\dot{Q} + PR(J_{xx} - J_{zz}) + (P^2 - R^2)J_{xz} \\ J_{zz}\dot{R} - J_{xz}\dot{P} + PQ(J_{yy} - J_{xx}) + QRJ_{xz} \end{bmatrix}. \quad (27.4)$$

27.2.2 Kinematics

Equations (27.1) and (27.2) give the evolution over time of the velocity of the aircraft's center of mass and of the aircraft's angular velocity with respect to the inertial frame (expressed in body frame). In order to reconstruct the position and attitude (i.e., orientation) of the aircraft as a function of time, it is necessary to write equations for the aircraft's kinematics.

Let R_{IB} be the rotation matrix that maps vectors in the body frame to vectors in the inertial frame. Given a vector $u \in \mathbb{R}^3$, define a “hat” operator such that the matrix $(u)^\wedge = \hat{u}$ is the unique matrix satisfying $\hat{u}v = u \times v$, $\forall v \in \mathbb{R}^3$. In coordinates, given $u = (u_1, u_2, u_3)$, one gets

$$\hat{u} = \begin{bmatrix} 0 & -u_3 & u_1 \\ u_3 & 0 & -u_2 \\ -u_1 & u_2 & 0 \end{bmatrix}.$$

The aircraft kinematics can be written as

$$\dot{p} = R_{IB} v_B, \quad (27.5)$$

$$\dot{R}_{IB} = R_{IB} \hat{\omega}_B, \quad (27.6)$$

where p is the position of the aircraft's center of mass. Even though the position and attitude of the aircraft can be completely specified by (27.5) and (27.6), maintaining all components of the rotation matrix R_{IB} may be expensive from the computational point of view, since rotation matrices are not a minimal representation of aircraft attitude. In practice, several other methods to represent an aircraft's attitude are commonly used and will be discussed in the following sections. However, some additional properties of rotation matrices are discussed first.

A first point to note is that the spectrum of any matrix in $SO(3)$ has the form $\text{eig}(R) = \{1, \cos \bar{\theta} \pm i \sin \bar{\theta}\}$. Hence, any matrix R admits a unit eigenvector v such that $Rv = v$. Such an eigenvector identifies a fixed axis for the rotation R . The amplitude of the rotation is given by the angle $\bar{\theta}$. Since the trace of a matrix is the sum of its eigenvalues, it is the case that $\text{Tr}(R) = 1 + 2 \cos \bar{\theta}$. Conversely, given a rotation angle $\bar{\theta}$ and a fixed axis unit vector v , the corresponding rotation is computed through Rodriguez' formula:

$$\text{Rot}(\bar{\theta}, v) = I + \sin \bar{\theta} \hat{v} + (1 - \cos \bar{\theta}) \hat{v}^2. \quad (27.7)$$

The pair $(\bar{\theta}, v)$ is also referred to as the exponential coordinates of a rotation matrix $R = \text{Rot}(\bar{\theta}, v)$, since $\text{Rot}(\bar{\theta}, v) = \exp(\bar{\theta} \hat{v})$.

27.2.2.1 Quaternions

A widely used parametrization of rotations that requires fewer parameters than rotation matrices is based on unit quaternions, that is, unit vectors in \mathbb{R}^4 (Kuipers 2002). A quaternion $q = (\hat{q}, \vec{q})$ is typically written in terms of a scalar part $\hat{q} \in \mathbb{R}$ and a vector part $\vec{q} \in \mathbb{R}^3$. A unit quaternion is such that

$$q \cdot q = \hat{q}^2 + \vec{q} \cdot \vec{q} = 1.$$

The set of unit vectors in \mathbb{R}^4 is denoted as the sphere \mathbb{S}^3 . A unit quaternion $q = (\hat{q}, \vec{q}) \in \mathbb{S}^3$ defines a rotation

$$R = \text{Rot}(2 \arccos \hat{q}, \vec{q} / \|\vec{q}\|). \quad (27.8)$$

In other words, the vector part of a unit quaternion is parallel to the fixed axis of the rotation, and the scalar part determines the rotation angle. Conversely, a rotation matrix $R = \text{Rot}(\bar{\theta}, v)$ can be represented by the unit quaternion

$$q = (\cos(\bar{\theta}/2), \sin(\bar{\theta}/2) v). \quad (27.9)$$

Note that this representation is not unique, since for all $q \in \mathbb{S}^3$, q and $-q$ map to the same rotation matrix. (In other words, \mathbb{S}^3 is a double covering of $SO(3)$).

Quaternions are composed through a multiplication operation, indicated with \circ , defined as follows:

$$q_1 \circ q_2 = (\overset{\circ}{q}_1 \overset{\circ}{q}_2 - \vec{q}_1 \cdot \vec{q}_2, \quad \overset{\circ}{q}_1 \vec{q}_2 + \overset{\circ}{q}_2 \vec{q}_1 - \vec{q}_1 \times \vec{q}_2),$$

or in matrix form

$$q_1 \circ q_2 = \begin{bmatrix} \overset{\circ}{q}_1 & -\vec{q}_1^T \\ \vec{q}_1 & \hat{q} \end{bmatrix} q_2.$$

It can be verified that quaternion multiplication as defined above is consistent with matrix multiplication, in the sense that if R_1 and R_2 are rotation matrices corresponding (according to (27.8)) to the unit quaternions q_1 and q_2 , respectively, then $R_1 R_2$ corresponds to $q_1 \circ q_2$. Also, for any unit quaternion $q = (\overset{\circ}{q}, \vec{q})$, its conjugate quaternion $q^* = (\overset{\circ}{q}, -\vec{q})$ corresponds to the “inverse” rotation, in the sense that $q \circ q^* = q^* \circ q = (1, 0)$.

Finally, quaternion kinematics take the form

$$\dot{q}_{IB} = \frac{1}{2}(0, \omega_I) \circ q_{IB} = \frac{1}{2}q_{IB} \circ (0, \omega_B). \quad (27.10)$$

This equation can be written in matrix form as follows:

$$\dot{q} = -\frac{1}{2}\Omega(\omega_B)q, \quad (27.11)$$

where

$$\Omega(\omega) \equiv \begin{bmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{bmatrix}. \quad (27.12)$$

Quaternions are often the preferred method to model aircraft kinematics, since storing and multiplying quaternions require less memory and computation time than storing and multiplying rotation matrices. On the other hand, the fact that the set of unit quaternions is a double covering of the set of rotation matrices can cause stability and robustness issues (Chaturvedi et al. 2011). Numerical errors in quaternion propagation can result in the quaternion losing its normalization constraint. It is common practice therefore to normalize a quaternion to ensure $\|q\|_2 = 1$ if it is numerically integrated for control or navigation purposes.

27.2.2.2 Euler Angles

A rotation can also be parameterized through a sequence of three elementary rotations about coordinated axes. While many choices are possible, in the field of

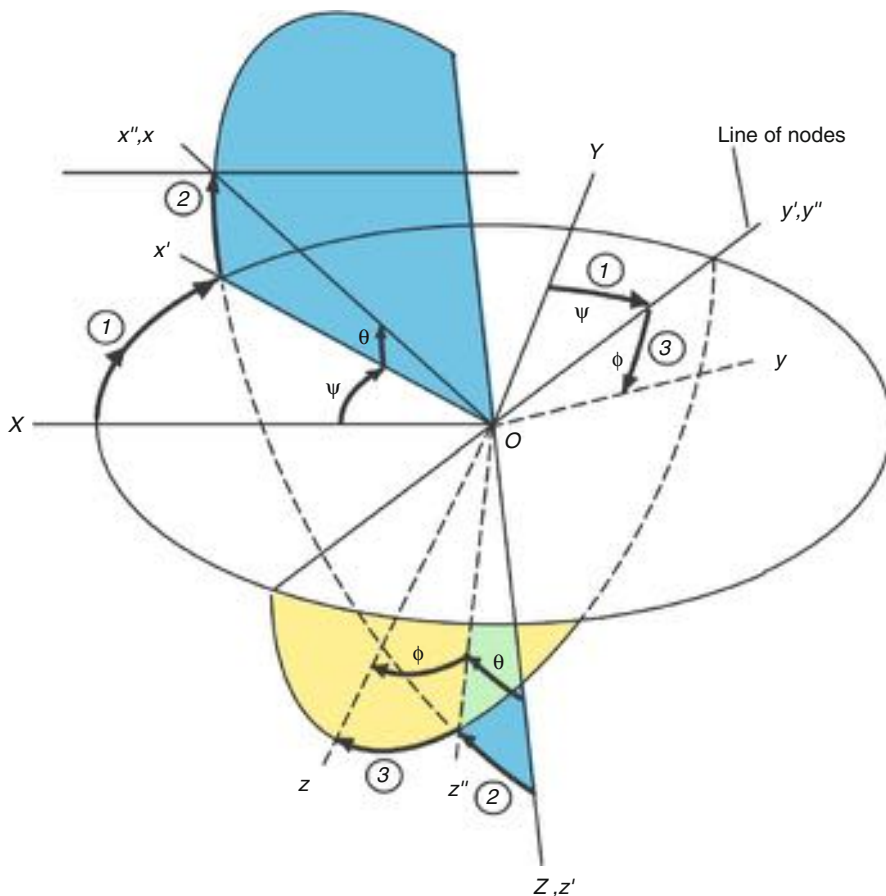


Fig. 27.4 Definition of the three Euler angles used and the associated sequence of rotations

aircraft flight dynamics and control, it is customary to use the “roll,” “pitch,” and “yaw” Euler angles, defined as follows.

Let (X, Y, Z) be unit vectors defining the inertial frame, and let e_1, e_2, e_3 be the columns of the identity matrix. Starting with a body frame coinciding with the inertial frame, rotate it by an angle ψ (yaw) about the Z -axis (coinciding with the third body axis), obtaining a first intermediate body frame given by $R_{IB'} = \text{Rot}(\psi, e_3) = [x', y', z']$ (with $z' = Z$, see Fig. 27.4). Then, rotate this body frame by an angle θ (pitch) about the y' -axis, obtaining a second intermediate body frame given by $R_{IB''} = R_{IB'} \text{Rot}(\theta, e_2) = [x'', y'', z'']$ (with $y'' = y'$). Finally, rotate this body frame by an angle ϕ (roll) about the x'' -axis, obtaining the final body frame $R_{IB} = R_{IB''} \text{Rot}(\phi, e_1) = [x, y, z]$, with $x = x''$. In summary, the rotation matrix R_{IB} is computed as a function of the roll, pitch, and yaw Euler angles (ϕ, θ, ψ) as

$$\begin{aligned}
 R_{IB} &= \text{Rot}(\psi, e_3) \text{Rot}(\theta, e_2) \text{Rot}(\phi, e_1) \\
 &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \phi \sin \theta - \cos \phi \sin \psi & \sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \psi \sin \theta & \cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \phi \cos \theta \end{bmatrix}.
 \end{aligned} \tag{27.13}$$

Conversely, given a rotation matrix R , Euler angles can be obtained as

$$\theta = \arcsin R_{31}, \quad \phi = \arctan \frac{R_{32}}{R_{33}}, \quad \psi = \arctan \frac{R_{21}}{R_{11}},$$

where the 4-quadrant arctangent should be taken for ϕ and ψ .

A direct relationship can also be given between Euler angles and quaternions. Let the components of a quaternion be given by $q = [\overset{\circ}{q}, q_x, q_y, q_z]$; then

$$\begin{aligned}
 \theta &= -\arcsin 2(q_x q_z - \overset{\circ}{q} q_y), \\
 \phi &= \arctan 2 \left[2(q_y q_z + \overset{\circ}{q} q_x), 1 - 2(q_x^2 + q_y^2) \right], \\
 \psi &= \arctan 2 \left[2(\vec{q}_x q_y + \overset{\circ}{q} q_z), 1 - 2(q_y^2 + q_z^2) \right],
 \end{aligned} \tag{27.14}$$

and

$$\overset{\circ}{q} = \pm (\cos(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \sin(\psi/2)), \tag{27.15}$$

$$q_x = \pm (\sin(\phi/2) \cos(\theta/2) \cos(\psi/2) - \cos(\phi/2) \sin(\theta/2) \sin(\psi/2)), \tag{27.16}$$

$$q_y = \pm (\cos(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \sin(\psi/2)), \tag{27.17}$$

$$q_z = \pm (\cos(\phi/2) \cos(\theta/2) \sin(\psi/2) - \sin(\phi/2) \sin(\theta/2) \cos(\psi/2)), \tag{27.18}$$

where the sign is arbitrary but must be consistent.

The kinematics of the Euler angles can be obtained by differentiating (27.13) with respect to time and recalling (27.6). After simplification, one obtains

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix}. \tag{27.19}$$

Singularities are a potential problem for an Euler angle representation of the attitude kinematics. In the case of roll, pitch, and yaw Euler angles, the singularities at $\theta = \pm 90^\circ$ can cause computational difficulties. Hence, the recommended practice, especially for agile vehicles, such as fighter aircraft or small UAVs, that perform large-angle attitude maneuvers, is to use quaternions (or rotation matrices) to model the aircraft's kinematics. On the other hand, since Euler angles may be

more intuitive to work with, it is common practice to use them as a front end for humans, for example, to specify initial conditions and to present simulation or flight test results.

Equations (27.3)–(27.5) and (27.11)–(27.14) can be combined to form the nonlinear rigid-body equations of motion for a UAV. The next logical step is to discuss how the forces and moments acting on the UAV are generated. The full equations of motion will then be linearized to develop models for the linear control design.

27.2.3 Forces

Forces and torques acting on a UAV are from a variety of sources, including aerodynamic, gravity, and thrust. In general, forces and torques will depend on the aircraft's position, attitude, on its linear and angular speed with respect to the surrounding air, and on the control settings.

The aerodynamic forces are typically resolved into two components, lift (L) and drag (D) (see Fig. 27.5). The aerodynamic lift force is perpendicular to the relative wind vector, while the drag force resists the vehicle motion along the relative wind. The direction of the aircraft's velocity relative to the wind with respect to the body frame is expressed by two angles, which can be thought of as spherical coordinates. The sideslip angle β is the angle formed by the aircraft's velocity with the aircraft's symmetry plane. The angle of attack α is the angle formed by the projection of the aircraft velocity on the aircraft's symmetry plane with the x body axis.

Aerodynamic forces primarily depend on the angle of attack α , on the sideslip β , and on the dynamic pressure $\bar{Q} = \frac{1}{2}\rho V_T^2$, where ρ is the air density, which depends

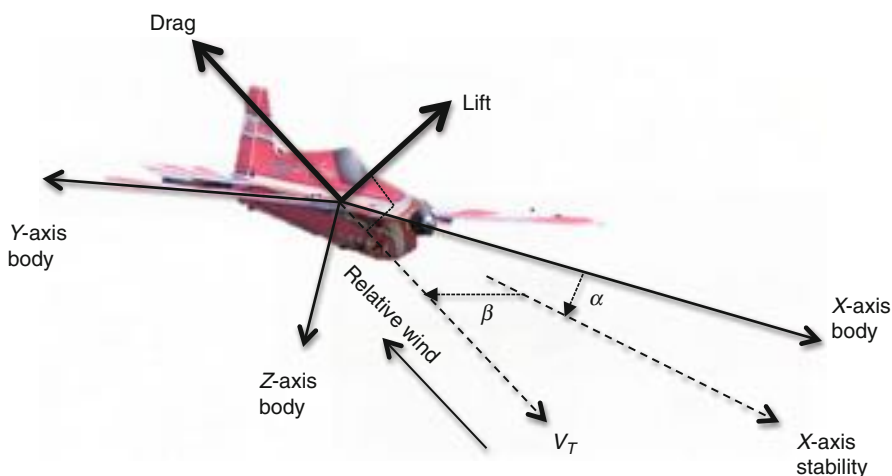


Fig. 27.5 Lift and drag acting on the aircraft

on the altitude, and V_T is the free-stream airspeed of the aircraft. The lift and drag forces on the body will be in the plane of the x - and z -axes of the wind axis system shown in Fig. 27.5. These can be rotated into the body frame using a rotation matrix that is a function of the angles α and β :

$$R_{BW} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix}. \quad (27.20)$$

In this case, the aerodynamic forces in the body frame can be written in terms of lift and drag (assuming no side forces):

$$(F_{\text{aero}})_B = \begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = R_{BW} \begin{bmatrix} -D \\ 0 \\ -L \end{bmatrix}. \quad (27.21)$$

Propulsive forces are typically contained within the aircraft's symmetry plane, possibly with a small angular offset α_T with respect to the x body axis, yielding

$$(F_{\text{prop}})_B = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = \|F_{\text{prop}}\| \begin{bmatrix} \cos \alpha_T \\ 0 \\ -\sin \alpha_T \end{bmatrix}. \quad (27.22)$$

Given the usual (NED) choice of inertial axis, wherein the z inertial axis points “down,” the forces due to gravity can be written as

$$(F_{\text{gravity}})_B = \begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix} = mg R_{IB}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = mg \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix}, \quad (27.23)$$

where g is the gravity acceleration.

27.2.4 Linearization of the Equations of Motion

The right-hand side of Eqs. (27.3) and (27.4) are nonlinear and typically far more complicated than can be addressed by standard control techniques and are more complicated than necessary for most flight regimes. Furthermore, the left-hand side, which specify the total forces and moments acting on the vehicle are even more complicated. Thus, numerous simplifications have been developed.

27.2.4.1 Relative Equilibria

The standard approach is to assume that the vehicle is flying in a *relative equilibrium condition* and then linearize the equations of motion about this nominal

flight condition. In the case of aircraft dynamics, a relative equilibrium is defined as a “steady-state” trajectory along which $\dot{\omega}_B = 0$ and $\dot{v}_B = 0$, and the control inputs are maintained fixed or “trimmed” to some value δ_0 . Clearly, such a trajectory will be such that the linear and angular velocities in body frame are constant. These nominal values of the linear and angular velocities will be denoted by v_0 and ω_0 , respectively.

At equilibrium conditions, after writing out explicitly the dependency of the forces and moments, Eqs. (27.1) and (27.2) take the form

$$F_B(p, R_{IB}, v_0, \omega_0, \delta_0) = \omega_0 \times m v_0, \quad (27.24)$$

$$T_B(p, R_{IB}, v_0, \omega_0, \delta_0) = \omega_0 \times J_B \omega_0. \quad (27.25)$$

Note that the position p and attitude R change along the trajectory, according to the kinematics equations $\dot{p} = R_{IB} v_0$ and $\dot{R} = R_{IB} \hat{\omega}_0$. For Eqs. (27.24) and (27.25) to hold over time, it is necessary that the left-hand sides do not change along the trajectory, even though the aircraft’s position and attitude change. In the case of aircraft dynamics, and over small vertical excursions (i.e., neglecting the dependency of the air density on altitude), it can be safely assumed that the body forces and moments are invariant to translations and to rotations about a vertical (inertial) axis, that is,

$$F_B(p, R_{IB}, v_0, \omega_0, \delta_0) = F_B(p + \Delta p, \text{Rot}(\Delta\psi, e_3) R_{IB}, v_0, \omega_0, \delta_0),$$

$$T_B(p, R_{IB}, v_0, \omega_0, \delta_0) = T_B(p + \Delta p, \text{Rot}(\Delta\psi, e_3) R_{IB}, v_0, \omega_0, \delta_0),$$

for any translation Δp and for any heading rotation $\Delta\psi$.

In order for Eqs. (27.24) and (27.25) to hold throughout the trajectory and hence for the trim condition $(v_0, \omega_0, \delta_0)$ to be a valid relative equilibrium, it is necessary that $R_{IB} \hat{\omega}_0 = [0, 0, \dot{\psi}_0]^T$ for some constant heading rate $\dot{\psi}_0$. In other words, in the inertial frame, trajectories corresponding to relative equilibria take the form of circular helices with a vertical axis, flown at constant speed, angle of attack, and sideslip angle. In the special case in which $\dot{\psi}_0 = 0$, that is, on relative equilibria corresponding to straight-line motion, the total forces and moments are zero – but this is not true for the general case.

When analyzing the dynamics of an aircraft near a relative equilibrium, it is convenient to choose the body frame in such a way that the angle of attack is zero at equilibrium. In other words, the x body axis is chosen in such a way that the relative wind direction is in the (x, y) body plane at equilibrium. Such a body frame is called the stability frame, due to its importance in the aircraft stability analysis. Notice that the stability frame depends on the particular relative equilibrium under consideration; for different flight conditions, stability frames will in general be different.

In the following, a class of “symmetric” relative equilibria are considered, along which the sideslip, roll, and yaw angle are zero. To proceed with the linearization of the dynamics for various flight conditions, define the perturbed velocity and accelerations in terms of the nominal values, as given in Table 27.1. The assumption

Table 27.1 Definition of the perturbation variables about the equilibrium condition

| | Nominal velocity | Perturbed velocity | Perturbed acceleration |
|---------------|------------------|------------------------------|-------------------------------|
| Velocities | U_0 | $U = U_0 + u$ | $\dot{U} = \dot{u}$ |
| | $W_0 = 0$ | $W = w$ | $\dot{W} = \dot{w}$ |
| | $V_0 = 0$ | $V = v$ | $\dot{V} = \dot{v}$ |
| Angular rates | $P_0 = 0$ | $P = p$ | $\dot{P} = \dot{p}$ |
| | $Q_0 = 0$ | $Q = q$ | $\dot{Q} = \dot{q}$ |
| | $R_0 = 0$ | $R = r$ | $\dot{R} = \dot{r}$ |
| Angles | Θ_0 | $\Theta = \Theta_0 + \theta$ | $\dot{\Theta} = \dot{\theta}$ |
| | $\Phi_0 = 0$ | $\Phi = \phi$ | $\dot{\Phi} = \dot{\phi}$ |
| | $\Psi_0 = 0$ | $\Psi = \psi$ | $\dot{\Psi} = \dot{\psi}$ |

herein is that the perturbations to the nominal are much smaller in magnitude than the nominal (i.e., $|u| \ll |U_0|$).

A key element of the complexity here is that typically the aerodynamic, thrust, and gravity forces and moments will also be perturbed by the perturbed motion of the vehicle, which are denoted in the following as $\Delta X, \dots, \Delta N$:

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} + rU_0 \\ \dot{w} - qU_0 \end{bmatrix} \quad (27.26)$$

$$\begin{bmatrix} \Delta \bar{L} \\ \Delta M \\ \Delta N \end{bmatrix} = \begin{bmatrix} J_{xx}\dot{p} - J_{xz}\dot{r} \\ J_{yy}\dot{q} \\ J_{zz}\dot{r} - J_{xz}\dot{p} \end{bmatrix} \quad (27.27)$$

where we use the $\overline{(\cdot)}$ to distinguish between the lift L and the moment \bar{L} . The key aerodynamic parameters are also perturbed:

$$\text{Total velocity} \quad V_T = ((U_0 + u)^2 + v^2 + w^2)^{1/2} \approx U_0 + u \quad (27.28)$$

$$\text{Perturbed sideslip angle} \quad \beta = \sin^{-1}(v/V_T) \approx v/U_0 \quad (27.29)$$

$$\text{Perturbed angle of attack} \quad \alpha_x = \tan^{-1}(w/U) \approx w/U_0 \quad (27.30)$$

27.2.4.2 Stability Derivatives

To develop the equations of motion further, the terms $\Delta X \dots \Delta N$ must be investigated. Recall that at equilibrium, the net forces and moments must be zero. But since the aerodynamic and gravity forces are a function of equilibrium condition and the perturbations about this equilibrium, in general, it is very difficult to determine the exact nature of these aerodynamic perturbations. Thus, the standard approach (see detailed discussions in Etkin (1982), Etkin and Reid (1996), and Nelson and Smith (1989)) is to try to predict the changes in the aerodynamic forces and moments using a first-order expansion in the key flight parameters:

$$\Delta X = \frac{\partial X}{\partial U} \Delta U + \frac{\partial X}{\partial W} \Delta W + \frac{\partial X}{\partial \dot{W}} \Delta \dot{W} + \frac{\partial X}{\partial \Theta} \Delta \Theta + \dots + \frac{\partial X_g}{\partial \Theta} \Delta \Theta + \Delta X^c \quad (27.31)$$

$$= \frac{\partial X}{\partial U} u + \frac{\partial X}{\partial W} w + \frac{\partial X}{\partial \dot{W}} \dot{w} + \frac{\partial X}{\partial \Theta} \theta + \dots + \frac{\partial X_g}{\partial \Theta} \theta + \Delta X^c, \quad (27.32)$$

where $\frac{\partial X}{\partial U}$ is called a stability derivative, which is evaluated at the equilibrium condition. Note that both dimensional and nondimensional forms are used. Clearly, this is an approximation since it ignores any time lags in aerodynamics forces (assumes that forces are only functions of instantaneous values).

As before, X_g in Eq. (27.32) corresponds to the X body force component due to gravity and the perturbation gravity and thrust forces and moments. Assuming $\Phi_0 = 0$, then

$$\left. \frac{\partial X_g}{\partial \Theta} \right|_0 = -mg \cos \Theta_0 \quad \left. \frac{\partial Z_g}{\partial \Theta} \right|_0 = -mg \sin \Theta_0.$$

Also ΔX^c denotes the perturbations due to the control actuators (e.g., rudder, ailerons, elevators, thrust).

While Eq. (27.32) leads to simplified force (and moment) perturbations, it is still clear that the linearized expansion can involve many terms $u, \dot{u}, \ddot{u}, \dots, w, \dot{w}, \ddot{w}, \dots$. Thus, it is typical to only retain a few terms to capture the dominant effects. For symmetric aircraft, this dominant behavior is most easily discussed in terms of the *symmetric* variables U, W, Q and forces/torques X, Z , and M , and the *asymmetric* variables V, P, R , and forces/torques Y, \bar{L} , and N (Table 27.2).

Furthermore, for most flight conditions, further simplifications can often be made. For example, for truly symmetric flight, Y, \bar{L} , and N will be exactly *zero* for any value of U, W, Q . So the derivatives of asymmetric forces/torques with respect to the symmetric motion variables are *zero*. Also, the derivatives of symmetric forces/torques with respect to the asymmetric motion variables are small and can often be neglected. Often derivatives with respect to the derivatives of the motion variables can also be neglected, but $\partial Z / \partial \dot{w}$ and $M_{\dot{w}} \equiv \partial M / \partial \dot{w}$ (aerodynamic lag involved in forming new pressure distribution on the wing in response to the perturbed angle of attack) should not be neglected. Also, $\partial X / \partial q$ is often negligibly small. A summary of the effects of the aerodynamic perturbations is as follows:

- (1) $\Delta X = \left(\frac{\partial X}{\partial U} \right)_0 u + \left(\frac{\partial X}{\partial W} \right)_0 w \Rightarrow \Delta X \sim u, \alpha_x \approx w / U_0$
- (2) $\Delta Y \sim \beta \approx v / U_0, p, r$
- (3) $\Delta Z \sim u, \alpha_x \approx w / U_0, \dot{\alpha}_x \approx \dot{w} / U_0, q$
- (4) $\Delta \bar{L} \sim \beta \approx v / U_0, p, r$
- (5) $\Delta M \sim u, \alpha_x \approx w / U_0, \dot{\alpha}_x \approx \dot{w} / U_0, q$
- (6) $\Delta N \sim \beta \approx v / U_0, p, r$

The result is that, with these force and torque approximations, Eqs. (27.1), (27.3), and (27.5) decouple from (27.2), (27.4), and (27.6). In particular, Eqs. (27.1), (27.3),

Table 27.2 Standard stability derivatives for a typical aircraft planform showing that numerous terms are 0 or small. The other nonzero terms, denoted as •, must be computed using the methods described in Etkin and Reid (1996) and Nelson and Smith (1989)

| $\partial()/\partial()$ | X | Y | Z | \bar{L} | M | N |
|-------------------------|-------------|-----|-----|-----------|-----|-----|
| u | • | 0 | • | 0 | • | 0 |
| v | 0 | • | 0 | • | 0 | • |
| w | • | 0 | • | 0 | • | 0 |
| p | 0 | • | 0 | • | 0 | • |
| q | ≈ 0 | 0 | • | 0 | • | 0 |
| r | 0 | • | 0 | • | 0 | • |

and (27.5) are the longitudinal dynamics in u , w , and q :

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} m\dot{u} \\ m(\dot{w} - qU_0) \\ J_{yy}\dot{q} \end{bmatrix} \quad (27.33)$$

$$\approx \begin{bmatrix} \left(\frac{\partial X}{\partial U}\right)_0 u + \left(\frac{\partial X}{\partial W}\right)_0 w + \left(\frac{\partial X_g}{\partial \Theta}\right)_0 \theta + \Delta X^c \\ \left(\frac{\partial Z}{\partial U}\right)_0 u + \left(\frac{\partial Z}{\partial W}\right)_0 w + \left(\frac{\partial Z}{\partial \dot{W}}\right)_0 \dot{w} + \left(\frac{\partial Z}{\partial Q}\right)_0 q + \left(\frac{\partial Z_g}{\partial \Theta}\right)_0 \theta + \Delta Z^c \\ \left(\frac{\partial M}{\partial U}\right)_0 u + \left(\frac{\partial M}{\partial W}\right)_0 w + \left(\frac{\partial M}{\partial \dot{W}}\right)_0 \dot{w} + \left(\frac{\partial M}{\partial Q}\right)_0 q + \Delta M^c \end{bmatrix}, \quad (27.34)$$

and Eqs. (27.2), (27.4), and (27.6) are the lateral dynamics in v , p , and r :

$$\begin{bmatrix} \Delta Y \\ \Delta \bar{L} \\ \Delta N \end{bmatrix} = \begin{bmatrix} m(\dot{v} + rU_0) \\ J_{xx}\dot{p} - J_{xz}\dot{r} \\ J_{zz}\dot{r} - J_{xz}\dot{p} \end{bmatrix} \quad (27.35)$$

$$\approx \begin{bmatrix} \left(\frac{\partial Y}{\partial V}\right)_0 v + \left(\frac{\partial Y}{\partial P}\right)_0 p + \left(\frac{\partial Y}{\partial R}\right)_0 r + \Delta Y^c \\ \left(\frac{\partial \bar{L}}{\partial V}\right)_0 v + \left(\frac{\partial \bar{L}}{\partial P}\right)_0 p + \left(\frac{\partial \bar{L}}{\partial R}\right)_0 r + \Delta \bar{L}^c \\ \left(\frac{\partial N}{\partial V}\right)_0 v + \left(\frac{\partial N}{\partial P}\right)_0 p + \left(\frac{\partial N}{\partial R}\right)_0 r + \Delta N^c \end{bmatrix}. \quad (27.36)$$

27.2.4.3 Actuators

The primary actuators in the longitudinal direction are the elevators and thrust. Clearly the thrusters and elevators play a key role in defining the steady-state and equilibrium flight condition. The focus now is on determining how they also influence the aircraft motion about this equilibrium condition. For example, if the elevator is deflected, a perturbation in the vehicle motion would be expected, as captured by $u(t)$, $w(t)$, $q(t)$.

Recall that ΔX^c is the perturbation in the total force in the X direction as a result of the actuator commands, that is, a force change due to an actuator deflection

from trim. As before, one can approximate these aerodynamic terms using the same perturbation approach:

$$\Delta X^c = X_{\delta_{th}} \delta_{th} + X_{\delta_e} \delta_e, \quad (27.37)$$

where δ_e is the deflection of the elevator from trim (down positive), δ_{th} is the change in thrust, and X_{δ_e} and $X_{\delta_{th}}$ are the control stability derivatives. This results in

$$\begin{bmatrix} \Delta X^c \\ \Delta Z^c \\ \Delta M^c \end{bmatrix} = \begin{bmatrix} X_{\delta_{th}} & X_{\delta_e} \\ Z_{\delta_{th}} & Z_{\delta_e} \\ M_{\delta_{th}} & M_{\delta_e} \end{bmatrix} \begin{bmatrix} \delta_{th} \\ \delta_e \end{bmatrix}. \quad (27.38)$$

A similar process can be performed in the lateral direction for the rudder and ailerons.

Equations of Motion

Combining the results of the previous sections for the longitudinal dynamics yields the equations of motion:

$$m\dot{u} = X_u u + X_w w - mg \cos \Theta_0 \theta + \Delta X^c \quad (27.39)$$

$$m(\dot{w} - qU_0) = Z_u u + Z_w w + Z_{\dot{w}} \dot{w} + Z_q q - mg \sin \Theta_0 \theta + \Delta Z^c \quad (27.40)$$

$$J_{yy} \dot{q} = M_u u + M_w w + M_{\dot{w}} \dot{w} + M_q q + \Delta M^c. \quad (27.41)$$

If there is no roll/yaw motion so that $q = \dot{\theta}$ and $M_{\dot{w}}$ and $Z_{\dot{w}}$ are assumed to be small, these can be rewritten in state space form using $\alpha = w/U_0$ as

$$\begin{bmatrix} m\dot{u} \\ m\dot{\alpha} \\ J_{yy}\dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -mg \cos \Theta_0 \\ Z_u/U_0 & Z_\alpha/U_0 & Z_q/U_0 + m & -mg/U_0 \sin \Theta_0 \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta X^c \\ \Delta Z^c \\ \Delta M^c \\ 0 \end{bmatrix} \quad (27.42)$$

or equivalently as $\dot{x} = Ax + Bu$ where

$$A = \begin{bmatrix} \frac{X_u}{m} & \frac{X_\alpha}{m} & 0 & -g \cos \Theta_0 \\ \frac{Z_u}{mU_0} & \frac{Z_\alpha}{mU_0} & \frac{Z_q}{mU_0} + \frac{g}{U_0} \sin \Theta_0 & \\ \frac{M_u}{J_{yy}} & \frac{M_\alpha}{J_{yy}} & \frac{M_q}{J_{yy}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \frac{X_{\delta_{th}}}{m} & \frac{X_{\delta_e}}{m} \\ \frac{Z_{\delta_{th}}}{mU_0} & \frac{Z_{\delta_e}}{mU_0} \\ \frac{M_{\delta_{th}}}{J_{yy}} & \frac{M_{\delta_e}}{J_{yy}} \\ 0 & 0 \end{bmatrix}. \quad (27.43)$$

Note there are significant notational simplifications if the stability derivatives are defined to include m and J_{yy} .

Using a procedure similar to the longitudinal case, the equations of motion for the lateral dynamics can be developed using the state vector:

$$x = \begin{bmatrix} \phi \\ \beta \\ p \\ r \end{bmatrix} \quad \text{with inputs} \quad u = \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix},$$

where $\dot{\psi} = r \sec \Theta_0$

$$A = \begin{bmatrix} 0 & 0 & 1 & \tan \theta_0 \\ \frac{g}{U_0} \cos \theta_0 & \frac{Y_\beta}{mU_0} & \frac{Y_p}{mU_0} & \frac{Y_r}{mU_0} - 1 \\ 0 & \left(\frac{\bar{L}_\beta}{J'_{xx}} + J'_{zx} N_\beta \right) & \left(\frac{\bar{L}_p}{J'_{xx}} + J'_{zx} N_p \right) & \left(\frac{\bar{L}_r}{J'_{xx}} + J'_{zx} N_r \right) \\ 0 & \left(J'_{zx} \bar{L}_\beta + \frac{N_\beta}{J'_{zz}} \right) & \left(J'_{zx} \bar{L}_p + \frac{N_p}{J'_{zz}} \right) & \left(J'_{zx} \bar{L}_r + \frac{N_r}{J'_{zz}} \right) \end{bmatrix} \quad (27.44)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ (mU_0)^{-1} & 0 & 0 \\ 0 & (J'_{xx})^{-1} & J'_{zx} \\ 0 & J'_{zx} & (J'_{zz})^{-1} \end{bmatrix} \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ \bar{L}_{\delta_a} & \bar{L}_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \quad (27.45)$$

and

$$\begin{aligned} J'_{xx} &= (J_{xx}J_{zz} - J_{zx}^2)/J_{zz} \\ J'_{zz} &= (J_{xx}J_{zz} - J_{zx}^2)/J_{xx} \\ J'_{zx} &= J_{zx}/(J_{xx}J_{zz} - J_{zx}^2). \end{aligned} \quad (27.46)$$

Thus, in either case, the aircraft equations of motion can be written as

$$\dot{x} = Ax + Bu \quad (27.47)$$

$$y = Cx + Du, \quad (27.48)$$

where typically $D = 0$, but this depends on the sensor and actuator selection. One can also convert the dynamics into transfer function form (for each sensor/actuator pair) $y_i = G_{ij}(s)u_j$, $G_{ij}(s) = C_i(sI - A)^{-1}B_j + D_{ij}$ that can be used in the classical control design process.

27.2.5 Wind Disturbances

In the presence of wind, the atmosphere is moving relative to the Earth frame (Nelson and Smith 1989; Stevens and Lewis 2003). Therefore, the equations of motion must be modified to account for the fact that the aerodynamic force and moments are functions of the relative motion between the aircraft and the atmosphere and not the inertial velocities. Thus, we must differentiate between the inertial velocity of the vehicle v_i and the velocity of the vehicle with respect to the air mass v_a resulting from wind effects W_g

$$v_a = v_i - R_{BI} W_g. \quad (27.49)$$

In this case, v_a should be used to calculate the aerodynamic forces and moments rather than v_i , and this quantity can be reliably estimated using an airspeed measurement device. More specifically, in the X direction, let u be the aircraft perturbation speed and u_g be the gust speed in that direction, then the aircraft speed with respect to the atmosphere is $u_a = u - u_g$. In this case, the linearization of the aerodynamic forces and moments in Eq. (27.32) should be rewritten in terms of the perturbations relative to the atmosphere:

$$\begin{aligned} \Delta X = & \frac{\partial X}{\partial U}(u - u_g) + \frac{\partial X}{\partial W}(w - w_g) + \frac{\partial X}{\partial \dot{W}}(\dot{w} - \dot{w}_g) + \frac{\partial X}{\partial Q}(q - q_g) \\ & + \dots + \frac{\partial X}{\partial \Theta}\theta + \dots + \frac{\partial X_g}{\partial \Theta}\theta + \Delta X^c \end{aligned} \quad (27.50)$$

but note that the gravity and control input terms remain the same. The rotational aspects of the gusts are caused by spatial variations in the gust components, so that $p_g = \frac{\partial w_g}{\partial y}$ and $q_g = \frac{\partial w_g}{\partial x}$. Repeating the linearization process outlined before leads to a new input term

$$\dot{x} = Ax + Bu + B_w w, \quad (27.51)$$

where, for example, in the longitudinal case,

$$B_w = \begin{bmatrix} -\frac{X_u}{m} & -\frac{X_\alpha}{m} & 0 \\ -\frac{Z_u}{mU_0} & -\frac{Z_\alpha}{mU_0} & -\frac{Z_q}{mU_0} \\ -\frac{M_u}{J_{yy}} & -\frac{M_\alpha}{J_{yy}} & -\frac{M_q}{J_{yy}} \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad w = \begin{bmatrix} u_g \\ \alpha_g \\ q_g \end{bmatrix}, \quad (27.52)$$

and a similar operation can be performed for the lateral dynamics in terms of the disturbance inputs β_g , p_g , and r_g .

The input w can now be used to model the effects of various steady and stochastic wind models, as well as wind shear. Numerous stochastic models have been developed for studying the effect of winds on the aircraft dynamics and performance (Moorhouse and Woodcock 1982; Tatom and Smith 1981; Stevens and Lewis 2003), but these must be used with care when applied to the low altitude operations expected for most small-scale UAV flights.

27.2.6 Simplified Aircraft Equations of Motion for Path Planning

A simplified set of aircraft equations of motion can be derived for path planning purposes. In deriving these equations, the aircraft roll rate, pitch rate, and yaw rate dynamics are ignored and are replaced using kinematic approximations. The resulting model describes the motion of a rigid point mass with kinematic path constraints. For the purpose of this model, the position of the aircraft in an inertial frame whose x -axis is parallel to the local horizon is denoted by x_e, y_e, z_e .

The flight path angle γ denotes the angle between the local horizon and the velocity vector of the aircraft V_T (see Fig. 27.6). The heading angle ψ is the angle between V_T and the z -axis of the local inertial frame. The bank angle ϕ is the angle that the aircraft is banked about the velocity vector V_T . The forces acting on the aircraft consist of the weight mg , thrust T , lift L , and drag D . The equations for a point mass model of a fixed wing aircraft can then be formulated by assuming small γ and ψ (Zhao and Tsiotras 2010; Etkin and Reid 1996):

$$\dot{x}_e = V_T \cos \gamma \cos \psi, \quad (27.53)$$

$$\dot{y}_e = V_T \cos \gamma \sin \psi, \quad (27.54)$$

$$\dot{z}_e = -V_T \sin \gamma, \quad (27.55)$$

$$\dot{V}_T = \frac{1}{m} [T - D - mg \sin \gamma], \quad (27.56)$$

$$\dot{\gamma} = \frac{1}{m V_T} [L \cos \phi - mg \cos \gamma], \quad (27.57)$$

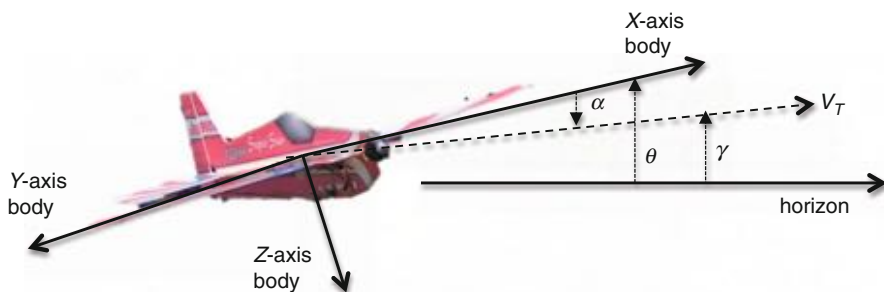


Fig. 27.6 Definition of the flight path angle

$$\dot{\psi} = \frac{L \sin \phi}{m V_T \cos \gamma}, \quad (27.58)$$

and the lift and drag of the aircraft can be approximated as

$$L = \overline{Q} S C_L \quad \text{and} \quad D = \overline{Q} S (C_{D_0} + K C_L^2), \quad (27.59)$$

where S is the wing surface area, C_L is the lift coefficient, C_{D_0} is the parasitic drag coefficient, and K is a constant dependent on the aircraft wing geometry.

27.3 Flight Vehicle Control Using Linear Techniques

As presented above, even in the simplest case of decoupled linearized dynamics, the equations of motion of an aircraft are quite complicated, with both the longitudinal and lateral dynamics being described by a fourth-order system with multiple actuator inputs and sensor outputs y . Furthermore, the speed of the natural response of the system associated with the pole locations of the open-loop dynamics are typically inconsistent with fast and/or stable response to commands, such as waypoint, trajectory tracking, and agile maneuvers. Finally, the dynamics of several UAVs can be inherently unstable, typical examples include rotorcraft such as helicopters, quadrotors, and ducted fans (see, e.g., Mettler 2003; Chowdhary and Jategaonkar 2009). Thus, feedback control is required to stabilize and/or speed up the response.

Given the complexity of the dynamics, there are two basic strategies for the control design. The first is to continue the decomposition in the previous section to identify components of the dynamics that are well controlled by specific choices of the actuators and then perform *successive loop closure* (see, e.g., Lawrence et al. 2008; Johnson and Kannan 2005). In this case, the loops are nested by arranging that the reference commands for the inner loop are provided by the outer-loop controller. An example of this is shown in Fig. 27.13 in which the outermost position control loop provides desired velocity commands using path following guidance discussed in Sect. 27.4. The outer velocity control loop provides a reference (in this case, a desired quaternion value) for the inner attitude control loop. One key advantage of this approach is that it leads to a natural mechanism of handling limits on flights variables (e.g., such as bank or pitch angles) and actuator inputs because the reference commands can be saturated before being passed to the inner loop. Each step of the control design process is simpler, but the nesting of the control loops leads to some challenges. The general rule of thumb is to ensure that the inner control loops result in “fast” dynamics, and then each successive loop added is “slower than the previous one.” The primary difficulties here are to determine what is meant by fast and slow and how to determine if there is too much interaction between the inner/outer loops being closed (e.g., closing the outer loop might reduce the performance of the inner loop, requiring a redesign).

The second approach is to design a controller for the full dynamics – either linear or nonlinear. The advantage of this approach is that it employs the power of state space control approaches to handle the fully coupled dynamics. However, it is difficult to handle actuator saturation and very hard to include state constraints. Furthermore, unless done with extreme care, these controllers, especially in high performance flight, can be very sensitive to modeling errors and omissions.

Having determined the architecture, the next step in any control design is to determine the dynamics of the system of interest (i.e., the full set of dynamics or the approximate inner loop dynamics). Having identified this, one should then determine the requirements and the extent to which the dynamics meet these goals. For example, there may be requirements on certain frequency (to ensure the dynamics are “fast”) and damping (to ensure that the oscillations die out quickly) specifications on the pole locations. There may also be requirements on the maximum steady tracking error to a step command input. Since the open-loop dynamics of the vehicle rarely satisfy these requirements, the typical approach is to use linear feedback control to modify the pole locations and loop gains.

27.3.1 Standard Performance Specifications

Figure 27.7 shows a standard feedback control loop, where a typical performance metric, the tracking error, can be written as

$$e = r - (y + v) = S(r - d_o - v) - SGd_i$$

yielding $y = T(r - v) + Sd_o + SGd_i$ with $L = GG_c$, $S = (I + L)^{-1}$, $T = L(I + L)^{-1}$. So good tracking performance of $r(t)$ (signal typically has low-frequency content) requires e be small, which translates to $\|S(j\omega)\|$ being small at low frequencies, that is, $0 \leq \omega \leq \omega_l$. Furthermore, to reduce the impact of sensor noise v (which typically has high-frequency content) requires $\|T(j\omega)\|$ be small for all $\omega \geq \omega_h$. Since $T(s) + S(s) = I \forall s$, one cannot make both

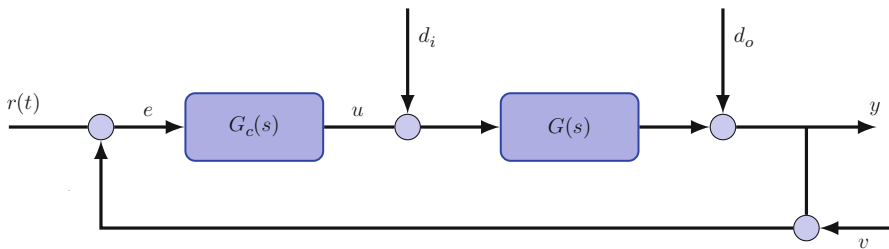


Fig. 27.7 Classical control feedback loop

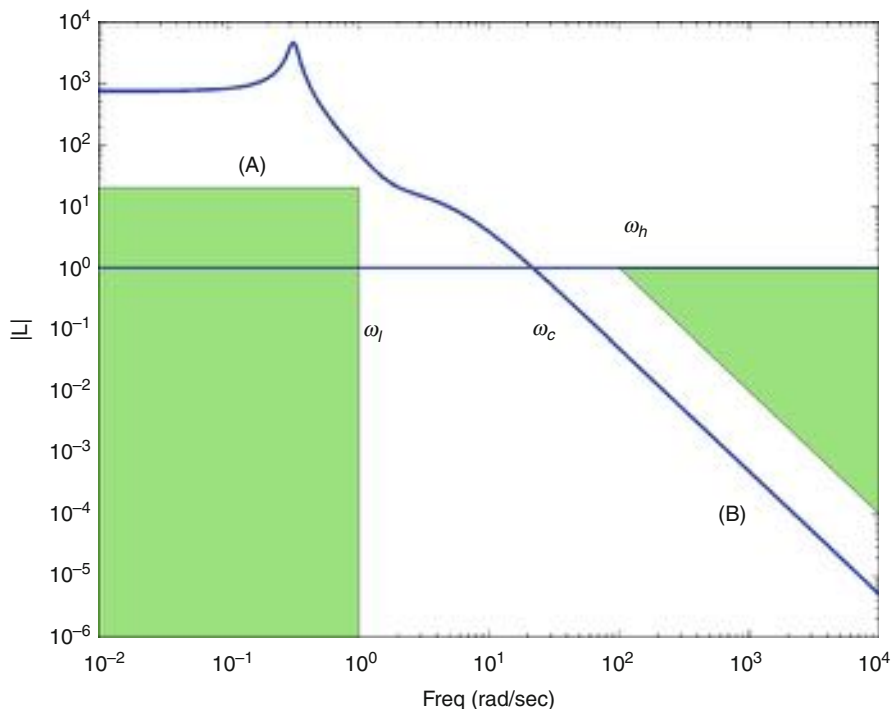


Fig. 27.8 Standard loop shaping goals for the indirect design approach

$\|S(j\omega)\|$ and $\|T(j\omega)\|$ small at the same frequencies, which represents a fundamental design constraint.

There are two basic approaches to design: the indirect approach that works on L and the *loop transfer function*, rather than S and T . Much of classical control design takes this approach. The direct approach, which works with S and T , is discussed in Sect. 27.3.4.4.

For the indirect approach, note that if $|L(j\omega)| \gg 1$, $S = (1 + L)^{-1} \approx L^{-1}$, so $|S| \ll 1$ and $|T| \approx 1$. Furthermore, if $|L(j\omega)| \ll 1$, $S = (1 + L)^{-1} \approx 1$, and $T \approx L$, so $|T| \ll 1$. So this converts the performance requirements on S , T into specifications on L , with the two regions in Fig. 27.8: (A) high loop gain which leads to good command following and disturbance rejection and (B) low loop gain leads to attenuation of any sensor noise. One must be careful at crossover, when $|L(j\omega_c)| \approx 1$, which requires that $\arg L(j\omega_c) \neq \pm 180^\circ$ to maintain stability (Levine 1996; Franklin et al. 1994; Kuo 1991; Dorf and Bishop 1995; Skogestad and Postlethwaite 2005). In summary, the typical control design challenges are to achieve high enough gain at low frequency to obtain good performance, low enough gain at high frequency to attenuate sensor noise effects, and sufficient stability margins in the transition range.

27.3.2 Classical Techniques

27.3.2.1 Classical and PID Controllers

There are three standard forms of the classical controller, which in the form of Fig. 27.7 can be written as $u = G_c(s)e = \frac{N_c(s)}{D_c(s)}e$, where e is the *error signal*, typically the difference between the actual variable, for example, θ and the commanded value θ_c ($e = \theta_c - \theta$).

The simplest is proportional feedback, which uses $G_c \equiv K_g$ a gain, so that $N_c = D_c = 1$. This is adequate for some systems but is typically constrained in what gains can be applied before at least some of the dynamics of the vehicle are destabilized. Integral feedback uses $u(t) = K_i \int_0^t e(\tau) d\tau$ which means that $G_c(s) = \frac{K_i}{s}$. This is typically used to reduce/eliminate steady-state error since, if $e(\tau)$ is approximately constant, then the magnitude of $u(t)$ will grow and thus hopefully correct the error.

Example 1. Consider the error response of $G_p(s) = \frac{1}{(s+a)(s+b)}$ ($a > 0, b > 0$) to a step $r(t) = \mathbf{1}(t) \rightarrow r(s) = 1/s$ where

$$\frac{e}{r} = \frac{1}{1 + G_c G_p} = S(s) \rightarrow e(s) = \frac{r(s)}{(1 + G_c G_p)}$$

and $S(s)$ is the *sensitivity transfer function* for the closed-loop system. The final value theorem ($\lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s e(s)$) can be used to analyze the error. So with proportional control,

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} \left(\frac{s}{s} \right) \frac{1}{1 + K_g G_p(s)} = \frac{1}{1 + \frac{K_g}{ab}}$$

indicating that e_{ss} can be made small but only with a very large K_g . Note, however, that with integral control, $\lim_{s \rightarrow 0} G_c(s) = \infty$, so $e_{ss} \rightarrow 0$. To summarize, integral control improves the steady state, but this is at the expense of the transient response, which typically gets worse because the system is less well damped.

Example 2. $G_p(s) = \frac{1}{(s+a)(s+b)}$, add integral feedback to improve the steady-state response. Increasing K_i to increase the speed of the response pushes the poles toward the imaginary axis leading to a more oscillatory response (compare the possible locus of roots of the closed-loop system shown in Figs. 27.9 and 27.10 using the two control approaches).

If the proportional and integral (PI) feedback are combined, then $G_c(s) = K_1 + \frac{K_2}{s} = \frac{K_1 s + K_2}{s}$ which introduces a pole at the origin and zero at $s = -K_2/K_1$. The zero has the effect of reducing the tendency of the closed-loop poles to move into the

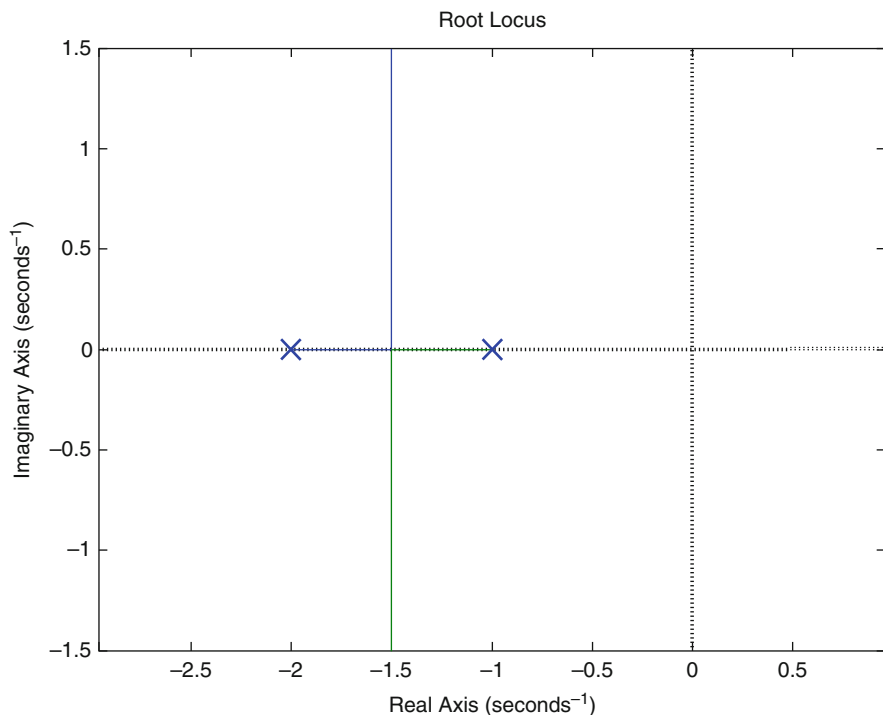


Fig. 27.9 Root locus for $G(s)$ in Example 2 using proportional feedback

right-half plane as the gain is increased, and thus, this PI combination solves some of the problems with using just integral control (compare Figs. 27.10 and 27.11).

The third approach is derivative feedback which uses $u = K_d \dot{e}$ so that $G_c(s) = K_d s$. Note that this will not help with the steady-state response since, if stable, at steady state one should have $\dot{e} \approx 0$. However, it does provide feedback on the rate of change of $e(t)$ so that the control can anticipate future errors.

Example 3. $G(s) = \frac{1}{(s-a)(s-b)}$, ($a > 0$, $b > 0$) with $G_c(s) = K_d s$. As shown in Fig. 27.12, derivative feedback is very useful for pulling the root locus into the left-hand plane or increasing the damping leading to a more stable response. It is typically used in combination with proportional feedback to form proportional-derivative feedback PD $G_c(s) = K_1 + K_2 s$ which moves the zero in Fig. 27.12 away from the origin.

The three approaches discussed above can be combined into the standard PID controller:

$$G_c(s) = \left(K_p + \frac{K_I}{s} + K_D s \right). \quad (27.60)$$

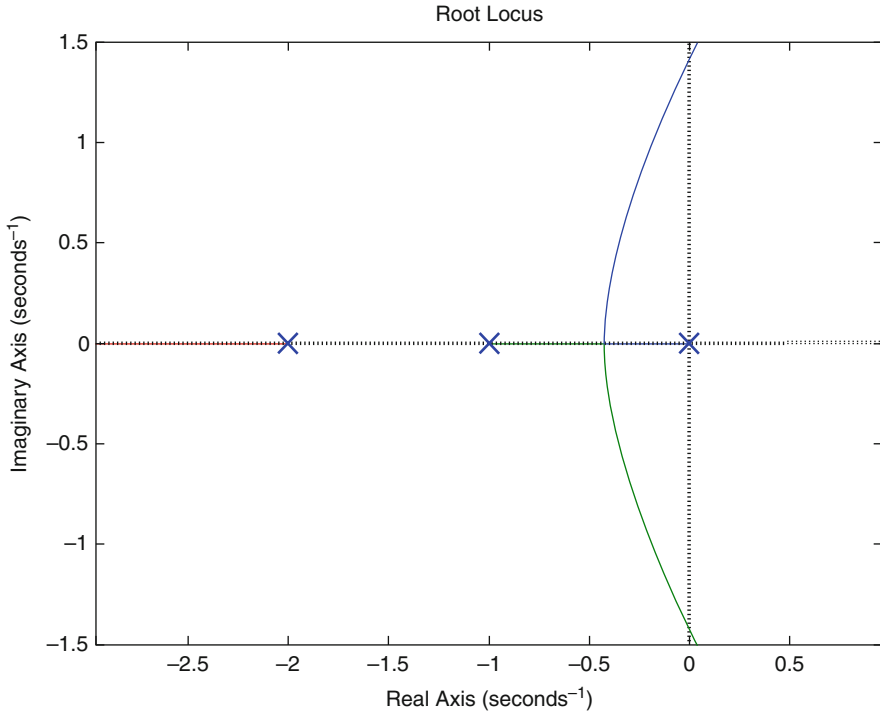


Fig. 27.10 Root locus for $G(s)$ in Example 2 with integral feedback

Typical difficulties in this case are that numerical differentiation of a noisy measured signal can result in noisy or lagged estimates in practice. Furthermore, long-term integration of a signal that has an unknown bias can lead to large control commands. Thus, one would typically use band-limited differentiation/integration instead, by rolling off the PD control with a high-frequency pole (or two) and only integrating signals above a certain frequency (high pass filter) in the PI controller. This leads to a slightly different primary building block for the controller components that is of the form

$$G_B(s) = K_c \frac{(s + z)}{(s + p)} \quad (27.61)$$

which, depending on how the gains are selected, can be morphed into various types of controllers. For example, if one picks $z > p$, with p small, then $G_B(s) \approx K_c \frac{(s+z)}{s}$ which is essentially a PI compensator. In this form, the compensator is known as a *lag*. If instead p is chosen such that $p \gg z$, then at low frequency, the impact of $\frac{p}{(s+p)}$ is small, so $G_B(s) = \frac{K_c}{p}(s + z) \frac{p}{(s+p)} \approx K_c'(s + z)$ which is essentially a PD compensator. In this form, the compensator is called a *lead*.

Various algorithms exist for selecting the components of $G_B(s)$. One basic procedure is to first use a lead controller to augment the phase margin of the loop transfer function $L = G_c(s)G(s)$ at the crossover frequency ω_c to meet

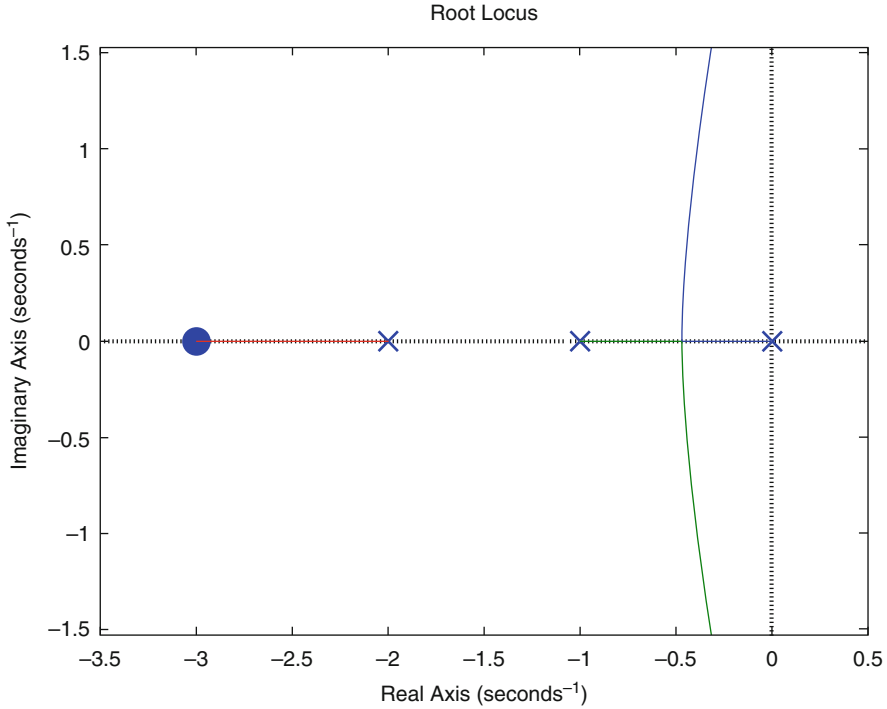


Fig. 27.11 Root locus for $G(s)$ using proportional and integral feedback

the requirements (e.g., desired phase margin). The phase margin is a standard stability margin for a system that measures the amount that the phase of the loop transfer function $L(s)$ differs from -180° when $|L(j\omega)| = 1$. Typically one wants a phase margin of greater than 30° , but other important performance metrics include the gain margin, the peak of the allowable transient, rise, and settling times, and time-delay margin (Dorf and Bishop 1995; Özbay 2000).

1. Find ϕ_{required} required from the performance specifications. Note that $\phi_{\text{required}} = PM - (180^\circ + \angle G(j\omega_c))$.
2. Arrange to put ϕ_{max} at the crossover frequency. Note that the maximum phase added by a lead is $\sin \phi_{\text{max}} = \frac{1-\alpha}{1+\alpha}$ where $\alpha = \frac{|z|}{|p|}$, which implies that $\alpha = \frac{1-\sin \phi_{\text{max}}}{1+\sin \phi_{\text{max}}}$. Further, since the frequency of the maximum phase addition is $\omega_{\text{max}} = \sqrt{|z| \cdot |p|}$, set $\omega_c^2 = |p| \cdot |z|$.
3. Select K_c so that the crossover frequency is at ω_c .

To increase the low-frequency gain to improve the steady-state response, one could use an integrator or use a lag to increase the loop gain. This is done by picking k_c to yield the desired low-frequency gain increase for the loop transfer function (which now includes the vehicle dynamics and the lead controller if added). It is also important to include the constraint that $\lim_{|s| \rightarrow \infty} |G_{\text{lag}}(s)| = 1$, so $k_c = |z|/|p|$

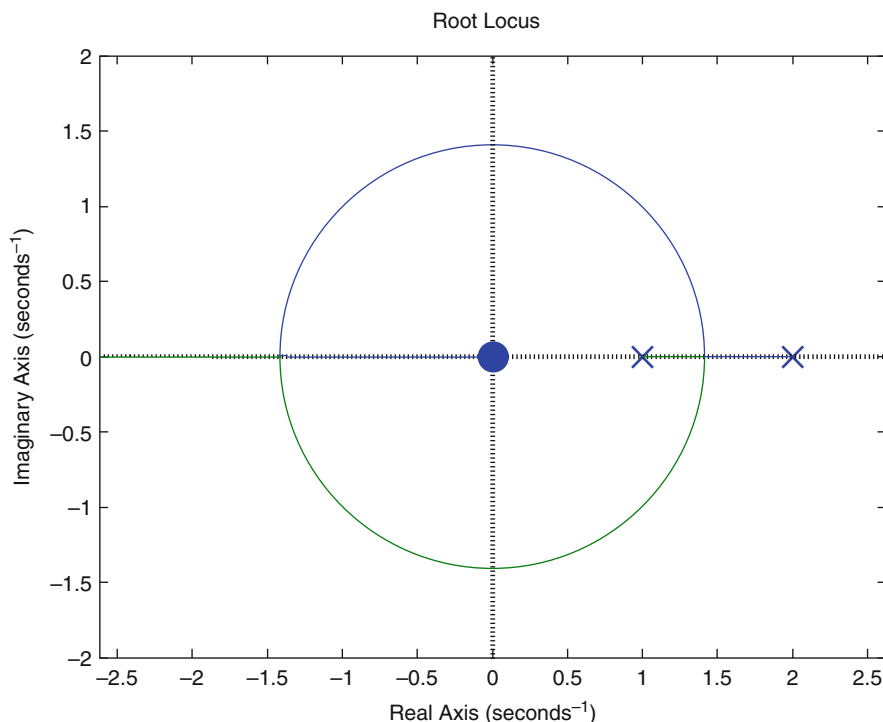


Fig. 27.12 Root locus for $G(s)$ using derivative feedback

to avoid impacting the higher-frequency dynamics. A rule of thumb is to limit the frequency of the zero of the lag compensator so that there is a minimal impact of the phase lag at ω_c . Thus, with $z \approx \omega_c/10$ and k_c both specified, one can solve for the pole location using $|p| = |z|/k_c$.

27.3.3 Successive Loop-Closure Examples

As an example of the successive loop-closure process in Fig. 27.13, the following considers the combination of the velocity and attitude control loops, wherein the attitude reference command is created by the velocity controller. Note that for small angles, the attitude controller could be based on independent PD control of each Euler angle, but the following presents a more general attitude control approach that is based on the quaternion representation of the vehicle attitude. A second example of the successive loop-closure approach is provided in Sect. 27.4.

27.3.3.1 Quaternion-Based Attitude Control

The sequential rotation properties of quaternions can be used to devise a generalized attitude controller for flight vehicles (Wie and Barba 1985; Knoebel 2007; Hall et al. 2008; Michini 2009). Consider some desired aircraft attitude that is

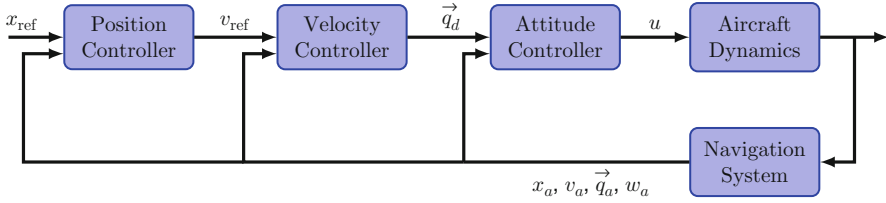


Fig. 27.13 Example of a possible successive loop-closure control architecture where the measured quantities $(x_a, v_a, \vec{q}_a, \omega_a)$ are fed back to specific control loops. Each outer loop generates a reference command that is used to create error signals for the inner loop. Saturation on these reference commands can easily be incorporated (not shown). As shown in the text, the particular architecture required depends on the vehicle and control objectives

specified as a quaternion rotation from the global frame \vec{q}_d . This desired rotation can be constructed intuitively with an axis and an angle as in Eq. (27.9). The actual *measured* attitude of the aircraft can also be represented as a quaternion rotation from the global frame \vec{q}_a . Now consider that the desired attitude \vec{q}_d can be constructed as the measured attitude \vec{q}_a sequentially rotated by some error quaternion \vec{q}_e . Since quaternion rotations are sequential, the error quaternion represents a rotation from the *body frame*:

$$\underbrace{q_d}_{\text{global frame}} = \underbrace{q_e}_{\text{body frame}} \circ \underbrace{q_a}_{\text{global frame}} \quad (27.62)$$

The error quaternion can then be solved for explicitly using the conjugate of the measured attitude quaternion:

$$\underbrace{q_e}_{\text{body frame}} = \underbrace{q_d}_{\text{global frame}} \circ \underbrace{q_a^*}_{\text{global frame}} \quad (27.63)$$

The error quaternion q_e represents the rotation required to get from the measured attitude to the desired attitude. Since the rotation is from the body frame, the x , y , and z components of q_e correspond to the rotations needed about the x , y , and z body axes of the aircraft. Thus, the three components correspond directly to the required aileron, elevator, and rudder commands (or roll cyclic, pitch cyclic, and pedal commands for a helicopter UAV) without any transformation. The scalar part of q_e , \hat{q}_e represents the angle through which the aircraft must be rotated and is thus proportional to the amount of control effort required. As in Sect. 27.2.2.1, let \vec{q}_e denote the vector part of q_e .

The quaternion representation suggests a simple control law of the form $u = -K_p \vec{q}_e - K_d \omega$, where $u = [\delta_a, \delta_e, \delta_r]^T$ is the vector containing the aileron, elevator, and rudder deflections, and K_p and K_d are positive definite proportional and derivative gain matrices. However, this control law suffers from the unwinding phenomena (see, e.g., Mayhew et al. 2011) characterized by the aircraft undergoing

a larger full rotation in order to achieve the desired attitude instead of a more efficient smaller rotation. One reason this happens is because the unit quaternions $[1, 0, 0, 0]$ and $[-1, 0, 0, 0]$ both represent the unique zero attitude and are the equilibria of the quaternion dynamics given in Eq. (27.10). Hence, if a naïve control law was used for regulation to zero attitude represented by $q_d = [-1, 0, 0, 0]$, an undesirable rotation, possibly leading to instability, could occur. On the other hand, the control law $u = K_p \vec{q}_e - K_d \omega$ would stabilize $q_d = [-1, 0, 0, 0]$ but result in an undesirable rotation if $q_d = [1, 0, 0, 0]$. This indicates that the sign of the error quaternion determines the sign of the gain K_p . Therefore, a *consistent* control law that satisfies $u(\overset{\circ}{q}_e, \vec{q}, \omega) = u(-\overset{\circ}{q}_e, \vec{q}, \omega)$ is desirable (Mayhew et al. 2011). Several such consistent asymptotically stabilizing quaternion control laws can be formulated (see, e.g., Mayhew et al. 2011; Wie and Barba 1985), with two examples being

$$u = -\text{sgn}(\overset{\circ}{q}_e) K_p \vec{q}_e - K_d \omega, \quad (27.64)$$

or

$$u = -\frac{\overset{\circ}{q}_e}{2} K_p \vec{q}_e - K_d \omega. \quad (27.65)$$

Another way to implement a consistent control law is to simply monitor the sign of $\overset{\circ}{q}_e$ online and enforce that the right error quaternion representation is chosen. With this constraint, the following control law with time-varying gains has yielded good attitude control performance (Sobolic 2009; Michini 2009; Cutler and How 2012)

$$u = -\frac{\bar{\theta}_e}{\sin(\bar{\theta}_e/2)} K_p \vec{q}_e - K_d \omega, \quad (27.66)$$

where $\bar{\theta}_e = 2 \arccos(\overset{\circ}{q}_e)$.

While denoted above as aileron, elevator, and rudder, the control commands are given in the body frame and thus easily generalizable to any actuator set (e.g., a three-wing tailsitter or quadrotor helicopter) as long as the commands can be mapped to the appropriate actuators. Another key advantage of this attitude controller over Euler angle methods is the lack of singularities, that is, the output of the controller is the direct rotation needed (from the body frame) to get from the measured attitude to the desired attitude (Hall et al. 2008; Knoebel 2007; Michini 2009; Cutler and How 2012). Computationally, no trigonometric functions or rotation matrices are needed, making the controller amenable to onboard or embedded implementation.

27.3.3.2 Outer-Loop Horizontal Velocity Controller

The outer-loop velocity controller can be designed to command a desired attitude based on the desired velocity. For airplanes, this means that in order to achieve a desired velocity in the body y direction, the aircraft must rotate (bank) about

its body x -axis and/or rotate (yaw) about its body z -axis. Furthermore, in order to achieve a desired flight path angle γ , it must rotate (pitch) about its y -axis.

Velocity error and desired attitudes are also tightly coupled for rotorcraft UAVs, where horizontal velocity is achieved by tilting the thrust vector. For a rotorcraft with non-hinged blades such as typical quadrotors, this is often achieved by tilting the body itself. Tilting the vehicle in any direction will generate a horizontal component of velocity.

With the attitude controller in place, a simple outer-loop horizontal velocity controller can be added to generate the desired attitude q_d . To calculate the desired quaternion given the desired x , y , and z velocities $[v_{d_x}, v_{d_y}, v_{d_z}]$ and the actual measured x , y , and z velocities $[v_{a_x}, v_{a_y}, v_{a_z}]$, the velocity errors are calculated:

$$\begin{aligned} v_{e_x} &= v_{d_x} - v_{a_x} \\ v_{e_y} &= v_{d_y} - v_{a_y} \\ v_{e_z} &= v_{d_z} - v_{a_z}. \end{aligned} \quad (27.67)$$

The errors are then used to determine the desired quaternion rotation q_d in the form

$$q_d = [1.0, q_{d_x}(\vec{v}_e), q_{d_y}(\vec{v}_e), q_{d_z}(\vec{v}_e)]. \quad (27.68)$$

A simple control law that works particularly well for rotorcraft by generating the elements of the desired rotation in Eq. (27.68) based on the velocity error is

$$\begin{aligned} q_{d_x} &= K_{p_{v_y}} v_{e_y} + K_{i_{v_y}} \int v_{e_y} dt, \\ q_{d_y} &= -K_{p_{v_x}} v_{e_x} - K_{i_{v_x}} \int v_{e_x} dt, \\ q_{d_z} &= 0, \end{aligned} \quad (27.69)$$

where K_p are the proportional gains and K_i are the integral gains. In this control law, lateral motion is achieved by a rotation about the x -axis, and longitudinal motion is achieved by a rotation about the y -axis. For a rotorcraft, such a controller is based around the hover configuration, where thrust is being generated vertically to counteract the acceleration of gravity.

For airplanes, an outer-loop controller that commands the desired attitude is based around the level forward flight configuration. For airplanes, it is also desirable to command a yaw rotation in the direction that reduces lateral velocity error and a pitch rotation to help achieve the desired flight path angle. This can be achieved by the following control law:

$$q_{d_x} = K_{p_{v_y}} v_{e_y} + K_{i_{v_y}} \int v_{e_y} dt,$$

$$q_{d_y} = -K_{p_{v_z}} v_{e_z} - K_{i_{v_z}} \int v_{e_z} dt, \quad (27.70)$$

$$q_{d_z} = K_{p_{v_r}} v_{e_y} + K_{i_{v_r}} \int v_{e_y} dt.$$

Finally, the desired quaternion is normalized to ensure that $\|q_d\|_2 = 1$. The result is a combined proportional-integral (PI) horizontal velocity controller and quaternion attitude controller.

Care must be taken in commanding q_{d_y} for airplanes in order to avoid stall due to high angle of attacks. If airspeed measurements are available, this can be achieved by limiting commanded pitch attitude to keep the estimated angle of attack within limits (Chowdhary et al. 2011). Care must also be taken in order to ensure that the commanded attitude does not result in an undesired loss of altitude. For airplanes, this can be partially achieved by ensuring that the airplane pitches to maintain vertical speed as in Eq. (27.70). More generally, this can be achieved by limiting the magnitude of the desired attitudes in Eq. (27.68). In order to achieve this, define the inertial altitude error as $z_e = z_d - z_a$, where z_d is the desired altitude and z_a is the actual altitude, and redefine the authority limited quaternion attitude controller as

$$q_d = [1.0, q_{d_x}(\vec{v}_e) - \kappa_x, q_{d_y}(\vec{v}_e) - \kappa_y, q_{d_z}(\vec{v}_e)]. \quad (27.71)$$

An effective control authority limiting term is Sobolic (2009)

$$\begin{aligned} \kappa_x &= K_{p_z} \text{sgn}(v_{a_y}) z_e + K_{v_z} \text{sgn}(v_{a_y}) v_{d_z} \\ \kappa_y &= K_{p_z} \text{sgn}(v_{a_x}) z_e + K_{v_z} \text{sgn}(v_{a_x}) v_{d_z}. \end{aligned} \quad (27.72)$$

This controller is typically implemented in combination with a forward speed (or airspeed) controller and an altitude controller.

27.3.4 Multi-input–Multi-output Control

While PID and classical control are very effective ways of closing single-input–single-output (SISO) loops, it is often challenging to use those techniques to design controllers for systems with multiple sensors and actuators (MIMO). State space techniques can handle that additional complexity very easily.

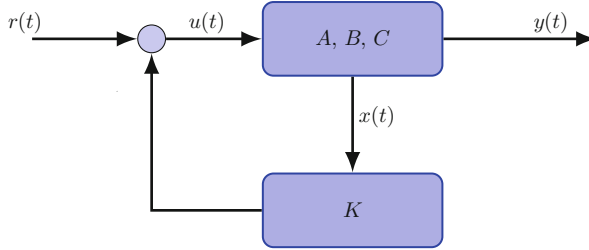
27.3.4.1 Full-State Feedback Controller

The decoupled system dynamics were given in state space form in Eqs. (27.47) and (27.48), where it is assumed that $D = 0$. The time response of this linear state space system is determined by the eigenvalues of the matrix A , which are the same as the poles of a transfer function representation (Chen 1984; Franklin et al. 1994; Stengel 1994). As before, if the eigenvalues (poles) are unstable or in the wrong location for the desired response, the goal is to use the inputs $u(t)$ to modify

the eigenvalues of A to change the system dynamics. The first approach is to assume a full-state feedback of the form

$$u(t) = r(t) - Kx(t), \quad (27.73)$$

where $r(t) \in \mathcal{R}^m$ is a reference input and $K \in \mathcal{R}^{m \times n}$ is the controller gain.



This gives the closed-loop dynamics:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B(r(t) - Kx(t)) = (A - BK)x(t) + Br(t) \\ &= A_{cl}x(t) + Br(t) \end{aligned} \quad (27.74)$$

$$y(t) = Cx(t). \quad (27.75)$$

Note that for a system with one input, there are n parameters in K that have to be selected to attempt to place the n eigenvalues in A . For a system with n states, a necessary and sufficient condition for controllability (see, e.g., Dorf and Bishop 1995) is that

$$\text{rank } \mathcal{M}_c \triangleq \text{rank } [B \ AB \ A^2B \ \dots \ A^{n-1}B] = n.$$

Ackermann's Formula

This gives us a method of doing this entire design process in one step. For single-input system, the gains are

$$K = [0 \ \dots \ 0 \ 1] \mathcal{M}_c^{-1} \Phi_c(A),$$

where the roots of the polynomial $\Phi_c(s)$ give the desired pole locations.

Linear Quadratic Regulator (LQR)

It may be difficult to relate the desired performance criteria to desired pole locations. In such cases, an alternative approach is to automatically select pole locations so that the closed-loop system optimizes the cost function

$$J_{\text{LQR}} = \int_0^\infty [x(t)^T R_{xx} x(t) + u(t)^T R_{uu} u(t)] dt, \quad (27.76)$$

where $x^T R_{xx} x$ is the state cost with weight $R_{xx} = R_{xx}^T \geq 0$ and $u^T R_{uu} u$ is called the control cost with weight $R_{uu} = R_{uu}^T > 0$. It is well known that the solution to this infinite horizon optimal control problem is a linear time-invariant state feedback of the form (see, e.g., Bryson and Ho 1969)

$$u(t) = -K_{\text{lqr}} x(t), \quad (27.77)$$

where $K_{\text{lqr}} = R^{-1} B^T P_r$ and P_r is found by solving the algebraic Riccati equation (ARE) for the positive semidefinite symmetric solution P_r

$$0 = A^T P_r + P_r A + R_{xx} - P_r B R_{uu}^{-1} B^T P_r. \quad (27.78)$$

Equation (27.78) can be easily solved numerically, for example, by using the commands `are` or `lqr` in MATLAB.

The design freedom in this case is to select the R_{xx} and R_{uu} matrices. A good rule of thumb, typically called the Bryson's rule (Bryson and Ho 1969), when selecting the weighting matrices R_{xx} and R_{uu} is to normalize the signals: $R_{xx} = \text{diag} \left(\frac{\alpha_1^2}{(x_1)_{\max}^2}, \dots, \frac{\alpha_n^2}{(x_n)_{\max}^2} \right)$ and $R_{uu} = \rho \text{diag} \left(\frac{\beta_1^2}{(u_1)_{\max}^2}, \dots, \frac{\beta_m^2}{(u_m)_{\max}^2} \right)$. The $(x_i)_{\max}$ and $(u_i)_{\max}$ represent the largest desired response or control input for that component of the state/actuator signal. $\sum_i \alpha_i^2 = 1$ and $\sum_i \beta_i^2 = 1$ are used to add an additional relative weighting on the various components of the state/control. The ρ is used as the last relative weighting between the control and state penalties – it effectively sets the controller bandwidth. This gives a relatively concrete way to discuss the relative size of R_{xx} and R_{uu} and their ratio R_{xx}/R_{uu} .

LQR is a very attractive control approach because it easily handles multiple actuators and complex system dynamics. Furthermore, it offers very large stability margins to errors in the loop gain (gain margin of infinity, gain reduction margin of 1/2, and a minimum phase margin of 60° in each control input channel) (Stengel 1994). However, LQR assumes access to the full state, which is rarely available. This assumption is removed in Sect. 27.3.4.2.

Reference Inputs

The proceeding showed how to pick K to change the dynamics to achieve some desired properties (i.e., stabilize A). The question remains as to how well this controller allows the system to track a reference command, which is a performance issue rather than just stability. For good tracking performance the goal is to have $y(t) \approx r(t)$ as $t \rightarrow \infty$. To consider this performance issue in the frequency domain, the final value theorem can be used. This indicates that for good performance, $sY(s) \approx sR(s)$ as $s \rightarrow 0$, which gives $\left. \frac{Y(s)}{R(s)} \right|_{s=0} = 1$. So, for good performance, the transfer function from $R(s)$ to $Y(s)$ should be approximately 1 at low frequency (DC).

One approach is to scale a constant reference input r so that $u = \bar{N}r - Kx(t)$ where \bar{N} is an extra gain used to scale the closed-loop transfer function.

The closed-loop equations are now $y = Cx(t)$ and

$$\dot{x}(t) = (A - BK)x(t) + B\bar{N}r \quad (27.79)$$

so that

$$\frac{Y(s)}{R(s)} = C(sI - (A - BK))^{-1} B\bar{N} = G_{cl}(s)\bar{N}.$$

Given the goal at DC, one can compute

$$\bar{N} = G_{cl}(0)^{-1} = -(C(A - BK)^{-1}B)^{-1}. \quad (27.80)$$

Note that this development assumed that r was constant, but it could also be used if r is a slowly time-varying command. This approach is simple to implement but is typically not robust to errors in the choice of \bar{N} .

An alternative strategy within the LQR framework is to embed an integrator into the feedback control to obtain an *LQR servo*. If the relevant system output is $y(t) = C_y x(t)$ and reference $r(t)$, add extra states $x_I(t)$, where $\dot{x}_I(t) = e(t)$. In this case, if the state of the original system is $x(t)$, then the dynamics are modified to be

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_I(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C_y & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_I(t) \end{bmatrix} + \begin{bmatrix} B_u \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} r(t) \quad (27.81)$$

or

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}u(t) + \bar{B}_r r(t), \quad (27.82)$$

where $\bar{x}(t) = [x^T(t) \ x_I^T(t)]^T$. The modified optimal control problem is to obtain the feedback controller that minimizes the cost:

$$J_{\text{LQR servo}} = \int_0^\infty [\bar{x}^T \bar{R}_{xx} \bar{x} + u^T R_{uu} u] dt$$

which is found by solving the Riccati equation (27.78) using the augmented dynamics \bar{A} and \bar{B} to obtain

$$u(t) = -\bar{K}\bar{x}(t) = -[K \ K_I] \begin{bmatrix} x \\ x_I \end{bmatrix}.$$

Note that the implementation of this LQR servo requires that the integrator states x_I be implemented in software. While slightly more complicated to implement, the advantage of this approach is the additional robustness of ensuring a sufficiently high low-frequency gain in the loop transfer function that the steady-state tracking error to a step reference input will be zero. As such, the LQR-servo form is highly recommended.

27.3.4.2 Estimation

The preceding assumed access to the full state for the feedback control. If that information is available, the control given is very effective, but typically only the output of a few sensors are available. In this case, consider the n -th order system model of the form in Eqs. (27.47) and (27.48) (again with $D = 0$), $x(0)$ unknown, and C such that $x(t)$ cannot be directly determined. One could try direct (static) output feedback of the form $u(t) = K_y y(t)$, but finding K_y is very difficult, and this approach will be fundamentally limited in what it can achieve.

The alternative typically used is to create a dynamic output feedback (DOFB) controller, which uses an *observer* or *estimator* to estimate the system state as $\hat{x}(t)$, and then employs full-state feedback on the estimated state (so $u(t) = -K\hat{x}(t)$) to control the system. This approach, called the *separation principle*, separates the design of the DOFB compensator into the regulator problem (picking K) and the design of the estimator (Stengel 1994; Franklin et al. 1994; Bryson and Ho 1969).

The estimator design starts with a simulation of the system dynamics, which are assumed known, but then adds a feedback term on the measurement error – the difference between what was measured and what was expected based on the estimator, $\tilde{y}(t) = y(t) - C\hat{x}(t)$. The feedback gain is called L_e , and it yields the closed-loop estimator:

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L_e\tilde{y}(t) \\ &= (A - L_eC)\hat{x}(t) + Bu(t) + L_e y(t)\end{aligned}\quad (27.83)$$

which is a dynamic system with poles given by $\lambda_i(A - L_eC)$ and which takes the measured plant outputs as an input and generates an estimate of $x(t)$. Note that if the pair A, C is observable, then the eigenvalues of $A - L_eC$ can be placed arbitrarily, leading to a similar (actually the *dual*) design problem as the regulator. A necessary and sufficient condition for observability (see, e.g., Dorf and Bishop 1995; Özbay 2000) is that

$$\text{rank } \mathcal{M}_o \triangleq \text{rank} \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = n. \quad (27.84)$$

L_e can be determined using pole placement, with the typical strategy of choosing the real part of the poles to be approximately a factor of 2 larger in magnitude than the associated regulator poles. Given the desired estimator pole locations specified as the roots of the polynomial $\Phi_e(s)$, the estimator equivalent of Ackermann's formula is

$$L_e = \Phi_e(A)\mathcal{M}_o^{-1} [0 \dots 0 \ 1]^T. \quad (27.85)$$

Optimal Estimation

Similar to Eq. (27.52), noise in the system is typically modeled as

$$\dot{x}(t) = Ax(t) + Bu(t) + B_w w(t) \quad (27.86)$$

$$y(t) = C_y x(t) + v(t) \quad (27.87)$$

where w is the *process noise* that models the uncertainty in the system model and v is the *sensor noise* that models uncertainty in the measurements. It is typically assumed that $w(t)$ and $v(t)$ are zero mean ($E[w(t)] = 0$, $E[v(t)] = 0$) and uncorrelated Gaussian white random noises so that

$$E[w(t_1)w(t_2)^T] = R_{ww}(t_1)\delta(t_1 - t_2) \Rightarrow w(t) \sim \mathcal{N}(0, R_{ww}),$$

$$E[v(t_1)v(t_2)^T] = R_{vv}(t_1)\delta(t_1 - t_2) \Rightarrow v(t) \sim \mathcal{N}(0, R_{vv}),$$

$$E[w(t_1)v(t_2)^T] = 0.$$

Given this noise, one can develop an optimal estimator, where the key step in the design is to balance the effect of the various types of random noise in the system on the estimation error. With noise in the system, the dynamics of the estimation error $\tilde{x}(t)$ for the closed-loop estimator are

$$\begin{aligned} \dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}} = [Ax + Bu + B_w w] - [A\hat{x} + Bu + L_e(y - \hat{y})] \\ &= (A - L_e C_y)\tilde{x} + B_w w - L_e v. \end{aligned} \quad (27.88)$$

Equation (27.88) explicitly shows the conflict in the estimator design process because one must achieve a balance between the speed of the estimator decay rate, which is governed by $\mathcal{R}[\lambda_i(A - L_e C_y)]$, and the impact of the sensing noise v through the gain L_e . Fast state reconstruction requires rapid decay rate, which typically requires a large L_e , but that tends to magnify the effect of v on the estimation process. The effect of the process noise cannot be modified, but the choice of L_e will tend to mitigate/accentuate the effect of v on $\tilde{x}(t)$.

An optimal filter can be designed that minimizes $J = \text{trace}(Q_e(t))$ with (see Bryson and Ho 1969; Kwakernaak and Sivan 1972; Lewis 1986; Gelb 1974; Crassidis and Junkins 2004; Stengel 1994; Simon 2006 for details)

$$Q_e(t) = E[\{x(t) - \hat{x}(t)\}\{x(t) - \hat{x}(t)\}^T]$$

which is the covariance matrix for the estimation error. The result is a closed-loop estimator with $L_e(t) = Q_e(t)C_y^T R_{vv}^{-1}$ and $Q_e(t) \geq 0$ solves

$$\dot{Q}_e(t) = A Q_e(t) + Q_e(t) A^T + B_w R_{ww} B_w^T - Q_e(t) C_y^T R_{vv}^{-1} C_y Q_e(t)$$

where $\hat{x}(0)$ and $Q_e(0)$ are known. It is standard to assume that $R_{vv} > 0$, $R_{ww} > 0$, all plant dynamics are constant in time, $[A, C_y]$ is detectable, and $[A, B_w]$ is stabilizable. A system is said to be stabilizable if all unstable modes are controllable and is detectable if all unstable modes are observable. In this case, the forward propagation of the covariance $Q_e(t)$ settles down to a constant Q_{ss} independent of $Q_e(0)$, as $t \rightarrow \infty$ where

$$AQ_{ss} + Q_{ss}A^T + B_w R_{ww} B_w^T - Q_{ss} C_y^T R_{vv}^{-1} C_y Q_{ss} = 0$$

and $L_{ss} = Q_{ss} C_y^T R_{vv}^{-1}$. Note that the stabilizable/detectable assumptions give a unique $Q_{ss} \geq 0$. Also, if Q_{ss} exists, the steady-state filter

$$\dot{\hat{x}}(t) = (A - L_{ss} C_y) \hat{x}(t) + L_{ss} y(t) \quad (27.89)$$

is asymptotically stable given these assumption, and thus, L_{ss} can be used as the estimator gains L_e for the system.

27.3.4.3 Dynamic Output Feedback Control System

Given a closed-loop estimator with gain L_e ,

$$\dot{\hat{x}}(t) = (A - L_e C) \hat{x}(t) + B u(t) + L_e y(t). \quad (27.90)$$

Then if L_e is chosen as outlined in the previous sections, it is known that $\hat{x}(t) \rightarrow x(t)$ as $t \rightarrow \infty$, but in general, $\hat{x}(t) \neq x(t) \forall t$. Even so, the approach taken in the output feedback control case is to implement $u(t) = -K \hat{x}(t)$, leading to the closed-loop system dynamics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\hat{x}}(t) \end{bmatrix} = \begin{bmatrix} A & -BK \\ L_e C & A - BK - L_e C \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} \quad (27.91)$$

or

$$\dot{x}_{cl}(t) = A_{cl} x_{cl}(t). \quad (27.92)$$

The stability of this closed-loop system can be analyzed using the *similarity transformation* matrix $T = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$ that preserves the location of the eigenvalues.

In fact it is easy to show that the dynamics matrix of the transformed system is given by

$$\bar{A}_{cl} \triangleq T A_{cl} T^{-1} = \begin{bmatrix} A - BK & BK \\ 0 & A - L_e C \end{bmatrix}.$$

Thus, the closed-loop pole locations are given by the roots of the polynomial

$$\det(sI - \bar{A}_{cl}) \triangleq \det(sI - (A - BK)) \cdot \det(sI - (A - L_e C)) = 0$$

which consist of the union of the regulator poles and estimator poles. The significance of this result is that it implies that one can design the estimator and regulator separately and combine them at the end, and the closed-loop system will be stable. This is called the *separation principle*, and when it holds, it greatly simplifies the control design process.

The resulting *dynamic output feedback compensator* is the combination of the regulator and closed-loop estimator written using the new state $x_c(t) \equiv \hat{x}(t)$ as

$$\dot{x}_c(t) = A_c x_c(t) + B_c y(t) \quad (27.93)$$

$$u(t) = -C_c x_c(t), \quad (27.94)$$

where the compensator dynamics are $A_c \triangleq A - BK - L_e C$, $B_c \triangleq L_e$, and $C_c \triangleq K$. Note that, as expected, the compensator maps *sensor measurements* to *actuator commands*.

The DOFB solution provided in this section is called the (infinite horizon) linear quadratic Gaussian (LQG), and it is the controller that optimizes the expected time-averaged value of the cost in Eq. (27.76) for the stochastic system in Eqs. (27.86) and (27.87) (Bryson and Ho 1969; Stengel 1994; Franklin et al. 1994; Skogestad and Postlethwaite 2005). While LQG has proven to be very effective design algorithm for many MIMO systems, it does not have the same robustness properties guaranteed for the LQR design (Doyle 1978). Furthermore, LQG designs can exhibit high sensitivity to modeling errors. Thus, it is prudent to be cautious and check the peak magnitude of the sensitivity $S(s)$, which provides a measure of the closeness of the loop transfer function to the critical point at $s = -1$. High sensitivities (above 20–50) are indicative of designs that could be susceptible to modeling errors, so further testing and simulation using perturbed dynamics should be considered.

27.3.4.4 Robust Control Techniques

The alternative approach to control design mentioned in Sect. 27.3.1 works with the $S(s)$ and $T(s)$ functions to shape them directly (McFarlane and Glover 1992). For example, consider the following design problem in Fig. 27.14 where

$$G(s) = \frac{200}{(0.05s + 1)^2(10s + 1)}.$$

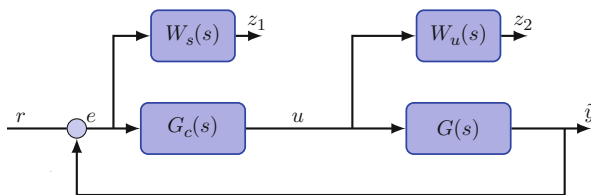


Fig. 27.14 Direct design problem

Note that there is one input (r) and two performance outputs – one that penalizes the sensitivity $S(s)$ of the system and the other that penalizes the control effort used with the weight functions $W_s(s)$ and $W_u(s)$ to be defined later. With

$$z_1 = W_s(s)(r - Gu) \quad (27.95)$$

$$z_2 = W_u(s)u \quad (27.96)$$

$$e = r - Gu, \quad (27.97)$$

then closing the loop with $u = G_c(s)e$ yields

$$P_{CL}(s) = \begin{bmatrix} W_s \\ 0 \end{bmatrix} + \begin{bmatrix} -W_s G \\ W_u \end{bmatrix} G_c(I + G G_c)^{-1} \quad (27.98)$$

$$= \begin{bmatrix} W_s - W_s G G_c S \\ W_u G_c S \end{bmatrix} = \begin{bmatrix} W_s S \\ W_u G_c S \end{bmatrix} \quad (27.99)$$

so that

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} W_s S \\ W_u G_c S \end{bmatrix} r, \quad (27.100)$$

and thus, the closed-loop transfer function from the reference input to the performance variable z_1 is the weighted sensitivity function.

This setup gives us direct access to the sensitivity function in the analysis. It is now possible to embed tests such as whether $|S(s)|$ satisfies a desired bound such as

$$|S(j\omega)| < \frac{1}{|W_s(j\omega)|} \quad \forall \omega$$

or equivalently, whether $|W_s(j\omega)S(j\omega)| < 1$, $\forall \omega$. To make this bound useful, one can select the weight $W_s(s)$ to emphasize different frequency regions. For example, to achieve good low-frequency tracking and a crossover frequency of about 10 rad/s, pick

$$W_s = \frac{s/1.5 + 10}{s + (10) \cdot (0.0001)} \quad W_u = 1.$$

Once the architecture of the problem is determined, the structure of $P_{CL}(s)$ is set. The design choice is in the selection of the weights $W_s(s)$ and $W_u(s)$ which then “shape” the closed-loop $S(s)$. The output z_2 participates by providing access to a measure of the amount of control authority being used.

The overall objective then is to bound the size of the closed-loop transfer function $P_{CL}(s)$, which in most cases will be a matrix. The bound used is based on the singular values of the transfer function matrix evaluated over all frequencies. For a complex transfer function matrix $G(j\omega)$, the singular values are the positive square roots of the eigenvalues of $G^H G$. The largest of these is called the maximum singular value, so that $\bar{\sigma}[G(j\omega)] = \max_i \sqrt{\lambda_i(G^H G)}$. Thus, for a stable system,

define

$$\|P_{CL}(s)\|_{\infty} = \sup_{\omega} \bar{\sigma}[P_{CL}(\mathbf{j}\omega)]. \quad (27.101)$$

One interpretation of this \mathcal{H}_{∞} norm, $\|P_{CL}(s)\|_{\infty}$, of the system is that it is the energy gain from the input to output. This maximum gain is achieved using a “worst case” input signal that is essentially a sinusoid at frequency ω^* with (appropriate input direction) that yields $\bar{\sigma}[P_{CL}(\mathbf{j}\omega^*)]$ as the amplification. This interpretation is consistent with the graphical test which finds the peaks of $\bar{\sigma}$ plotted versus frequency. Note that the difference here is that the performance test in the \mathcal{H}_{∞} case is concerned with the maximum amplification of the input signals at any one frequency (worst case). This is contrast to the LQG design, which is concerned with minimizing the sum of the square of all singular values over all frequencies (average case).

The \mathcal{H}_{∞} norm can also be evaluated using a state space test based on the Hamiltonian matrix \mathcal{H} (Zhou et al. 1996), and the following result, which states that the gain of a generic stable system $G(s)$ with state space representation

$$\dot{x}_n(t) = A_n x_n(t) + B_n u(t) \quad (27.102)$$

$$y(t) = C_n x_n(t) + D_n u(t), \quad (27.103)$$

is bounded in the sense that $G_{\max} = \|G(s)\|_{\infty} < \gamma$ if and only if

1. $|D| < \gamma$.
2. The Hamiltonian matrix

$$\mathcal{H} = \left[\begin{array}{c|c} A_n + B_n(\gamma^2 I - D_n^T D_n)^{-1} D_n^T C_n & B_n(\gamma^2 I - D_n^T D_n)^{-1} B_n^T \\ \hline -C_n^T (I + D_n(\gamma^2 I - D_n^T D_n)^{-1} D_n^T) C_n & -A_n^T - C_n^T D_n(\gamma^2 I - D_n^T D_n)^{-1} B_n^T \end{array} \right]$$

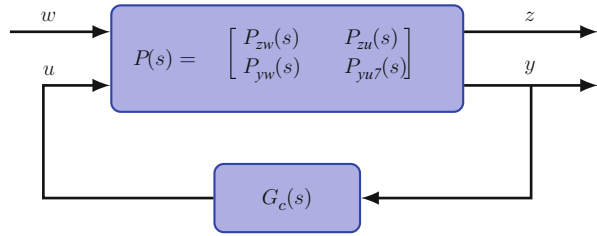
has no eigenvalues on the imaginary axis.

Thus, finding the \mathcal{H}_{∞} norm of a system requires to find the smallest γ for which no imaginary eigenvalues of \mathcal{H} exist. This is an iterative process, typically done using a bisection search called the γ -iteration (Skogestad and Postlethwaite 2005; Zhou et al. 1996). Given these analysis tools, the synthesis problem can now be defined.

Robust Control Synthesis

The synthesis problem using the analysis techniques in the previous section is to design a controller that, to the extent possible, ensures that the loop-shaping objectives are satisfied. The approach works with the generalized plant that has all weighting functions embedded. As depicted in Fig. 27.15, the input/output signals of the generalized plant $P(s)$ are defined as z performance output, w disturbance/reference inputs, y sensor outputs, and u actuator inputs. With the loop closed ($u = G_c(s)y$), one can show that

$$P_{CL}(s) \equiv F_l(P, G_c) = P_{zw} + P_{zu} G_c (I - P_{yu} G_c)^{-1} P_{yw} \quad (27.104)$$

Fig. 27.15 Generalized plant for \mathcal{H}_∞ synthesis

which is called a (lower) linear fractional transformation (LFT). The design objective is to find $G_c(s)$ to stabilize the closed-loop system and minimize $\|F_l(P, G_c)\|_\infty$. As noted previously, even finding the \mathcal{H}_∞ norm is challenging, so the approach taken to this synthesis problem is to consider a suboptimal problem:

1. Find $G_c(s)$ to satisfy $\|F_l(P, G_c)\|_\infty < \gamma$.
2. Then use the γ -iteration to find the smallest value (γ_{opt}) for which $\|F_l(P, G_c)\|_\infty < \gamma_{\text{opt}}$.

Thus, the suboptimal \mathcal{H}_∞ synthesis problem (Maciejowski 1989; Skogestad and Postlethwaite 2005; Zhou et al. 1996) is to find $G_c(s)$ to satisfy $\|F_l(P, G_c)\|_\infty < \gamma$:

$$P(s) = \begin{bmatrix} P_{zw}(s) & P_{zu}(s) \\ P_{yw}(s) & P_{yu}(s) \end{bmatrix} := \left[\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & 0 & D_{zu} \\ \hline C_y & D_{yw} & 0 \end{array} \right]$$

where it is assumed that

1. (A, B_u, C_y) are stabilizable/detectable (essential).
2. (A, B_w, C_z) are stabilizable/detectable (essential).
3. $D_{zu}^T [C_z \ D_{zu}] = [0 \ I]$ (simplifying, essential).
4. $\begin{bmatrix} B_w \\ D_{yw} \end{bmatrix} D_{yw}^T = \begin{bmatrix} 0 \\ I \end{bmatrix}$ (simplifying, essential).

Skogestad and Postlethwaite (2005) and Zhou et al. (1996) derive the controller for this case, and to summarize, there exists a stabilizing $G_c(s)$ such that $\|F_l(P, G_c)\|_\infty < \gamma$ if and only if

- (1) $\exists X \geq 0$ that solves

$$A^T X + XA + C_z^T C_z + X(\gamma^{-2} B_w B_w^T - B_u B_u^T) X = 0 \quad (27.105)$$

$$\text{and } \mathcal{R}\lambda_i [A + (\gamma^{-2} B_w B_w^T - B_u B_u^T) X] < 0 \quad \forall i$$

- (2) $\exists Y \geq 0$ that solves

$$AY + YA^T + B_w^T B_w + Y(\gamma^{-2} C_z^T C_z - C_y^T C_y) Y = 0 \quad (27.106)$$

$$\text{and } \mathcal{R}\lambda_i [A + Y(\gamma^{-2} C_z^T C_z - C_y^T C_y)] < 0 \quad \forall i$$

- (3) $\rho(XY) < \gamma^2$ (27.107)

where ρ is the spectral radius ($\rho(A) = \max_i |\lambda_i(A)|$). Given these solutions, the central \mathcal{H}_∞ controller is written as

$$G_c(s) := \left[\begin{array}{c|c} A + (\gamma^{-2} B_w B_w^T - B_u B_u^T) X - Z Y C_y^T C_y & Z Y C_y^T \\ \hline -B_u^T X & 0 \end{array} \right],$$

where $Z = (I - \gamma^{-2} Y X)^{-1}$. Note that the central controller has as many states as the generalized plant. Also, this design does not decouple as well as the regulator/estimator for LQG, though it can be shown that a separation principle does exist (Green and Limbeer 1995; Zhou et al. 1996). More general forms of the \mathcal{H}_∞ synthesis problem are available that relax some of the assumptions given above (Zhou et al. 1996; Skogestad and Postlethwaite 2005).

The standard design approach is to define an architecture such as Fig. 27.14 that captures the key transfer functions that must be shaped in $P_{CL}(s)$. The functional form of the weighting functions $W_s(s)$ and $W_u(s)$ must then be selected, written in state space form and then embedded into the generalized plant leading to a state space representation of $P(s)$. One can then use the various MATLAB toolboxes (e.g., μ -tools or the robust control toolbox (Balas et al. 1995; Chiang and Safonov 1997; Gahinet et al. 1994)) to solve for the controller $G_c(s)$. The parameters of the weighting functions can then be adjusted to tune the resulting closed-loop transfer functions. Numerous examples of using these approaches for aircraft flight control exist in the literature, see, for example, Hyde (1995), Hu et al. (2000), Sadraey and Colgren (2005), Balas et al. (1998), Reiner et al. (1993), Gadewadikar et al. (2009), and Chao et al. (2007).

One appealing aspect of this design process is that, in contrast to the LQG controller framework, there are direct extensions of the \mathcal{H}_∞ control design process to account for model uncertainty (Balas et al. 1995; Chiang and Safonov 1997; Gahinet et al. 1994). However, a challenge in this case is finding an appropriate *error model* for the model uncertainty that fit into the required assumptions. Also note that the dimension of the resulting controller can increase dramatically as the various weighting functions required to capture this model uncertainty are included. Controller order reduction is possible, but it is often unclear what impact this has on the performance and robustness of the closed-loop system.

27.4 Path Following

Once the controllers for the inner-loop dynamics (e.g., attitude and velocity) have been designed, it is then possible to consider tracking a desired flight path. This flight path might be in the form of interpolated waypoints or a continuous smooth curve. In either case, this path represents the desired location of the vehicle in 3D (possibly 4D if it is time-parametrized), and the goal of the outer-loop controller is to generate reference commands for the inner loops which ensure that the path is tracked.

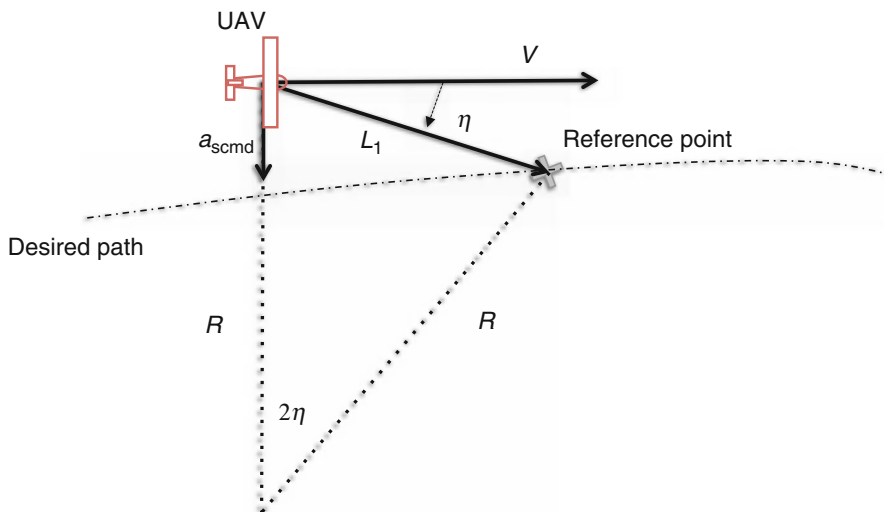


Fig. 27.16 Pure pursuit guidance logic (Park et al. 2004, 2007)

One way to track the trajectory is a UAV implementation of the “pure pursuit” (Amidi and Thorpe 1990; Ollero and Heredia 1995). The resulting guidance logic in Park et al. (2004, 2007) selects a reference point on the desired trajectory and uses it to generate a lateral acceleration command. This reference point is on the desired path at a distance (L_1) forward of the vehicle, as shown in Fig. 27.16. The algorithm then generates a lateral acceleration command that is determined by

$$a_{scmd} = 2 \frac{V^2}{L_1} \sin \eta. \quad (27.108)$$

The direction of acceleration depends on the sign of the angle between the L_1 line segment and the vehicle velocity vector. If the selected reference point is to the right of the vehicle velocity vector, then the vehicle will be commanded to accelerate to the right (see Fig. 27.16), which means that the vehicle will tend to align its velocity direction with the direction of the L_1 line segment.

Asymptotic Lyapunov stability of the nonlinear guidance method is demonstrated when the UAV is following circular paths (for details, see Park et al. 2004, 2007). This stability analysis is also extended to show robust stability of the guidance logic in the presence of saturated lateral acceleration, which typically occur when there are limitations on the vehicle bank angle.

Further insights on the approach can be obtained through linearization. If d is the cross-track error and V is the vehicle nominal speed, then, assuming η is small,

$$a_{scmd} = 2 \frac{V^2}{L_1} \sin \eta \approx 2 \frac{V}{L_1} \left(\dot{d} + \frac{V}{L_1} d \right). \quad (27.109)$$

Thus, linearization yields a PD controller for the cross-track error. The result also shows that the vehicle speed V and look-ahead distance L_1 determine the proportional and derivative gains. Note that the nonlinear form in Eq. 27.108 works for all η and thus is more powerful than a simple PD implementation.

Several other nonlinear approaches have been proposed for UAV trajectory tracking. For example, Nelson et al. (2006) describes an approach using vector fields to represent desired ground track headings so as to direct the vehicle onto the desired path. In Lawrence (2003), a Lyapunov approach is used to control the vehicle velocity vector to ensure convergence to a limit cycle. The method is well suited to guiding a vehicle from any initial position to a circular orbit above a target location. In Niculescu (2001), an approach that can accommodate large cross-track or heading deviations from a straight line path between way points is presented.

In contrast to the complexity of these approaches, the guidance law in Eq. 27.108 is very simple and easy to implement. In particular, some key properties of this algorithm are as follows (Park et al. 2007). It can naturally follow any circular path of radius greater than a specific limit imposed by the dynamics of the vehicle. It has an element of anticipation of the desired flight path, enabling tight tracking of curved flight trajectories. By incorporating the instantaneous vehicle speed, it adds an adaptive feature with respect to changes in vehicle ground speed caused by external disturbances, such as wind. The approach is proven to be asymptotically stable, for all velocities, in the entire state space of useful initial conditions and in the presence of saturation limits on lateral acceleration.

The implementation typically follows the successive loop-closure ideas discussed in Sect. 27.3. For example, one can implement this command by assuming that vehicle maintains sufficient lift to balance weight, even though banked at angle ϕ . This requires that the vehicle speed up and/or change to a larger α . Assume that $L \sin \phi = ma_s$ to achieve the desired lateral acceleration and that $L \cos \phi = mg$, then $\tan \phi = a_s/g$. The requisite lift increment is

$$\Delta C_L = \frac{L - mg}{QS} \equiv (n - 1) \frac{mg}{QS}, \quad (27.110)$$

where $n \equiv L/(mg)$ is the *load factor*. The expression $\tan \phi = a_s/g$ can be used to develop the required bank angle command ϕ_d that must be applied to the roll controller in the successive loop approach (similar to the architecture in Fig. 27.13). Depending on the vehicle, the command ϕ_d is typically saturated at $\pm 15^\circ$.

Note that this algorithm is designed to follow a given path. It can be used to fly between waypoints simply by connecting the waypoints with either straight lines or arcs, but additional algorithms are needed to generate the paths and/or waypoints. For example, Richards and How (2002) and Richards and How (2004) develop a waypoint optimization approach using mixed-integer linear programming that was demonstrated using UAVs (King et al. 2006). Cowlagi and Tsiotras (2009), Zhao and Tsiotras (2010), and Cowlagi and Tsiotras (2012) develop an alternative

approach based on optimal control theory. In Zhao and Tsiotras (2010) the authors use the kinematic model described in Sect. 27.2.6 along with state and control constraints to formulate and solve an optimal control problem to create time-optimal trajectories.

27.5 Conclusion and Future Directions

Linear control provides a very powerful set of procedures for controller, a vehicle near trim conditions. As shown in this chapter, there are numerous choices to be made, with the primary being classical/PID versus state space. Classical techniques are useful for relatively simple systems (low-state dimension). This is often the case for most airplanes in that the full dynamics decouple into the longitudinal and lateral dynamics and then can be further decomposed into even simpler models associated with the primary dynamic modes (e.g., phugoid, short period, spiral, dutch roll (Nelson and Smith 1989)). UAVs, with rotorcraft or other less traditional planforms, might not exhibit equivalent decompositions, but this option should be considered. If there remains significant coupling in the dynamics, then the large number of modes (poles) can greatly complicate a classical design, and a state space approach should be considered. Two state space options were presented: LQG and \mathcal{H}_∞ , both of which consider different performance objectives. The particular challenge with the LQG (and to some extent LQR) and \mathcal{H}_∞ control systems is determining the system dynamics with sufficient accuracy to ensure that the controllers are robust and work well in practice. This can often require extensive modeling, wind tunnel testing, and system identification. Robust control approaches exist if necessary, but the design procedures are quite complicated compared to those shown here.

With all of the design approaches, the fundamental challenge is to determine what are the design objectives (i.e., the pole locations in classical design, the R_{xx} and R_{uu} matrices in LQR, and the $W_s(s)$ and $W_u(s)$ in \mathcal{H}_∞ control). It should be noted that well-studied guidance and control strategies for manned aircraft may be overly conservative for unmanned vehicles. Choosing these performance specifications can be an iterative process requiring many tests to achieve good/robust overall vehicle performance, especially for a controller with many nested loops. Typically, the tendency with many control designs is to push the performance as high as possible. This is certainly true for the inner loops of the successive loop architecture in Fig. 27.13. Actuator magnitude (and rate) saturation are then natural limits on the achievable performance and should be modeled accurately to test the stability of the system. Excessive feedback of sensor noise and bias can then often be a problem, as can fatigue of the vehicle actuators if the control gains are very high.

As seen in Sect. 27.2, UAV dynamics are inherently nonlinear. Therefore, any linear control approach can only be guaranteed to be locally stable. Hence, it may be difficult to extract the desired performance or even guarantee stability when agile maneuvers are performed. For traditional fixed wing configurations executing way-point flight, this is usually not a significant problem due to vehicle symmetry and inherent stability. However, for rotorcraft platform and other non-

traditional configurations, it may be difficult to guarantee satisfactory performance with a single linear controller. Nonlinear and adaptive control techniques must also be evoked when the linear models of the UAV are not exactly known or are time varying. The problem of design and analysis of nonlinear controllers is studied in the companion chapter on nonlinear flight control techniques for UAVs (chapter ► [Nonlinear Flight Control Techniques for Unmanned Aerial Vehicles](#)).

References

- O. Amidi, C. Thorpe, Integrated mobile robot control. *Proc SPIE* **1388**, 505–523 (1990)
- G.J. Balas, J.C. Doyle, K. Glover, A. Packard, R. Smith, *μ -Analysis and Synthesis Toolbox* (The Math Works, Natick, 1995)
- G.J. Balas, A.K. Packard, J. Renfrow, C. Mullaney, R.T. M'Closkey, Control of the f-14 aircraft lateral-directional axis during powered approach. *J. Guid. Control Dyn.* **21**(6), 899–908 (1998)
- J.H. Blakelock, *Automatic Control of Aircraft and Missiles* (Wiley, New York, 1965)
- A.E. Bryson, Y.-C. Ho, *Applied Optimal Control* (Blaisdell Publishing Company, Waltham, 1969)
- H. Chao, Y. Cao, Y.Q. Chen, Autopilots for small fixed-wing unmanned air vehicles: a survey, in *International Conference on Mechatronics and Automation, 2007. ICMA 2007* (IEEE, Piscataway, 2007), pp. 3144–3149
- N.A. Chaturvedi, A.K. Sanyal, N.H. McClamroch, Rigid-body attitude control. *IEEE Control Syst.* **31**(3), 30–51 (2011)
- C.T. Chen, *Linear System Theory and Design* (Saunders College Publishing, Fort Worth, 1984)
- R.Y. Chiang, M.G. Safonov, *Robust Control Toolbox for Use with MATLAB®: User's Guide* (MathWorks, Natick, 1997). Incorporated
- G. Chowdhary, R. Jategaonkar, Parameter estimation from flight data applying extended and unscented kalman filter. *J. Aerosp. Sci. Technol.* (2009). doi:10.1016/j.ast.2009.10.003
- G. Chowdhary, E.N. Johnson, R. Chandramohan, M.S. Kimbrell, A. Calise, H. Jeong, Autonomous guidance and control of an airplane under severe damage, in *AIAA Infotech@Aerospace*, 2011. AIAA-2011-1428
- R.V. Cowlagi, P. Tsiotras, Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning, in *Proceedings of the American Control Conference*, St. Louis, 2009, pp. 414–419
- R.V. Cowlagi, P. Tsiotras, Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles. *IEEE Trans. Robot.* **28**(2), 379–395 (2012)
- J.L. Crassidis, J.L. Junkins, *Optimal Estimation of Dynamic Systems* (Chapman & Hall/CRC, Boca Raton, 2004)
- M. Cutler, J.P. How, Actuator constrained trajectory generation and control for variable-pitch quadrotors, in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Minneapolis, Aug 2012. (submitted)
- R.C. Dorf, R.H. Bishop, *Modern Control Systems*. (Addison-Wesley, Reading, 1995)
- J.C. Doyle, Guaranteed margins for LQG regulators. *IEEE Trans. Autom. Control* **AC-23**(4), 756–757 (1978)
- B. Etkin, *Dynamics of Flight: Stability and Control*, 2nd edn. (Wiley, New York, 1982)
- B. Etkin, L.D. Reid, *Dynamics of Flight, Stability and Control* (Wiley, New York, 1996)
- G.F. Franklin, J.D. Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems* (Addison-Wesley, Reading, 1994)
- J. Gadewadikar, F.L. Lewis, K. Subbarao, K. Peng, B.M. Chen, H-infinity static output-feedback control for rotorcraft. *J. Intell. Robot. Syst.* **54**(4), 629–646 (2009)
- P. Gahinet, A. Nemirovskii, A.J. Laub, M. Chilali, The lmi control toolbox, in *IEEE Conference on Decision and Control (CDC)*, vol. 3 (IEEE, New York, 1994), pp. 2038–2041
- A. Gelb, *Applied Optimal Estimation* (MIT (QA402.A5), Cambridge, 1974)

- M. Green, D.J. Limbeer, *Linear Robust Control* (Prentice Hall, Englewood Cliffs, 1995)
- J.K. Hall, N.B. Knoebel, T.W. McLain, Quaternion attitude estimation for miniature air vehicles using a multiplicative extended kalman filter, in *IEEE/ION Position, Location and Navigation Symposium*, Monterey, May 2008, pp. 1230–1237
- J. Hu, C. Bohn, H.R. Wu, Systematic h [infinity] weighting function selection and its application to the real-time control of a vertical take-off aircraft. *Control Eng. Pract.* **8**(3), 241–252 (2000)
- R.A. Hyde, *H-Infinity Aerospace Control Design- A VSTOL Flight Application*. (Springer, Berlin/New York, 1995)
- E. Johnson, S. Kannan, Adaptive trajectory control for autonomous helicopters. *J. Guid. Control Dyn.* **28**(3), 524–538 (2005)
- E. King, Y. Kuwata, J.P. How, Experimental demonstration of coordinated control for multi-vehicle teams. *Int. J. Syst. Sci.* **37**(6), 385–398 (2006)
- N. Knoebel, Adaptive control of a miniature tailsitter UAV. Master's thesis, Brigham Young University, Dec 2007
- J.B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality* (Princeton University Press, Princeton, 2002)
- B.C. Kuo, *Automatic Control Systems* (Prentice-Hall, Englewood Cliffs, 1991)
- H. Kwakernaak, R. Sivan, *Linear Optimal Control Systems* (Wiley, New York, 1972)
- D.A. Lawrence, Lyapunov vector fields for UAV flock coordination, in *2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations – Aerospace, Land, and Sea Conference, Workshop and Exhibition*, San Diego, 2003
- D. Lawrence, E.W. Frew, J.W. Pisano, Vector fields for autonomous unmanned aircraft flight control. *J. Guid. Control* **31**(5), 1220–1229 (2008)
- W. Levine (ed.), *The Control Handbook* (CRC, Boca Raton, 1996)
- F.L. Lewis, *Optimal Estimation* (Wiley, New York, 1986)
- J.M. Maciejowski, *Multivariable Feedback Design* (Addison-Wesley, Reading, 1989)
- C.G. Mayhew, R.G. Sanfelice, A.R. Teel, On quaternion-based attitude control and the unwinding phenomenon, in *American Control Conference (ACC)*, 2011, San Francisco, 29 June 2011–1 July 2011, pp. 299–304
- D. McFarlane, K. Glover, A Loop Shaping Design Procedure Using H_∞ Synthesis. *IEEE Trans. Autom. Control* **37**(6), 759–769 (1992)
- B. Mettler, *Modeling Identification and Characteristics of Miniature Rotorcrafts* (Kluwer, Boston, 2003)
- B. Michini, Modeling and adaptive control of indoor unmanned aerial vehicles. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, Sept 2009
- D.J. Moorhouse, R.J. Woodcock, Background Information and User Guide for MIL-F-8785C, Military Specification-Flying Qualities of Piloted Airplanes. Technical report, Wright-Patterson AFB, OH, 1982
- R.C. Nelson, S.E. Smith, *Flight Stability and Automatic Control* (McGraw-Hill, New York, 1989)
- D.R. Nelson, D.B. Barber, T.W. McLain, R.W. Beard, Vector field path following for small unmanned air vehicles, in *American Control Conference (ACC)*, Minneapolis, June 2006
- M. Niculesu, Lateral track control law for aerosonde UAV, in *Proceedings of the AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Jan 2001
- A. Ollero, G. Heredia, Stability analysis of mobile robot path tracking, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, 1995, pp. 461–466
- H. Özbay, *Introduction to Feedback Control Theory* (CRC, Boca Raton, 2000)
- S. Park, J. Deyst, J. P. How, A new nonlinear guidance logic for trajectory tracking, in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Providence, Aug 2004 (AIAA 2004-4900)
- S. Park, J. Deyst, J.P. How, Performance and lyapunov stability of a nonlinear path-following guidance method. *AIAA J. Guid. Control Dyn.* **30**(6), 1718–1728 (2007)

- J. Reiner, G. Balas, W. Garrard, Design of a flight control system for a highly maneuverable aircraft using mu synthesis, in *AIAA Guidance, Navigation and Control Conference*, Monterey, 1993, pp. 710–719
- A. Richards, J.P. How, Aircraft trajectory planning with collision avoidance using mixed integer linear programming, in *American Control Conference (ACC)*, Anchorage, vol. 3, 2002, pp. 1936–1941
- A. Richards, J.P. How, Decentralized model predictive control of cooperating UAVs, in *IEEE Conference on Decision and Control (CDC)*, Paradise Island, Dec 2004, pp. 4286–4291
- M. Sadraey, R. Colgren, Two dof robust nonlinear autopilot design for a small uav using a combination of dynamic inversion and loop shaping, in *AIAA Guidance Navigation and Control Conference*, San Francisco, vol. 2, 2005, pp. 5518–5537
- D. Simon, *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches* (Wiley-Interscience, Hoboken, 2006)
- S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control – Analysis and Design* (Wiley, Hoboken, 2005)
- F. Sobolic, Agile flight control techniques for a fixed-wing aircraft. Master's thesis, MIT Department Of Aeronautics and Astronautics, 2009
- R.F. Stengel, *Optimal Control and Estimation* (Dover, New York, 1994)
- R.F. Stengel, *Flight Dynamics* (Princeton University Press, Princeton, 2004)
- B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation*, 2 edn. (Wiley, Hoboken, 2003)
- F.B. Tatom, S.R. Smith, Simulation of atmospheric turbulent gusts and gust gradients. *J. Aircr.* **19**(04), 264–271 (1981)
- B. Wie, P.M. Barba, Quaternion feedback for spacecraft large angle maneuvers. *AIAA J. Guid. Control Dyn.* **8**, 360–365 (1985)
- Y. Zhao, P. Tsiotras, Time-optimal parameterization of geometric path for fixed-wing aircraft, in *AIAA@Infotech conference*, Atlanta (AIAA, 2010). AIAA-2010-3352
- K. Zhou, J.C. Doyle, K. Glover, *Robust and Optimal Control* (Prentice-Hall, Englewood Cliffs, 1996)

Nonlinear Flight Control Techniques for Unmanned Aerial Vehicles

28

Girish Vinayak Chowdhary, Emilio Frazzoli, Jonathan P. How, and
Hugh Liu

Contents

| | | |
|--------|--|-----|
| 28.1 | Introduction | 578 |
| 28.2 | An Overview of Nonlinear Stability Theory | 579 |
| 28.2.1 | Stability of a Nonlinear Dynamical System | 580 |
| 28.3 | Lyapunov-Based Control | 584 |
| 28.3.1 | Gain Scheduling | 584 |
| 28.3.2 | Backstepping Control | 586 |
| 28.3.3 | Model Predictive Control (MPC) | 588 |
| 28.3.4 | Model Inversion-Based Control | 590 |
| 28.3.5 | Model Reference Adaptive Control | 594 |
| 28.3.6 | Approximate Model Inversion-Based MRAC | 598 |
| 28.4 | Model-Based Fault-Tolerant Control | 599 |
| 28.4.1 | Control Design for Maximum Tolerance Range Using Model-Based Fault-Tolerant Control | 600 |
| 28.5 | Ongoing Research | 603 |
| 28.5.1 | Integrated Guidance and Control Under Uncertainty | 604 |

G.V. Chowdhary (✉)

Department of Aeronautics and Astronautics Aerospace Controls Laboratory, Massachusetts
Institute of Technology Laboratory for Information and Decision Systems, Cambridge, MA, USA
e-mail: girishc@MIT.EDU

E. Frazzoli

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge,
MA, USA
e-mail: frazzoli@mit.edu

J.P. How

Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts
Institute of Technology, Cambridge, MA, USA
e-mail: jhow@mit.edu

H. Liu

University of Toronto UTIAS Flight Systems and Control Research Group, Toronto, ON, Canada
e-mail: liu@utias.utoronto.ca

| | | |
|------------|---|-----|
| 28.5.2 | Output Feedback Control in Presence of Non-Gaussian Noise and Estimation Errors | 605 |
| 28.5.3 | Metrics on Stability and Performance of UAV Controllers..... | 605 |
| 28.5.4 | Agile and Fault-Tolerant Control | 606 |
| References | | 606 |

Abstract

In order to meet increasing demands on performance and reliability of unmanned aerial vehicles, nonlinear and adaptive control techniques are often utilized. These techniques are actively being studied to handle nonlinear aerodynamic and kinematic effects, actuator saturations and rate limitations, modeling uncertainty, and time-varying dynamics. This chapter presents an overview of some tools and techniques used for designing nonlinear flight controllers for UAVs. A brief overview of Lyapunov stability theory is provided. Nonlinear control techniques covered include gain scheduling, model predictive control, backstepping, dynamic inversion-based control, model reference adaptive control, and model-based fault-tolerant control.

28.1 Introduction

UAV flight control systems provide enabling technology for the aerial vehicles to fulfill their flight missions, especially when these missions are often planned to perform risky or tedious tasks under extreme flight conditions that are not suitable for piloted operation. Not surprisingly, UAV flight control systems are often considered safety/mission critical, as a flight control system failure could result in loss of the UAV or an unsuccessful mission. The purpose of this chapter, and its companion chapter on linear control (chapter ►[Linear Flight Control Techniques for Unmanned Aerial Vehicles](#)), is to outline some well-studied linear control methods and their applications on different types of UAVs as well as their customized missions.

The federal inventory of UAVs grew over 40 times in the last decade (Gertler 2012). Most UAVs in operation today are used for surveillance and reconnaissance (S&R) purposes (U.S. Army UAS Center of Excellence 2010) and in very few cases for payload delivery. In these cases a significant portion of the UAVs in operation remains remotely piloted, with autonomous flight control restricted to attitude hold, non-agile waypoint flight, or loiter maneuvers. Linear or gain-scheduled linear controllers are typically adequate for these maneuvers. But in many future scenarios, collaborating UAVs will be expected to perform agile maneuvers in the presence of significant model and environmental uncertainties (Office of the Secretary of Defense 2005, 2007; Gertler 2012). As seen in the companion chapter on linear flight control (chapter ►[Linear Flight Control Techniques for Unmanned Aerial Vehicles](#)), UAV dynamics are inherently nonlinear. Thus, any linear control approach can only be guaranteed to be locally stable, and it may be difficult to extract the desired performance or even guarantee stability when agile maneuvers are performed or when operating in the presence of significant nonlinear effects. This is particularly true for rotorcraft and other nontraditional

fixed-wing planform configurations that might be developed to improve sensor field of view, or for agile flight. In these cases, nonlinear and adaptive control techniques must be utilized to account for erroneous linear models (e.g., incorrect representation of the dynamics or time-varying/state-varying dynamics), nonlinear aerodynamic and kinematic effects, and actuator saturations and rate limitations.

The chapter begins with an overview of Lyapunov stability theory in Sect. 28.2. An overview of Lyapunov-based control techniques including gain scheduling, backstepping, and model predictive control is presented in Sect. 28.3. Dynamic inversion-based techniques are introduced in Sect. 28.3.4. A brief overview of model reference adaptive control is presented in Sect. 28.3.5. The intent of this chapter is to provide an overview of some of the more common methods used for nonlinear UAV control. There are several methods and active research direction beyond those discussed here for UAV nonlinear control; two examples are sliding mode control (see, e.g., Levant et al. 2000; Shkolnikov and Shtessel 2001; Koyuncu et al. 2010) and reinforcement learning-based methods (Abbeel et al. 2010).

28.2 An Overview of Nonlinear Stability Theory

This section presents a brief overview of Lyapunov theory-based mathematical tools used in nonlinear stability analysis and control. A detailed treatment of Lyapunov-based methods can be found in Khalil (2002), Haddad and Chellaboina (2008), Slotine and Li (1991), and Krstić et al. (1995). A state space representation of UAV dynamics will be used. Let $t \in \mathbb{R}^+$ denote the time; then the state $x(t) \in D \subset \mathbb{R}^n$ is defined as the minimal set of variable required to describe a system. Therefore, the choice of state variables is different for each problem. For a typical UAV control design problem, the state consists of the position of the UAV, its velocity, its angular rate, and its attitude. The admissible control inputs are defined by $u(t) \in U \subset \mathbb{R}^m$ and typically consist of actuator inputs provided by elevator, aileron, rudder, and throttle for a fixed wing UAV. Detailed derivation of UAV dynamics can be found in chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book. For the purpose of this chapter, it is sufficient to represent the UAV dynamics in the following generic form:

$$\dot{x}(t) = f(x(t), u(t)). \quad (28.1)$$

The initial condition for the above system is $x(0) = x_0$, with $t = 0$ being the initial time. Note that if the dynamics are time varying, then time itself can be considered as a state of the system (see, e.g., Chap. 4 of Haddad and Chellaboina (2008)). In a state-feedback framework, the control input u is often a function of the states x . Therefore, for stability analysis, it is sufficient to consider the following unforced dynamical system:

$$\dot{x}(t) = f(x(t)). \quad (28.2)$$

The set of all states $x_e \in \mathbb{R}^n$ that satisfy the equation $f(x_e) = 0$ is termed as the set of equilibrium points. Equilibrium points are of great interest in study of nonlinear dynamical systems as they are a set of states in which the system can stay indefinitely if not disturbed. A general nonlinear system may have several equilibrium points or may have none at all. The rigid-body equations of motion of a UAV as studied in chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) have several equilibrium points. Since a simple linear transformation can move an equilibrium point to the origin in the state space, it is assumed in the following that the nonlinear dynamical system of Eq. 28.2 has an equilibrium at the origin, that is, $f(0) = 0$. The solution $x(t)$ to Eq. 28.2 is

$$x(t) = x(0) + \int_0^t f(x(t))dt, \quad x(0) = x_0. \quad (28.3)$$

A unique solution is guaranteed to exist over D if $f(x)$ is Lipschitz continuous over D , that is, for all x and all y within a bounded distance of x , $\|f(x) - f(y)\| \leq c\|x - y\|$ for some constant c (Khalil 2002; Haddad and Chellaboina 2008). Most UAVs also satisfy some condition on controllability (Nijmeijer and van der Schaft 1990) that guarantees the existence of an admissible control u that drives the state close to any point in D in finite time.

28.2.1 Stability of a Nonlinear Dynamical System

The stability of a dynamical system is closely related with the predictability or well-behavedness of its solution. Particularly, the study of stability of a dynamical system answers the question: How far the solution of a dynamical system would stray from the origin if it started away from the origin? The most widely studied concept of stability are those of Lyapunov stability (Haddad and Chellaboina 2008; Khalil 2002; Slotine and Li 1991; Isidori 1995).

Definition 28.1. The origin of the dynamical system of Eq. 28.2 is said to be Lyapunov stable if for every $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that if $\|x(0)\| \leq \delta$ then $\|x(t)\| \leq \epsilon$.

Definition 28.2. The origin of the dynamical system of Eq. 28.2 is said to be asymptotically stable if it is Lyapunov stable and if $\|x(t)\| \leq \delta$ then $\lim_{t \rightarrow \infty} x(t) = 0$.

Definition 28.3. The origin of the dynamical system of Eq. 28.2 is said to be exponentially stable if there exist positive constants α and β such that the solution $x(t)$ satisfies $\|x(t)\| \leq \alpha\|x(0)\|e^{-\beta t}$.

If the above definitions hold for all initial conditions $x(0) \in \mathbb{R}^n$, then the stability definitions are said to be global (assuming a unique solution exists everywhere on \mathbb{R}^n). If a dynamical system is not Lyapunov stable, then it is called unstable.

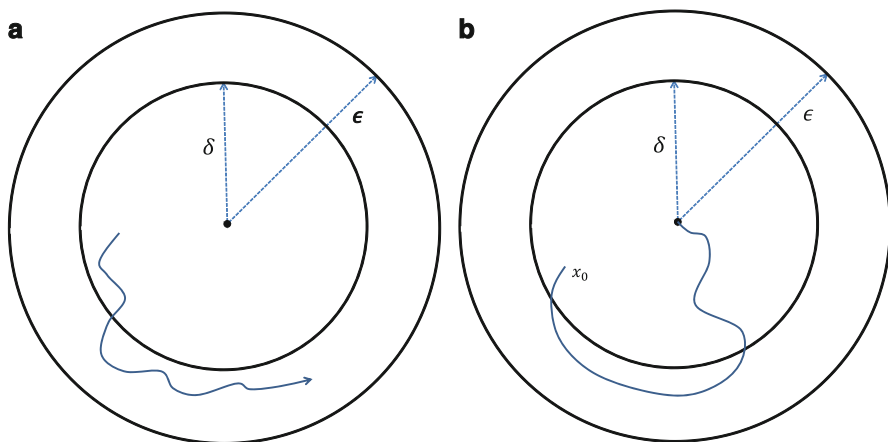


Fig. 28.1 A geometric depiction of Lyapunov stability concepts. (a) Depiction of a Lyapunov stable system. (b) Depiction of an asymptotically stable system

It follows from the above definitions that the stronger notion of stability is that of exponential stability, which requires that the solution goes to the origin at an exponential rate. This is different than the notion of asymptotic stability, which requires the solution to go to the origin eventually. The notion of exponential stability encompasses asymptotic stability, and the notion of asymptotic stability encompasses Lyapunov stability. A geometric depiction of Lyapunov stability is presented in Fig. 28.1.

Lyapunov's direct method, often referred to as Lyapunov's second method, is a powerful technique that provides sufficient conditions to determine the stability of a nonlinear system.

Theorem 28.1. (Khalil 2002; Haddad and Chellaboina 2008; Slotine and Li 1991) Consider the nonlinear dynamical system of Equation 28.2, and assume that there exists a continuously differentiable real-valued positive-definite function $V(x): D \rightarrow \mathbb{R}^+$ such that for $x \in D$

$$\begin{aligned} V(0) &= 0, \\ V(x) &> 0 \quad x \neq 0, \\ \frac{\partial V(x)}{\partial x} f(x) &\leq 0, \end{aligned} \tag{28.4}$$

then the origin is Lyapunov stable. If in addition

$$\frac{\partial V(x)}{\partial x} f(x) < 0, \tag{28.5}$$

the origin is asymptotically stable. Furthermore, if there exist positive constants $\alpha, \beta, \epsilon \in \mathbb{R}^+$ such that

$$\begin{aligned}\alpha \|x\|^2 &\leq V(x) \leq \beta \|x\|^2, \\ \frac{\partial V(x)}{\partial x} f(x) &\leq -\epsilon V(x),\end{aligned}\tag{28.6}$$

then the origin is exponentially stable.

The function $V(x)$ is said to be a Lyapunov function of the dynamical system in Eq. 28.2 if it satisfies these conditions. Therefore, the problem of establishing stability of a dynamical system can be reduced to that of finding a Lyapunov function for the system. If a system is stable, a Lyapunov function is guaranteed to exist (Slotine and Li 1991; Khalil 2002; Haddad and Chellaboina 2008), although finding one may not be always straightforward. It should be noted that the inability to find a Lyapunov function for a system does not imply its instability. A Lyapunov function is said to be radially unbounded if as $x \rightarrow \infty$, $V(x) \rightarrow \infty$. If a radially unbounded Lyapunov function exists for a dynamical system whose solution exists globally, then the stability of that system can be established globally. In the following, $\dot{V}(x(t))$ and $\frac{\partial V(x(t))}{\partial x(t)} f(x(t))$ are used interchangeably.

It is possible to relax the strict negative definiteness condition on the Lyapunov derivative in Eq. 28.5 if it can be shown that the Lyapunov function is nonincreasing everywhere, and the only trajectories where $\dot{V}(x) = 0$ indefinitely are the origin of the system. This result is captured by the Barbashin-Krasovskii-LaSalle invariance principle (see, e.g., Khalil 2002; Haddad and Chellaboina 2008; Slotine and Li 1991), which provides sufficient conditions for convergence of the solution of nonlinear dynamical system to its largest invariant set (described shortly). A point x_p is said to be a positive limit point of the solution $x(t)$ to the nonlinear dynamical system of Eq. 28.2 if there exists an infinite sequence $\{t_i\}$ such that $t_i \rightarrow +\infty$ and $x(t_i) \rightarrow x_p$, as $i \rightarrow \infty$. The positive limit set (also referred to as ω -limit set) is the set of all positive limit points. A set \mathcal{M} is said to be positively invariant if $x(0) \in \mathcal{M}$ then $x(t) \in \mathcal{M}$ for all $t \geq 0$. That is, a positively invariant set is the set of all initial conditions for which the solution of Eq. 28.2 does not leave the set. The set of all equilibria of a nonlinear dynamical system, for example, is a positively invariant set. The Barbashin-Krasovskii-LaSalle theorem can be stated as follows:

Theorem 28.2. (Haddad and Chellaboina 2008; Khalil 2002; Slotine and Li 1991) Consider the dynamical system of Equation 28.2, and assume that \bar{D} is a compact positively invariant set. Furthermore, assume that there exists a continuously differentiable function $V(x) : D \rightarrow \mathbb{R}$ such that $\frac{\partial V(x)}{\partial x} f(x) \leq 0$ in \bar{D} , and let \mathcal{M} be the largest invariant set contained in the set $S = \{x \in D | \dot{V}(x) = 0\}$. Then, if $x(0)$ in \bar{D} , the solution $x(t)$ of the dynamical system approaches \mathcal{M} as $t \rightarrow \infty$.

A corollary to this theorem states that if $V(x)$ is also positive definite over \bar{D} and no solution except the solution $x(t) = 0$ can stay in the set S , then the origin is asymptotically stable (Khalil 2002; Haddad and Chellaboina 2008).

In the following the quaternion attitude controller discussed in chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) is used to illustrate the application of Lyapunov stability analysis. A unit quaternion

$$q = (\overset{\circ}{q}, \vec{q}), \quad \overset{\circ}{q}^2 + \vec{q} \cdot \vec{q} = 1,$$

can be interpreted as a representation of a rotation of an angle $\theta = 2 \arccos(\overset{\circ}{q})$, around an axis parallel to \vec{q} . The attitude dynamics of an airplane modeled as a rigid body with quaternion attitude representation can be given by

$$\begin{aligned} J_B \dot{\omega}_B &= -\omega_B \times J_B \omega_B + u, \\ \dot{q} &= \frac{1}{2} q \circ (0, \omega_B), \end{aligned} \quad (28.7)$$

where J_B is the inertia tensor expressed in the body frame, $\omega_B = [\omega_1, \omega_2, \omega_3]^T \in \mathbb{R}^3$ is the angular velocity of the body frame with respect to the inertial frame, and u is the control input. The operator \circ denotes the quaternion composition operator, using which the quaternion kinematics Eq. 28.7 can be expanded as

$$q_1 \circ (0, \omega_B) = (-\vec{q}_1 \cdot \omega_B, \quad \overset{\circ}{q}_1 \omega_B + \overset{\circ}{q}_1 \vec{q}_1 + \vec{q}_1 \times \omega_B).$$

The goal is to design a control law u such that the system achieves the unique zero attitude represented by the unit quaternions $[1, 0, 0, 0]$ and $[-1, 0, 0, 0]$. It is shown that a control law of the form

$$u = -\frac{\overset{\circ}{q}}{2} K_p \vec{q} - K_d \omega_B, \quad (28.8)$$

where K_p and K_d are positive-definite gain matrices, guarantees that the system achieves zero attitude using Lyapunov analysis. Consider the Lyapunov candidate

$$V(q, \omega) = \frac{1}{2} \vec{q} \cdot K_p \vec{q} + \frac{1}{2} \omega \cdot J \omega.$$

Note that $V(q, \omega) \geq 0$ and is zero only at $q = (\pm 1, \vec{0})$, $\omega = 0$; both points correspond to the rigid body at rest at the identity rotation. The time derivative of the Lyapunov candidate along the trajectories of system Eq. 28.7 under the feedback Eq. 28.8 is computed as

$$\dot{V}(q, \omega) = \vec{q} \cdot K_p \dot{\vec{q}} + \omega \cdot J \dot{\omega}.$$

Note that from the kinematics of unit quaternions, $\dot{\vec{q}} = 1/2 (\dot{\vec{q}}\omega + \vec{q} \times \omega)$. Then,

$$\dot{V}(q, \omega) = \frac{1}{2} \vec{q} \cdot K_p (\dot{\vec{q}}\omega + \vec{q} \times \omega) - \omega \cdot \left(\omega \times J\omega + \frac{\dot{\vec{q}}}{2} K_p \vec{q} + K_d \omega \right) = -\omega \cdot K_d \omega \leq 0.$$

Therefore, $V(q, \omega) \leq 0$; furthermore, the set $S = \{x \in D \mid \dot{V}(x) = 0\}$ consists of only $q = (\pm 1, \vec{0})$, $\omega = 0$. Therefore, Theorem 28.2 guarantees asymptotic convergence to this set.

28.3 Lyapunov-Based Control

28.3.1 Gain Scheduling

In the companion chapter (chapter ►[Linear Flight Control Techniques for Unmanned Aerial Vehicles](#)), it was shown that aircraft dynamics can be linearized around equilibrium points (or trim conditions). A commonly used approach in aircraft control leverages this fact by designing a finite number of linear controllers, each corresponding to a linear model of the aircraft dynamics near a design trim condition. The key motivation in this approach is to leverage well-understood tools in linear systems design. Let A_i, B_i , $i \in \{1, \dots, N\}$ denote the matrices containing the aerodynamic and control effectiveness derivatives around the i th trimmed condition \bar{x}_i . Let X_1, \dots, X_N be a partition of the state space, that is, $\cup_{i=1}^N X_i = \mathbb{R}^n$, $X_i \cap X_j = \emptyset$ for $i \neq j$, into regions that are “near” the design trim conditions; in other words, whenever the state x is in the region X_i , the aircraft dynamics are approximated by the linearization at $\bar{x}_i \in X_i$. Then, the dynamics of the aircraft can be approximated as a state-dependent switching linear system as follows:

$$\dot{x} = A_i x + B_i u \quad \text{when } x \in X_i. \quad (28.9)$$

The idea in gain scheduling-based control is to create a set of gains K_i corresponding to each of the switched model and apply the linear control $u = K_i x$, this is depicted in Fig. 28.2. Contrary to intuition, however, simply ensuring that the i th system is rendered stable (i.e., the real parts of the eigenvalues of $A_i - B_i K_i$ are negative) is not sufficient to guarantee the closed-loop stability of Eq. 28.9 (Branicky 1998; Liberzon 2003, 2005). A Lyapunov-based approach can be used to guarantee the stability of the closed loop when using gain scheduling controller.

Consider the following Lyapunov candidate:

$$V(x(t)) = x(t)^T P x(t), \quad (28.10)$$

where P is a positive-definite matrix, that is, for all $x \neq 0$, $x^T P x > 0$. Therefore, $V(0) = 0$, and $V(x) > 0$ for all $x \neq 0$ making V a valid Lyapunov candidate. The derivative of the Lyapunov candidate is

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x}. \quad (28.11)$$

For the i th system, Eq. 28.11 can be written as

$$\dot{V}(x) = (A_i x - B_i K_i x)^T P x + x^T P (A_i x - B_i K_i x). \quad (28.12)$$

Let $\bar{A}_i = (A_i - B_i K_i)$; then from Lyapunov theory, it follows that for a positive-definite matrix Q if for all i

$$\bar{A}_i^T P + P \bar{A}_i < -Q, \quad (28.13)$$

$\frac{\partial V(x)}{\partial x} f(x) < -x^T Q x$. In this case, $V(x)$ is a common Lyapunov function for the switched closed-loop system (see, e.g., Liberzon 2003) establishing the (globally uniformly exponential) stability of the equilibrium at the origin. Therefore, the control design task is to select the gains K_i such that Eq. 28.13 is satisfied. One way to tackle this problem is through the framework of linear matrix inequalities (LMI) (Boyd et al. 1994; Gahinet et al. 1994). It should be noted that the condition in Eq. 28.13 allows switching between the linear models to occur infinitely fast; this can be a fairly conservative assumption for most UAV control applications. This condition can be relaxed to $\bar{A}_i^T P_i + P_i \bar{A}_i < -Q_i$ for $Q_i > 0$ to guarantee the stability of the system if the system does not switch arbitrarily fast. A rigorous condition for proving asymptotic stability of a system of the form (Eq. 28.9) was introduced in Branicky (1998). Let $V_i, i \in \{1, \dots, N\}$ be Lyapunov-like functions, that is, positive-definite functions such that $V_i(x) < 0$ whenever $x \in X_i \setminus \{0\}$. Define $V_i[k]$ as the infimum of all the values taken by V_i during the k -th time interval over which $x \in X_i$. Then, if the system satisfies the *sequence nonincreasing condition* $V_i[k+1] < V_i[k]$, for all $k \in \mathbb{N}$, asymptotic stability is guaranteed (Branicky 1998; Liberzon 2003).

The above example illustrates the use of Lyapunov techniques in synthesizing controllers. In general, given the system of Eq. 28.2, and a positive-definite Lyapunov candidate, the Lyapunov synthesis approach to create robust exponentially stable controllers can be summarized as follows: Let $\dot{V}(x) = g(x, u)$, and find a control function $u(x)$ such that $\dot{V}(x) < -\epsilon V(x)$ for some positive constant ϵ . Robust methods for Lyapunov-based control synthesis are discussed in Haddad and Chellaboina (2008). Furthermore, Rodrigues and How (2003) provides an output feedback control algorithms for system with switching dynamics.

The framework of linear parameter-varying (LPV) systems lends naturally to design and analysis of controllers based on a UAV dynamics representation via linearization across multiple equilibria. Gain scheduling-based LPV control synthesis techniques have been studied for flight control, and conditions for stability have been established (see, e.g., Lee et al. 2001; He and Zhao 2012; Balas et al. 2003; Shamma and Michael Athans 1991; Masubuchi et al. 2004).

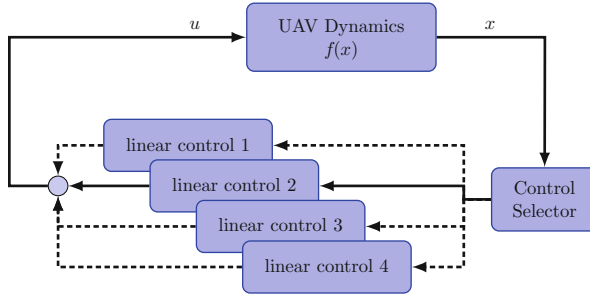


Fig. 28.2 Schematic of a gain-scheduled scheme for UAV control. The controller selector box decides which linear controller is to be used based on measured system states. In the depicted situation, controller 2 is active

28.3.2 Backstepping Control

Backstepping is an example of a Lyapunov-based technique, providing a powerful recursive approach for stabilizing systems that can be represented in nested loops (Kokotović 1992; Krstić et al. 1995).

28.3.2.1 Systems in Strict-Feedback Form

As a basic example, consider a system of the (strict-feedback) form

$$\begin{aligned}\dot{x} &= f_0(x) + g_0(x)z, \\ \dot{z} &= f_1(x, z) + g_1(x, z)u,\end{aligned}\tag{28.14}$$

where $x \in \mathbb{R}^n$, z is a scalar, and u is a scalar input. Assume that the “inner” system,

$$\dot{x} = f_0(x) + g_0(x)z,$$

has an equilibrium point for $x = 0$ and $z = 0$ and admits a known stabilizing feedback control law $x \mapsto u_0(x)$ with Lyapunov function $V_0 : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$, $V_0(0) = 0$. In other words, if $z = u_0(x)$, then

$$\frac{d}{dt}V_0(x) = \frac{\partial V_0(x)}{\partial x} (f_0(x) + g_0(x)u_0(x)) = -W(x) < 0, \quad \forall x \neq 0.$$

Now consider the system in Eq. 28.14; the idea is to design a control input u such that the state z tracks the control law u_0 , which is known to be stabilizing for the inner system. However, in general z will not match exactly $u_0(x)$; thus, stability for the whole closed-loop system must be established. Defining $e = z - u_0(x)$, Eq. 28.14 can be rewritten as

$$\begin{aligned}
\dot{x} &= f_0(x) + g_0(x)u_0(x) + g_0(x)e, \\
\dot{e} &= f_1(x, z) + g_1(x, z)u - \frac{\partial u_0(x)}{\partial x} (f_0(x) + g_0(x)z) = v.
\end{aligned} \tag{28.15}$$

Now consider the Lyapunov function candidate

$$V_1(x, e) = V_0(x) + \frac{1}{2}e^2;$$

its time derivative along the system's trajectories is

$$\frac{d}{dt} V_1(x, e) = \frac{\partial V_0(x)}{\partial x} (f_0(x) + g_0(x)u_0(x) + g_0(x)e) + ev.$$

If one picks $v = -\frac{\partial V_0(x)}{\partial x} g_0(x) - k_1 e$, with $k_1 > 0$, then

$$\frac{d}{dt} V_1(x, e) = -W(x) - k_1 e^2 < 0, \quad \forall (x, e) \neq 0,$$

thus proving stability of system Eq. 28.15.

Finally, assuming that $g_1(x, z) \neq 0$, $\forall (x, z) \in \mathbb{R}^{n+1}$, a stabilizing feedback for the original system in Eq. 28.14 can be recovered as

$$\begin{aligned}
u(x, z) &= \frac{1}{g_1(x, z)} \left(\frac{\partial u_0(x)}{\partial x} (f_0(x) + g_0(x)z) - f_1(x, z) + v \right) \\
&= \frac{1}{g_1(x, z)} \left(\frac{\partial u_0(x)}{\partial x} (f_0(x) + g_0(x)z) \right. \\
&\quad \left. - f_1(x, z) - \frac{\partial V_0(x)}{\partial x} g_0(x) - k_1(z - u_0(x)) \right).
\end{aligned} \tag{28.16}$$

Since the control law in Eq. 28.16 stabilizes system Eq. 28.14, with Lyapunov function V_1 , a similar argument can be used to recursively build a control law for a system in strict-feedback form of arbitrary order,

$$\begin{aligned}
\dot{x} &= f_0(x) + g_0(x)z_1, \\
\dot{z}_1 &= f_1(x, z_1) + g_1(x, z_1)z_2, \\
&\vdots \\
\dot{z}_m &= f_m(x, z_1, \dots, z_m) + g_m(x, z_m)u.
\end{aligned} \tag{28.17}$$

The procedure is summarized as follows: Start from the “inner” system, for which a stabilizing control law and a Lyapunov function are known. Define an error between the known control law and the actual input to the inner system. Augment the Lyapunov function with the square of the error. Design a new control law that

ensures the stability of the augmented system. Repeat using the augmented system just defined as the new inner system. This recursion, “stepping back” from the inner system all the way to the control input, gives the name to the method. A detailed description of control synthesis for small unmanned helicopters using backstepping techniques is provided in Raptis and Valavanis (2011).

28.3.3 Model Predictive Control (MPC)

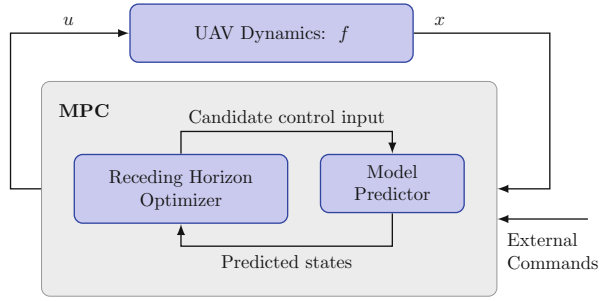
MPC has been successfully used for many years in industrial applications with relatively slow dynamics (e.g., chemical reactions (Maciejowski 2002; Kerrigan 2000; Mayne et al. 2000)); it is only in the past decade that the computational power has been available to enable online optimization for fast system dynamics typical in aerospace applications (early relevant demonstrations are in Manikonda et al. (1999), Dunbar and Murray (2002), Dunbar and Murray (2004), Dunbar (2007), Richards and How (2004), and Richards (2005)). MPC attempts to solve the following problem: Design an admissible piecewise continuous control input $u(t)$ that guarantees the system behaves like a reference model without violating the given state and input constraints. As such, a key benefit MPC is the ability to optimize the control input in the presence of state and input constraints. Furthermore, because MPC explicitly considers the operating constraints, it can operate closer to hard constraint boundaries than traditional control schemes.

MPC can be formulated using several types of cost function; however, due to the availability of robust quadratic solvers, the following formulation is popular: Find the optimal control $u(t)$ such that for given positive (semi)-definite matrices Q , R , and S , the following quadratic cost is minimized:

$$J_{\infty}(e, u) = \int_{t_0}^{\infty} e^T(t) Q e(t) + u^T(t) R u(t) dt, \quad x \in \Xi, u \in \Pi. \quad (28.18)$$

In the presence of constraints, a closed form solution for an infinite horizon optimization problem cannot be found in the general case (Camacho and Bordons 1999; Rawlings 2000; Nicolao et al. 2000; Mayne 2000). Hence, the approach in MPC is to numerically solve a receding horizon optimization (RHO) problem online over the interval $[t, t + N]$ to find the new control input at time $t + 1$; this process is then repeated over every discrete update (see, Fig. 28.3). The idea is that if the horizon is sufficiently large, the solution to the receding horizon optimization problem can guarantee stability. Let h be a mapping between the states x of the nonlinear dynamical system in Eq. 28.2 and the output z such that $z = h(x)$. If the function $h(x)$ is observable, this problem can be recast into a problem of minimizing a discrete output-based cost. Observability for linear systems was discussed in chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book. Local observability for nonlinear systems can be established in an analogous manner by considering the rank of the Jacobian matrix of $n - 1$ Lie derivatives of $h(x)$ along the trajectories of Eq. 28.2 (see, e.g., Kou et al. 1973; Nijmeijer and van der Schaft 1990). Typically, a quadratic cost function is preferred to leverage

Fig. 28.3 Schematic of a model predictive controller which minimizes a quadratic cost over a finite horizon



existing results in quadratic programming (Camacho and Bordons 1999; Rawlings 2000; Demircioglu and Karasu 2000). Let $z_{\text{rm}k}$ denote the sampled output of the reference model, and define the output error $\bar{e}_k = z_k - z_{\text{rm}k}$. Then the problem can be reformulated to finding the optimal sequence of inputs u_k such that the following quadratic cost function is minimized for given positive (semi)-definite matrices \bar{Q} and \bar{R} , subject to the reformulated output constraint $y \in \bar{\Xi}$, where $\bar{\Xi} = \{y : y = z_\sigma(x(t), u(t)), x \in \Xi\}$ and the input constraint $u \in \Pi$:

$$J_T(\bar{e}, u) = \sum_{k=t}^{t+N} \bar{e}_k^T \bar{Q} \bar{e}_k + u_k^T \bar{R} u_k + V_f(e_{t+N}), \quad y \in \bar{\Xi}, u \in \Pi, \quad (28.19)$$

where the term $V_f(e_{t+N})$ denotes a terminal penalty cost. Several computationally efficient nonlinear MPC algorithms have been proposed and their stability properties established (Zheng and Allgower 1998; Zheng 2000).

To numerically solve the RHO problem in Eq. 28.19, a prediction model $\hat{x} = \hat{f}(\hat{x})$ is required to predict how the system states behave in the future, where \hat{x} is the estimated state and \hat{f} is the prediction model. The prediction model is a central part of MPC, and in many cases (especially when the system dynamics are nonlinear or unstable), an inaccurate prediction model can result in instability (Rawlings 2000). In many MPC implementations, one of the most costly effort in control design is to develop a reliable prediction model (Camacho and Bordons 1999; Rawlings 2000; Lee 2000). Furthermore, approximations made in modeling, changes in the system dynamics due to wear and tear, reconfiguration of the system, or uncertainties introduced due to external effects can affect the accuracy of the predicted system response. Robustness of MPC methods to estimation errors (Findeisen et al. 2003; Michalska and Mayne 1995; Lee et al. 2002), plant variability (Chisci et al. 2001; Lee and Kouvaritakis 2000; Cuzzola et al. 2002; Richards 2005), and disturbances (Lee and Kouvaritakis 1999; Bemporad 1998; Kerrigan and Maciejowski 2001; Kerrigan and Mayne 2002; Richards and How 2007) remain active research areas. The resulting optimization problems are typically solved using linear matrix inequalities, linear programming, or quadratic programming. The key challenge here is to provide sufficient robustness guarantees while keeping the problem computationally tractable. The fact remains that without an accurate prediction model, the performance and stability guarantees remain very conservative (Rawlings 2000; Nicolao et al. 2000).

To this effect, some authors have recently explored adaptive MPC methods that estimate the modeling uncertainty (Camacho and Bordons 1999; Fukushima et al. 2007; Adetola et al. 2009; Karra et al. 2008). Another active area of research in MPC is that of analytically guaranteeing stability. It should be noted that stability and robustness guarantees for MPC when the system dynamics are linear are at least partially in place (Camacho and Bordons 1999). With an appropriate choice of $V_f(e_{t+N})$ and with terminal inequality state and input constraints (and in some cases without), stability guarantees for nonlinear MPC problems have been established (Nicolao et al. 2000; Zhang et al. 2010; Marruedo et al. 2002). However, due to the open loop nature of the optimization strategy, and dependence on the prediction model, guaranteeing stability and performance for a wide class of nonlinear systems still remains an active area of research. A challenge in implementing MPC methods is to ensure that the optimization problem can be solved in real time. While several computationally efficient nonlinear MPC strategies have been devised, ensuring that a feasible solution is obtained in face of processing and memory constraints remains an active challenge for MPC application on UAVs, where computational resources are often constrained (Sutton and Bitmead 2000).

28.3.4 Model Inversion-Based Control

28.3.4.1 Dynamic Model Inversion Using Differentially Flat Representation of Aircraft Dynamics

A complete derivation of the nonlinear six-degree-of-freedom UAV dynamics was presented in chapter ►[Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book. For trajectory tracking control using feedback linearization, simpler representations of UAV dynamics are often useful. One such representation is the differentially flat representation (see, e.g., Hauser and Hindman 1997). The dynamical system in Eq. 28.1 with output $y = h(x, u)$ is said to be differentially flat with flat output z if there exists a function g such that the state and input trajectories can be represented as a function of the flat output and a finite number of its derivatives:

$$(x, u) = g\left(y, \dot{y}, \ddot{y}, \dots, \frac{d^n y}{dt^n}\right). \quad (28.20)$$

In the following, assume that a smooth reference trajectory p_d is given for a UAV in the inertial frame (see chapter ►[Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book) and that the reference velocity \dot{p}_d and reference acceleration \ddot{p}_d can be calculated using the reference trajectory. Consider the case of a conventional fixed wing UAV that must be commanded a forward speed to maintain lift; hence, $\dot{p}_d \neq 0$. A right-handed orthonormal frame of reference wind axes can now be defined by requiring that the desired velocity vector is aligned with the x_W axis of the wind axis (see chapter ►[Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book for details), and the lift and drag forces are in the x_W – z_W plane of the wind axis, the z_W axis, such that there are no side forces. The acceleration can be written as

$$\ddot{p}_d = g + f_I/m, \quad (28.21)$$

where f_I is the sum of propulsive and aerodynamic forces on the aircraft in the inertial frame. Let R_{IW} denote the rotation matrix that transports vectors from the defined wind reference frame to the inertial frame. Then

$$\ddot{p}_d = g + R_{IW}a_W, \quad (28.22)$$

where a_W is the acceleration in the wind frame. Let $\omega = [\omega_1, \omega_2, \omega_3]^T$ denote the angular velocity in the wind frame, then the above equation can be differentiated to obtain

$$\frac{d^3 p_d}{dt^3} = R_{IW}(\omega \times a_W) + R_{IW}\dot{a}_W, \quad (28.23)$$

by using the relationship $\dot{R}_{IW}a_W = R_{IW}\hat{\omega}a_W = R_{IW}(\omega \times a_W)$ (see Sect. 27.2.2 of chapter [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book). Let a_t denote the tangential acceleration along the x_W direction, a_n denote the normal acceleration along the z_W direction, and V denote the forward speed. Furthermore, let $e_1 = [1, 0, 0]^T$. Then in coordinated flight $\dot{p}_d = VR_{IW}e_1$, hence

$$\ddot{p}_d = \dot{V}R_{IW}e_1 + VR_{IW}(\omega \times e_1). \quad (28.24)$$

Combining Eqs. 28.22 and 28.24, the following relationship can be formed:

$$\ddot{p}_d = R_{IW} \begin{bmatrix} \dot{V} \\ V\omega_3 \\ -V\omega_2 \end{bmatrix}. \quad (28.25)$$

Equation 28.25 allows the desired ω_2 and ω_3 to be calculated from the desired acceleration \ddot{p}_d . Furthermore, from Eq. 28.23,

$$\begin{bmatrix} \dot{a}_t \\ \omega_1 \\ \dot{a}_n \end{bmatrix} = \begin{bmatrix} -\omega_2 a_n \\ \omega_3 a_t / n \\ \omega_2 a_t \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1/a_n & 0 \\ 0 & 0 & 1 \end{bmatrix} R_{IW}^T \frac{d^3 p_d}{dt^3}. \quad (28.26)$$

The above equation defines a differentially flat system of aircraft dynamics with flat output p_d and inputs $[\dot{a}_t, \omega_1, \dot{a}_n]$, if $V = \|\dot{p}_d\| \neq 0$ (nonzero forward speed) and $a_n \neq 0$ (nonzero normal acceleration). The desired tangential and normal accelerations required to track the path p_d can be controlled through the thrust $T(\delta_T)$ which is a function of the throttle input δ_T , the lift $L(\alpha)$, and the drag $D(\alpha)$ which are functions of the angle of attack α by noting that in the wind axes:

$$a_t = T(\delta_T) \cos \alpha - D(\alpha), \quad (28.27)$$

$$a_n = -T(\delta_T) \sin \alpha - L(\alpha).$$

To counter any external disturbances that cause trajectory deviation, a feedback term can be added. Let p be the actual position of the aircraft and u be the control input, and consider a system in which $\frac{d^3 p_d}{dt^3} = u$:

$$\frac{d}{dt} \begin{bmatrix} p \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ \ddot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u. \quad (28.28)$$

Defining $e = p - p_d$, the above equation can be written in terms of the error:

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ \ddot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \left(u - \frac{d^3 p_d}{dt^3} \right). \quad (28.29)$$

Therefore, letting $u = \frac{d^3 p}{dt^3} - K [e \ \dot{e} \ \ddot{e}]^T$, where K is the stabilizing gain, and computing a_t, a_n, ω_1 from $(p, \dot{p}, \ddot{p}, u)$ guarantee asymptotic closed-loop stability of the system (see Hauser and Hindman 1997 for further detail of the particular approach presented). This approach is essentially that of feedback linearization and dynamic model inversion, which is explored in the general setting in the next section.

28.3.4.2 Approximate Dynamic Model Inversion

The idea in approximate dynamic inversion-based controllers is to use a (approximate) dynamic model of the UAV to assign control inputs based on desired angular rates and accelerations. Let $x(t) = [x_1^T(t), x_2^T(t)]^T \in \mathbb{R}^n$ be the known state vector, with $x_1(t) \in \mathbb{R}^{n_1}$ and $x_2(t) \in \mathbb{R}^{n_2}$; let $u(t) \in \mathbb{R}^{n_2}$ denote the control input; and consider the following multiple-input nonlinear uncertain dynamical system:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= f(x(t), u(t)), \end{aligned} \quad (28.30)$$

where the function f is assumed to be known and globally Lipschitz continuous, and control input u is assumed to be bounded and piecewise continuous. These conditions are required to ensure the existence and uniqueness of the solution to Eq. 28.30. Furthermore, a condition on controllability of f with respect to u must also be assumed. Note also the requirement on as many control inputs as the number of states directly affected by the input (x_2). For UAV control problem, this assumption can usually be met through the successive loop closure approach (see chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book). For example, for fixed wing control aileron, elevator, rudder, and throttle control directly affect roll, pitch, yaw rate, and velocity. This assumption can also meet for rotorcraft UAV velocity control with the attitudes acting as virtual inputs

for velocity dynamics and the three velocities acting as virtual inputs for the position dynamics (Johnson and Kannan 2005).

In dynamic model inversion-based control, the goal is to find the desired acceleration, referred to as the pseudo-control input $v(t) \in \mathbb{R}^{n_2}$, which can be used to find the control input u such that the system states track the output of a reference model. Let $z = (x, u)$, if the exact system model $f(z)$ in Eq. 28.30 is invertible; for a given $v(t)$, $u(t)$ can be found by inverting the system dynamics. However, since the exact system model is usually not invertible, let v be the output of an approximate inversion model \hat{f} such that $v = \hat{f}(x, u)$ is continuous and invertible with respect to u , that is, the operator $\hat{f}^{-1} : \mathbb{R}^{n+n_2} \rightarrow \mathbb{R}^l$ exists and assigns for every unique element of \mathbb{R}^{n+n_2} and a unique element of \mathbb{R}^l . An approximate inversion model that satisfies this requirement is required to guarantee that given a desired pseudo-control input $v \in \mathbb{R}^{n_2}$ a control command u can be found by dynamic inversion as follows:

$$u = \hat{f}^{-1}(x, v). \quad (28.31)$$

The model in Sect. 28.3.4.1 is an example of a differentially flat approximate inversion model. For the general system in Eq. 28.30, the use of an approximate inversion model results in a model error of the form

$$\dot{x}_2 = v + \Delta(x, u), \quad (28.32)$$

where Δ is the modeling error. The modeling error captures the difference between the system dynamics and the approximate inversion model:

$$\Delta(z) = f(z) - \hat{f}(z). \quad (28.33)$$

Note that if the control assignment function were known and invertible with respect to u , then an inversion model can be chosen such that the modeling error is only a function of the state x .

Often, UAV dynamics can be represented by models that are affine in control. In this case the existence of the approximate inversion model can be related directly to the invertibility of the control effectiveness matrix B (e.g., see Johnson 2000). For example, let $G \in \mathbb{R}^{n_2 \times n_2}$, and let $B \in \mathbb{R}^{n_2 \times l}$ denote the control assignment matrix and consider the following system:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= Gx_2(t) + B(\Theta(x) + u(t)), \end{aligned} \quad (28.34)$$

where $\Theta(x)$ is a nonlinear function. If $B^T B$ is invertible, and the pair (G, B) is controllable, one approximate inversion model is $v(t) = Bu(t)$, which results in a unique u for a unique v : $u(t) = (B^T B)^{-1} B^T v(t)$. Adding and subtracting $v = Bu$ yields Eq. 28.32, with $\Delta(x) = Gx_2 + B(\Theta(x) + u) - Bu = Gx_2 + B\Theta(x)$.

A reference model is used to characterize the desired response of the system:

$$\begin{aligned}\dot{x}_{1\text{rm}} &= x_{2\text{rm}}, \\ \dot{x}_{2\text{rm}} &= f_{\text{rm}}(x_{\text{rm}}, r),\end{aligned}\tag{28.35}$$

where $f_{\text{rm}}(x_{\text{rm}}(t), r(t))$ denote the reference model dynamics which are assumed to be continuously differentiable in x_{rm} for all $x_{\text{rm}} \in D_x \subset \mathbb{R}^n$. The command $r(t)$ is assumed to be bounded and piecewise continuous; furthermore, f_{rm} is assumed to be such that x_{rm} is bounded for a bounded reference input.

The pseudo-control input v is designed by combining a linear feedback part $v_{\text{pd}} = [K_1, K_2]e$ with $K_1 \in \mathbb{R}^{n_2 \times n_2}$ and $K_2 \in \mathbb{R}^{n_2 \times n_2}$, a linear feedforward part $v_{\text{rm}} = \dot{x}_{2\text{rm}}$, and an approximate feedback linearizing part $v_{\text{ai}}(z)$:

$$v = v_{\text{rm}} + v_{\text{pd}} - v_{\text{ai}}.\tag{28.36}$$

Defining the tracking error e as $e(t) = x_{\text{rm}}(t) - x(t)$ and using Eq. 28.32, the tracking error dynamics can be written as

$$\dot{e} = \dot{x}_{\text{rm}} - \begin{bmatrix} x_2 \\ v + \Delta \end{bmatrix}.\tag{28.37}$$

Letting $A = \begin{bmatrix} 0 & I_1 \\ -K_1 & -K_2 \end{bmatrix}$, $B = [0, I_2]^T$, where $0 \in \mathbb{R}^{n_2 \times n_2}$, $I_1 \in \mathbb{R}^{n_2 \times n_2}$, and $I_2 \in \mathbb{R}^{n_2 \times n_2}$ are the zero and identity matrices, and using Eq. 28.36 gives the following tracking error dynamics that are linear in e :

$$\dot{e} = Ae + B[v_{\text{ai}}(z) - \Delta(z)].\tag{28.38}$$

The baseline linear full-state feedback controller v_{pd} should be chosen such that A is a Hurwitz matrix. Furthermore, letting $v_{\text{ai}} = \Delta(z)$ using Eq. 28.33 ensures that the above tracking error dynamics is exponentially stable, and the states of the UAV track the reference model. This framework is depicted in (Fig. 28.4).

28.3.5 Model Reference Adaptive Control

The model reference adaptive (MRAC) control architecture has been widely studied for UAV control in presence of nonlinearities and modeling uncertainties (see, e.g., Johnson and Kannan 2005; Lavertsky and Wise 2005; Nguyen et al. 2006b; Patel et al. 2009). MRAC attempts to ensure that the controlled states track the output of an appropriately chosen reference model (see, e.g., Narendra and Annaswamy 1989; Ioannou and Sun 1996; Aström and Wittenmark 1995; Tao 2003). Most MRAC methods achieve this by using a parameterized model of the uncertainty, often referred to as the adaptive element and its parameters referred to as adaptive weights. Aircraft dynamics can often be separated into a linear part whose mathematical

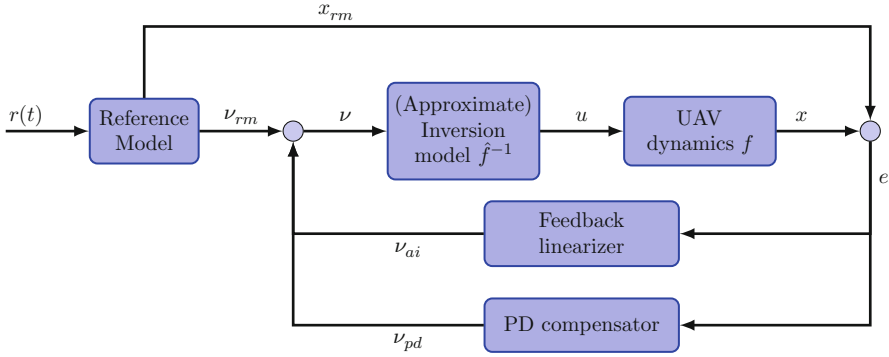


Fig. 28.4 Approximate dynamic model inversion framework using feedback linearization

model is fairly well known and an uncertain part that may contain unmodeled linear or nonlinear effects. This representation is also helpful in representing nonlinear external disturbances affecting the system dynamics. Therefore, one typical technique for implementing adaptive controllers is to augment a baseline linear controller, designed and verified using techniques discussed in chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book, with an adaptive controller that deals with nonlinearities and modeling uncertainties. Let $x(t) \in \mathfrak{R}^n$ be the known state vector, let $u \in \mathfrak{R}$ denote the control input, and consider a following system of this type:

$$\dot{x} = Ax(t) + B(u(t) + \Delta(z(t))), \quad (28.39)$$

where $A \in \mathfrak{R}^{n \times n}$, $B \in \mathfrak{R}^{n \times m}$, and $\Delta(x, u) \in \mathfrak{R}^m$ is a continuously differentiable function representing the uncertainty. It is assumed that the pair (A, B) is controllable (see chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book). For notational convenience, let $z = [x, u]$, and note that the function $\Delta(z)$ capturing the uncertain part of the dynamics is assumed to lie in the range space of the control effectiveness matrix B . More general formulations of adaptive control also assume that the matrix A is completely unknown (see, e.g., Narendra and Annaswamy 1989; Aström and Wittenmark 1995; Tao 2003; Ioannou and Sun 1996 and the chapter ► [Robust and Adaptive Control Methods for Aerial Vehicles](#) in this book).

A reference model can be designed that characterizes the desired response of the system

$$\dot{x}_{rm} = A_{rm}x_{rm}(t) + B_{rm}r(t), \quad (28.40)$$

where $A_{rm} \in \mathfrak{R}^{n \times n}$ is such that all of its eigenvalues are in the complex left-half plane, the pair (A, B) is controllable, and $r(t)$ denotes a bounded exogenous reference signal. These conditions are sufficient to guarantee that x_{rm} is bounded for a bounded reference signal $r(t)$. The tracking control law consists of a linear feedback part $u_{pd} = K(x_{rm}(t) - x(t))$, a linear feedforward part $u_{rm} = K_r[x_{rm}^T, r(t)]^T$, and an adaptive part $u_{ad}(x)$ and has the following form:

$$u = u_{rm} + u_{pd} - u_{ad}. \quad (28.41)$$

As before, define the tracking error $e(t) = x_{rm}(t) - x(t)$; with an appropriate choice of A_{rm} , B_{rm} , and u_{rm} such that $Bu_{rm} = (A_{rm} - A)x_{rm} + B_{rm}r(t)$ (conditions such as these are often required in MRAC and are referred to as *matching conditions*), the tracking error dynamics simplify to

$$\dot{e} = A_m e + B(u_{ad}(x, u) - \Delta(x, u)), \quad (28.42)$$

where the baseline full-state-feedback controller $u_{pd} = Kx$ is assumed to be designed such that $A_m = A - BK$ is a Hurwitz matrix. Hence, for any positive-definite matrix $Q \in \mathbb{R}^{n \times n}$, a positive-definite solution $P \in \mathbb{R}^{n \times n}$ exists to the Lyapunov equation

$$A_m^T P + P A_m = -Q. \quad (28.43)$$

MRAC architecture is depicted in Fig. 28.5. Several MRAC approaches (Narendra and Annaswamy 1989; Åström and Wittenmark 1995; Ioannou and Sun 1996; Tao 2003; Lavretsky 2009; Cao and Hovakimyan 2008) assume that the uncertainty $\Delta(z)$ can be linearly parameterized, that is, there exist a vector of constants $W = [w_1, w_2, \dots, w_m]^T$ and a vector of continuously differentiable functions $\Phi(z) = [\phi_1(z), \phi_2(z), \dots, \phi_m(z)]^T$ such that

$$\Delta(z) = W^* \Phi(z). \quad (28.44)$$

The case when the basis of the uncertainty is known, that is, the basis $\Phi(x)$ is known, has been referred to as the case of structured uncertainty (Chowdhary and Johnson

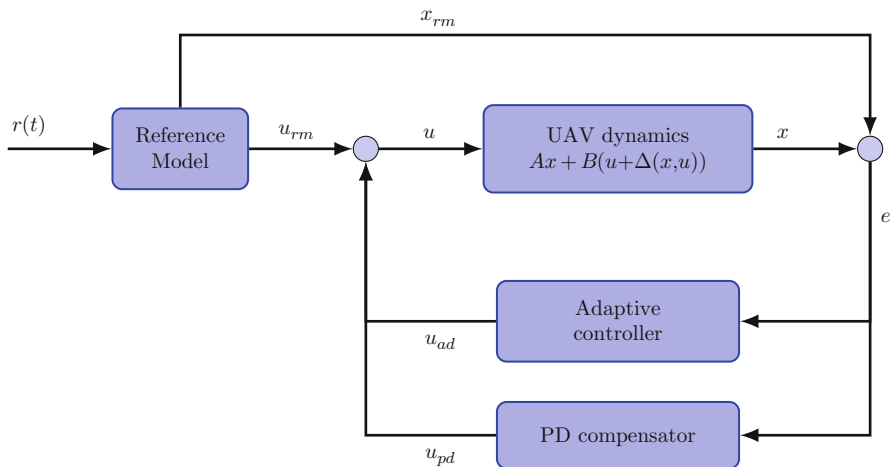


Fig. 28.5 Model reference control architecture that augments a baseline linear feedback-feedforward compensator

2010). In this case letting W denote the estimate W^* , the adaptive element is chosen as $u_{\text{ad}}(x) = W^T \Phi(x)$. For this case it is known that the adaptive law

$$\dot{W} = -\Gamma_W \Phi(z) e^T P B, \quad (28.45)$$

where Γ_W is a positive-definite learning rate matrix, results in $e(t) \rightarrow 0$. However, it should be noted that Eq. 28.50 does not guarantee the convergence (or even the boundedness) of W (Tao 2003; Narendra and Annaswamy 1987). A necessary and sufficient condition for guaranteeing $W(t) \rightarrow W$ is that $\Phi(t)$ be persistently exciting (PE) (Narendra and Annaswamy 1989; Ioannou and Sun 1996; Tao 2003; Boyd and Sastry 1986).

Several approaches have been explored to guarantee boundedness of the weights without needing PE; these include the classic σ -modification (Ioannou and Sun 1996), the e -modification (Narendra and Annaswamy 1989), the and projection-based adaptive control (Tao 2003; Pomet and Praly 1992). Let κ denote the σ -modification gain; then the σ -modification adaptive law is

$$\dot{W} = -\Gamma_W (\Phi(z(t)) e^T P B + \kappa W). \quad (28.46)$$

Thus, it can be seen that the goal of σ -modification is to add damping to the weight evolution. In e -modification, this damping is scaled by the norm of the error $\|e\|$. It should be noted that these adaptive laws guarantee the boundedness of the weights; however, they do not guarantee that the weights converge to their true values without PE.

A concurrent learning approach introduced in Chowdhary and Johnson (2010) guarantees exponential tracking error and weight convergence by concurrently using recorded data with current data without requiring PE. Other approaches that use recorded data include the Q -modification approach that guarantees convergence of the weights to a hyperplane where the ideal weights are contained (Volyanskyy et al. 2009), and the retrospective cost optimization approach (Santillo and Bernstein 2010). Other approaches to MRAC favor instantaneous domination of uncertainty over weight convergence, one such approach is L_1 adaptive control (Cao and Hovakimyan 2008). Several other approaches to MRAC also exist, including the composite adaptive control approach in which direct and indirect adaptive control are combined (Lavretsky 2009) and the observer-based reference dynamics modification in adaptive controllers in which transient performance is improved by drawing on parallels between the reference model and a Luenberger observer (Lavretsky 2011). The derivative-free MRAC in which a discrete derivative-free update law is used in a continuous framework (Yucelen and Calise 2010) and the optimal control modification (Nguyen 2010). Among approaches that deal with actuator time delays and constraints include the adaptive loop recovery modification which recovers nominal reference model dynamics in presence of time delays (Calise and Yucelen 2012), and the pseudo-control hedging which allows adaptive controllers to be implemented in presence of saturation (Johnson and Calise 2003).

In the more general case where the exact basis of the uncertainty is not a priori known, the adaptive part of the control law is often represented using a radial basis function (RBF) neural network (NN) (see, e.g., Sanner and Slotine 1992; Lewis 1999; Calise and Rysdyk 1998; Johnson and Kannan 2005; Chowdhary and Johnson 2011). For ease of notation let $z = [x, u]$; then the output of a RBF NN is given by

$$u_{\text{ad}}(z) = \hat{W}^T \sigma(z), \quad (28.47)$$

where $\hat{W} \in \Re^l$ and $\sigma = [1, \sigma_2(z), \sigma_3(z), \dots, \sigma_l(z)]^T$ is a vector of known radial basis functions. For $i = 2, 3, \dots, l$ let c_i denote the RBF centroid and μ_i denote the RBF width then for each RBF; then RBFs can be expressed as

$$\sigma_i(z) = e^{-\|z - c_i\|^2 / \mu_i}. \quad (28.48)$$

This approach relies on the universal approximation property of radial basis function neural networks (Park and Sandberg 1991) which asserts that given a fixed number of radial basis functions l , there exist ideal weights $W^* \in \Re^l$ and a real number $\tilde{\epsilon}$ such that

$$\Delta(z) = W^{*T} \sigma(z) + \tilde{\epsilon}, \quad (28.49)$$

where $\tilde{\epsilon}$ can be made arbitrarily small given sufficient number of radial basis functions. For this case, adaptive laws can be obtained by replacing $\Phi(z)$ in (Eq. 28.50) by $\sigma(z)$. For example, the adaptive law for an RBF NN adaptive element with σ -modification is

$$\dot{W} = -\Gamma_W (\sigma(z) e^T P B + \kappa W). \quad (28.50)$$

RBFs have gained significant popularity in adaptive control research because they are linearly parameterized. However, in practice, it is difficult to determine a priori how many RBFs should be chosen and where their centers should lie. Online adaptation and selection of RBF centers is an active research area (Sundararajan et al. 2002; Nardi 2000; Kingravi et al. 2012). Another approach that has been successful in practice is to use nonlinearly parameterized neural networks, including single hidden layer neural networks (see, e.g., Lewis 1999; Kannan 2005; Johnson and Kannan 2005; Chowdhary and Johnson 2011).

28.3.6 Approximate Model Inversion-Based MRAC

The key issue with the approximate model inversion scheme introduced in Sect. 28.3.4.2 is that the system dynamics $f(x, u)$ must be known in order to calculate v_{ai} to exactly cancel $\Delta(z)$. As this is often not the case, the following presents the approximate model inversion-based MRAC architecture (AMI-MRAC), which extends the dynamic model inversion-based control scheme of Sect. 28.3.4.2. Similar to the approach described there, the approach begins by choosing an approximate inversion model \hat{f} such that given a desired pseudo-control input

$v \in \mathbb{R}^{n_2}$, a control command u can be found by dynamic inversion as in Eq. 28.31. The parameters of the approximate inversion model do not have to be close to the real system model; however, in general, the mathematical structure should be similar to the real dynamical model. Typically, it is sufficient to map the inputs to the states that they directly affect; for example, it is sufficient to map the desired pitch rate to elevator deflection. Given the desired acceleration v (pseudo-control input) required to track the reference model, the control input can be found by $u = \hat{f}^{-1}(x, v)$. This results in a modeling error of the form

$$\Delta(z) = f(z) - \hat{f}(z). \quad (28.51)$$

The desired pseudo-control input can be formulated using the framework of dynamic model inversion as described in Sect. 28.3.4.2 as follows:

$$v = v_{rm} + v_{pd} - v_{ad}. \quad (28.52)$$

Note that the feedback linearizing part in (Eq. 28.36) has been replaced here with the output of an adaptive element v_{ad} . The tracking error dynamics of Eq. 28.37 now become

$$\dot{e} = Ae + B[v_{ad}(z) - \Delta(z)]. \quad (28.53)$$

This equation is similar to Eq. 28.42 of the MRAC architecture discussed in Sect. 28.3.5. Therefore, techniques similar to those discussed in Sect. 28.3.5 can be used. Particularly, if the uncertainty $\Delta(z)$ can be modeled using a linearly parameterized models such as in Eq. 28.44, then the adaptive element can take the form $v_{ad} = W(t)^T \Phi(z)$, and update law in Eq. 28.46 can be employed. It is also possible to use neural network adaptive elements in the same manner as Eq. 28.47. Note that with the formulation presented here the existence of a fixed point solution to $v_{ad} = \Delta(., v_{ad})$ needs to be assumed; sufficient conditions for guaranteeing this are also available (Kim 2003). AMI-MRAC-based adaptive controllers have been extensively flight-test verified on several fixed-wing and rotorcraft UAVs (see, e.g., Johnson and Kannan 2005; Johnson et al. 2006; Chowdhary and Johnson 2011; Chowdhary et al. 2012 and also the chapter ► [Adaptive Control of Unmanned Aerial Vehicles: Theory and Flight Tests](#) in this book). Figure 28.6 depicts the framework of an approximate model inversion-based MRAC architecture.

28.4 Model-Based Fault-Tolerant Control

Fault-tolerant controllers (FTC) are designed to maintain aircraft stability or reduce performance deterioration in case of failures (see, e.g., Patton and Chen 1997; Zhang and Jang 2008; Steinberg 2005). In general, there are mainly three types of failures (faults): sensor failure, actuator failure, and structural damage. Fault-tolerant control in the presence of sensor and actuator failures with identified failure characteristics has been studied (Patton and Chen 1997; Isermann 2006). On the other hand,

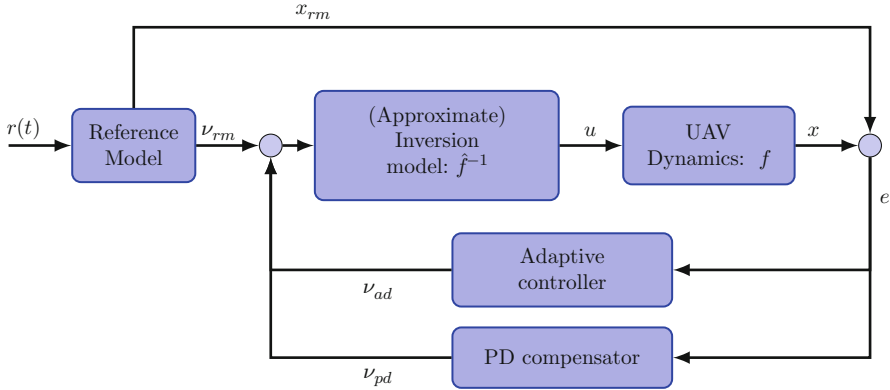


Fig. 28.6 Approximate dynamic model inversion based adaptive MRAC

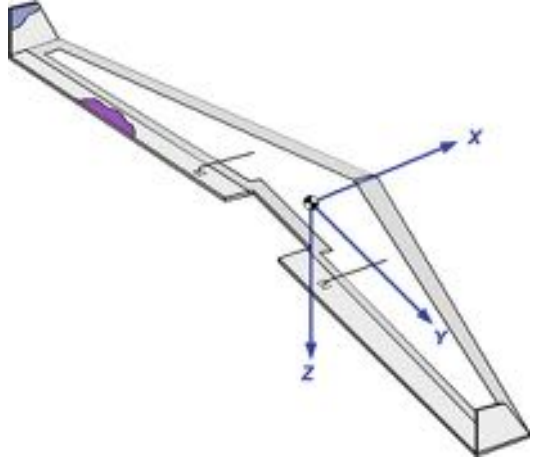
fault-tolerant control for structural damage, including partial loss of wing, vertical tail loss, horizontal tail loss, and engine loss, has been explored (Crider 2004; Hajiyev and Caliskan 2003; Haoouzi and Verhaegen 2008; Hitachi 2009; Lombaerts et al. 2008; Lunze and Steffen 2006; Nguyen et al. 2006a, 2008; Boskovic et al. 2007; Chowdhary et al. 2011).

Adaptive control has been widely studied for fault-tolerant control in the presence of modeling uncertainty brought about by structural damage or actuator degradation (Idan et al. 2002; Lavertsky and Wise 2005; Steinberg 2005; Kutay et al. 2008; Chowdhary et al. 2011; Calise and Yucelen 2012). The idea has been to use adaptive control techniques similar to those described in Sect. 28.3.5 to adapt to changing dynamics. Adaptive control-based fault-tolerant controllers do not require prior system identification or modeling efforts. Developing adaptive fault-tolerant controllers with associated quantifiable metrics for performance and without requiring restrictive *matching conditions* is an open area of research. Alternatively, a model-based approach can also be used for fault-tolerant control. An overview of one such approach is presented here.

28.4.1 Control Design for Maximum Tolerance Range Using Model-Based Fault-Tolerant Control

Fault-tolerant control technique presented here is applicable to aircraft with possible damage that lies approximately in the body $x-z$ plane of symmetry, in other words *vertical* tail damage as illustrated in Fig. 28.7. A linear model of aircraft dynamics was derived in chapter ►Linear Flight Control Techniques for Unmanned Aerial Vehicles in this book. In that model A represents the matrix containing aerodynamic derivatives and B represents the matrix containing the control effectiveness derivatives. The states of the model considered here are $x(t) = [u, w, q, \theta, v, p, r, \phi]^T$ and the input given by $u(t) = [\delta e, \delta f, \delta a, \delta r]^T$ (see chapter ►Linear Flight Control Techniques for Unmanned Aerial Vehicles in this book for definitions of these

Fig. 28.7 UAV with partial structural damage



variables; details of the corresponding A and B matrices can be found in Li and Liu 2012). Variation to aerodynamic derivatives due to structural damage is approximately proportional to the percentage of loss in structure (Shah 2008). Therefore, the linearized motion of the damaged aircraft can be represented by a parameter-dependent model based on the baseline linear model of the aircraft, where the parameters provide a notion of the degree of damage:

$$\dot{x}(t) = (A - \mu \bar{A})x(t) + (B - \mu \bar{B})u(t), \quad \mu \in [0, 1], \quad (28.54)$$

where μ is the parameter representing the *damage degree*. Specifically, $\mu = 0$ represents the case of no damage, $\mu = 1$ represents complete tail loss, and $0 < \mu < 1$ represents partial vertical tail loss. The damage loss can be related to a maximum change in the aerodynamic derivatives (e.g., $\Delta C_{y\beta}$) and control effectiveness derivatives (e.g., $\Delta C_{y\delta_r}$) under damage in the following way:

$$\begin{bmatrix} \Delta C_{y\beta} & \Delta C_{n\beta} & \Delta C_{l\beta} \\ \Delta C_{y_p} & \Delta C_{n_p} & \Delta C_{l_p} \\ \Delta C_{y_r} & \Delta C_{n_r} & \Delta C_{l_r} \\ \Delta C_{y\delta_r} & \Delta C_{n\delta_r} & \Delta C_{l\delta_r} \end{bmatrix} = \mu \begin{bmatrix} \Delta C_{y\beta}^{\max} & \Delta C_{n\beta}^{\max} & \Delta C_{l\beta}^{\max} \\ \Delta C_{y_p}^{\max} & \Delta C_{n_p}^{\max} & \Delta C_{l_p}^{\max} \\ \Delta C_{y_r}^{\max} & \Delta C_{n_r}^{\max} & \Delta C_{l_r}^{\max} \\ \Delta C_{y\delta_r}^{\max} & \Delta C_{n\delta_r}^{\max} & \Delta C_{l\delta_r}^{\max} \end{bmatrix} \quad (28.55)$$

For a given linear controller gain K , let $J(K, \mu)$ denote the performance metric of an aircraft with possible vertical tail damage as a function of the degree of damage μ . For example, a quadratic performance metric can be used:

$$J = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)] dt, \quad (28.56)$$

where Q and R are weighting positive-definite matrices. In extreme damage cases, such as complete tail loss ($\mu = 1$), the aircraft may not be able to recover.

The question then is under what level of damage a fault-tolerant control could still stabilize the aircraft. This notion of the maximum tolerance range, defined as the maximum allowable damage degree, presents a valuable design criterion in fault-tolerant control development. For most aircraft configurations, and without considering differential throttle as a way to control yaw motion, it is reasonable to expect that the maximum tolerance range would be less than 1. The notion of maximum tolerance, denoted by μ_m , is captured using the following relationship:

$$\mu_m := \min \left\{ 1, \max_K \{ \mu_u \geq 0 : J(K, \mu) \text{ is satisfied for } \mu \in [0, \mu_u] \} \right\}. \quad (28.57)$$

This definition indicates that for damage degree $0 \leq \mu \leq \mu_m \leq 1$, the aircraft control system is able to guarantee the desired level of performance J (see, e.g., Eq. 28.56) with a certain controller K . In particular, $\mu_m = 0$ means that there is no tolerance for the desired performance, while $\mu_m = 1$ means that the control strategy can guarantee the performance requirement up to a total loss of vertical tail. The bigger the μ_m is, the more tolerant the system becomes. Moreover, this notion implies a trade-off between damage tolerance and performance requirement. Since damage degree is unpredictable, a passive fault-tolerant strategy is to design a controller to maintain the expected performance under possible damage.

This can be achieved through a robust control design technique in which an upper bound is established for a linear quadratic cost function for all the considered uncertainty (Petersen et al. 2000; Magni et al. 1997). Consider the parameterized system of Eq. 28.54 describing damaged aircraft dynamics. Let $\Delta A = -\mu \bar{A}$ and $\Delta B = -\mu \bar{B}$ be uncertainty matrices expressed as $\begin{bmatrix} \Delta A & \Delta B \end{bmatrix} = DF \begin{bmatrix} E_1 & E_2 \end{bmatrix}$. In these expression, F satisfies $F^T F \leq I$. D and E are matrices containing the structural information of ΔA and ΔB and are assumed to be known a priori. For the given structure of vertical damage, $D = -I$, $F = \mu I$, $E_1 = \bar{A}$, and $E_2 = \bar{B}$.

One control design approach for the uncertain system in Eq. 28.54 is presented here based on LMI. If the following LMI with respect to a positive matrix X , matrix W , and positive scalar ε ($\varepsilon > 0$) is feasible (Petersen et al. 2000; Yu 2002),

$$\begin{bmatrix} AX + BW + XA^T + W^T B^T + \varepsilon DD^T X E_1^T + W^T E_2^T X (Q^{1/2})^T W^T (R^{1/2})^T & E_1 X + E_2 W & -\varepsilon I & 0 & 0 \\ Q^{1/2} X & 0 & 0 & -I & 0 \\ R^{1/2} W & 0 & 0 & 0 & -I \end{bmatrix} < 0, \quad (28.58)$$

then there exists a state feedback guaranteed cost control law $K = WX^{-1}$, and the corresponding cost has an upper bound $J \leq x_0^T X^{-1} x_0$.

As described above, one can formulate the aircraft dynamic model as an uncertain system by defining

$$\begin{bmatrix} \Delta A & \Delta B \end{bmatrix} = DF \begin{bmatrix} E_1 & E_2 \end{bmatrix} =: \mu_m (-\Delta) \begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} \text{ with } |\Delta| < 1.$$

The weighting matrices Q and R in the linear quadratic criterion in Eq. 28.56 can be chosen to represent a desired performance criterion. They also serve as free design parameters that indirectly reduce the effect of control surface limits on maximum tolerance and performance.

The above result gives a method to design a guaranteed cost controller. A controller with the maximum tolerance μ_m can be obtained by testing the feasibility of LMI via the bisection algorithm (Skogestad and Postlethwaite 2005; Zhou et al. 1996).

Algorithm:

1. Take $\mu_d = 0$ and $\mu_t = 1$. If the LMI (Eq. 28.58) is feasible for $\mu_t = 1$, then $\mu_m = 1$, output the corresponding controller and stop.
2. Let $\mu = \frac{\mu_d + \mu_u}{2}$.
3. If the LMI (Eq. 28.58) is feasible for μ , then $\mu_d = \mu$, otherwise, $\mu_t = \mu$.
4. If $\mu_t - \mu_d < \delta$ (δ is a predetermined scalar and is small enough), then $\mu_m = \mu_t$, output the controller with respect to μ_t and stop. Otherwise, go to 2.

28.5 Ongoing Research

Future UAV applications are expected to continue to leverage S&R capabilities of UAVs and go beyond. Some envisioned applications and some of the associated technical challenges in the field of autonomous control are listed below (see Fig. 28.8):

UCAV. Unmanned combat aerial vehicles (UCAV) have been researched extensively. The goal is to create an autonomous multi-role fighter capable of air-to-air combat and targeted munition delivery. UCAVs are expected to be highly agile and need to be designed to tolerate severe damage. Furthermore, the lack of human pilot onboard means that UCAVs can be designed to tolerate much higher flight loads in order to perform extremely agile maneuvers. The main technical challenge here is to create autonomous controllers capable of reliably operating in nonlinear flight regimes for extended durations.

Transport UAVs. These UAVs are expected to autonomously deliver valuable and fragile payload to forward operating units. The technical challenge here is to enable reliable automated waypoint navigation in presence of external disturbances and changed mass properties and reliable vertical takeoff and landing capabilities in harsh terrains with significant uncertainties.

HALE. High-altitude long-endurance (HALE) aircraft are being designed to support persistent S&R missions at stratospheric altitude. The main differentiating factor here is the use of extended wingspan equipped with solar panels for in situ power generation. The technical challenges here are brought about by unmodeled flexible body modes.

Optionally Manned Aircraft. Optionally manned aircraft are envisioned to support both piloted and autonomous modes. The idea is to extend the utility of

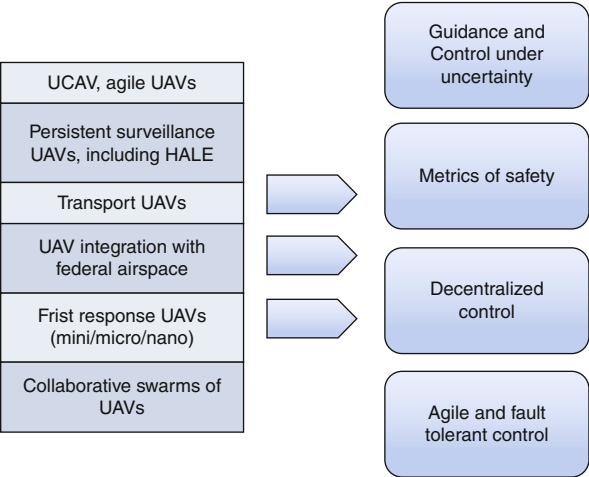


Fig. 28.8 Envisioned UAV applications and some related research thrusts

existing successful aircraft by enabling autonomous operation in absence of an onboard human pilot. The main technical challenge here is to ensure reliability and robustness to uncertainties, considering specifically that these aircraft may share airspace with manned aircraft.

Micro-/Nano-UAVs. UAVs with wingspan less than 30 cm are often referred to as micro-/nano-UAVs. These UAVs are characterized by high wing (or rotor blade) loading and are being designed, for example, to operate in the vicinity of human first responders. Some of these aircraft are expected to operate in dynamic or hostile indoor environments and are therefore required to be agile and low cost. Large-scale collaborative swarms of low-cost micro-UAVs have also been envisioned as flexible replacements to a single larger UAV. The key technical challenges here are creating low-cost, low-weight controllers capable of guaranteeing safe and agile operation in the vicinity of humans, and development of decentralized techniques for guidance and control.

In order to realize these visions, there is a need to develop robust and adaptive controllers that are able to guarantee excellent performance as the UAV performs agile maneuvers in nonlinear flight regime in presence of uncertainties. Furthermore, there is a significant thrust toward collaborative operation of manned and unmanned assets and toward UAVs sharing airspace with their manned counterparts. The predictable and adaptable behavior of UAVs as a part of such codependent networks is evermore important.

28.5.1 Integrated Guidance and Control Under Uncertainty

Guidance and control algorithms for autonomous systems are safety/mission critical. They need to ensure safe and efficient operation in presence of uncertainties

such as unmodeled dynamics, damage, sensor failure, and unknown environmental effects. Traditionally, control and guidance methods have often evolved separately. The ignored interconnections may lead to catastrophic failure if commanded trajectories end up exciting unmodeled dynamics such as unmodeled flexible dynamics (modes). Therefore, there is a need to establish a feedback between the guidance and command loops to ensure that the UAV is commanded a feasible trajectories. This is also important in order to avoid actuator saturation and particularly important if the UAV capabilities have been degraded due to damage (Chowdhary et al. 2010). Isolated works exist in the literature where authors have proposed a scheme to modify reference trajectories to accommodate saturation or to improve tracking performance (e.g., Johnson and Calise 2003; Lavretsky 2011); however, the results are problem specific and more general frameworks are required.

There has been significant recent research on UAV task allocation and planning algorithms. The algorithms developed here provide waypoints and reference commands to the guidance system in order to satisfy higher-level mission goals. There may be value in accounting specifically for the UAV's dynamic capabilities and health to improve planning performance.

28.5.2 Output Feedback Control in Presence of Non-Gaussian Noise and Estimation Errors

LQG theory provides a solution to designing optimal controllers for linear systems with Gaussian white measurement noise. However, such a generalized theory does not exist for nonlinear control. Furthermore, the choice of sensors on miniature UAVs in particular is often restricted to low-cost, low-weight options. Several of the standard sensors employed on miniature aircraft such as sonar altimeters, scanning lasers, and cameras do not have Gaussian noise properties. Therefore, even with an integrated navigation solution that fuses data from multiple sensors, the resulting state estimates may contain significant estimation error. One of the key future thrusts required therefore is a theory of output feedback control for nonlinear control implementations in presence of non-Gaussian noise.

28.5.3 Metrics on Stability and Performance of UAV Controllers

A strong theory exists for computing performance and stability metrics for linear time-invariant control laws. Examples of well-accepted metrics include gain and phase margin which quantify the ability of the controller to maintain stability in presence of unforeseen control effectiveness changes, time delays, and disturbances. However, these metrics do not easily generalize to nonlinear systems. On the other hand, several different groups have established that nonlinear control techniques in general can greatly improve command tracking performance over linear techniques. However, the inability to quantify stability and robustness of nonlinear controllers may pose a significant hurdle in wide acceptance of nonlinear techniques.

The problem is further complicated as controllers are implemented on digital computers while they are often designed using a continuous framework. Any further work in this area should avail the rich literature on the theory of sampled data systems.

28.5.4 Agile and Fault-Tolerant Control

Agile flight refers to flight conditions which cannot be easily represented by linear models linearized around equilibrium flight conditions. Agile flight is often characterized by rapid transitions between flight domains (e.g., hover flight domain and forward flight domain for rotorcraft UAVs), high wing (blade) loading, actuator saturation, and nonlinear flight regimes. The lack of a human pilot onboard means that the allowable g-force is not a limiting factor on the aggressive maneuvers a UAV platforms can potentially perform. Significant progress has been made in creating UAV autopilots that perform highly aggressive maneuvers. Authors in Gavrilets et al. (2004), Abbeel et al. (2007), Cutler and How (2012), and Johnson and Kannan (2005) have established control methods that can be used to perform specific agile maneuvers under test conditions. Future work in this area could lead to responsive agile flight maneuvers to meet higher-level mission requirements and quantifiable metrics of performance and stability during agile flight.

For some agile flight regimes, accurate modeling of the vehicle dynamics might be very difficult, requiring that models and/or control strategies be identified in real time, in essence pushing the capabilities discussed in Mettler (2003), Chowdhary and Lorenz (2005), Jategaonkar (2006), and Morelli (2000) to be online and for nonlinear models. In this case, model identification (or learning) combined with online optimization (e.g., model predictive control) (Favoreel et al. 1999; Ernst et al. 2006; Woodley et al. 2001) might provide an important direction to consider. The key challenge here is to simultaneously guarantee stability and performance while guaranteeing online learning of reasonable model approximations.

Fault-tolerant control (FTC) research for UAVs is focused on guaranteeing recoverable flight in presence of structural and actuator failures. Several directions exist in fault-tolerant flight control; excellent reviews can be found in Zhang and Jang (2008) and Steinberg (2005). Authors in Jourdan et al. (2010) and Chowdhary et al. (2012) have established through flight tests the effectiveness of fault-tolerant controllers in maintaining flight in presence of severe structural damage including loss of significant portions of the wing. Detection and diagnosis of structural faults is also being studied, along with detection of sensor anomalies.

References

- P. Abbeel, A. Coates, M. Quigley, A.Y. Ng, An application of reinforcement learning to aerobatic helicopter flight, in *Advances in Neural Information Processing Systems (NIPS)* (MIT, Cambridge/London, 2007), p. 2007

- P. Abbeel, A. Coates, A.Y. Ng, Autonomous helicopter aerobatics through apprenticeship learning. *Int. J. Robot. Res.* **29**(13), 1608–1639 (2010)
- V. Adetola, D. DeHaan, M. Guay, Adaptive model predictive control for constrained nonlinear systems. *Syst. Control Lett.* **58**(5), 320–326 (2009)
- K.J. Aström, B. Wittenmark, *Adaptive Control* (Addison-Wesley, Reading, 1995)
- G. Balas, J. Bokor, Z. Szabo, Invariant subspaces for lpv systems and their applications. *IEEE Trans. Autom. Control* **48**(11), 2065–2069 (2003)
- A. Bemporad, Reducing conservativeness in predictive control of constrained systems with disturbances, in *Proceedings of the 37th IEEE Conference on Decision and Control, 1998*, vol. 2, (IEEE, Piscataway, 1998), pp. 1384–1389
- J.D. Boskovic, R. Prasanth, R.K. Mehra, Retrofit fault-tolerant flight control design under control effector damage. *AIAA J. Guid. Control Dyn.* **30**(3), 703–712 (2007)
- S. Boyd, S. Sastry, Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica* **22**(6), 629–639 (1986)
- S. Boyd, E.G. Laurent, E. Feron, V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control*. Studies in Applied Mathematics (SIAM, Philadelphia, 1994)
- M.S. Branicky, Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Autom. Control* **43**(4), 475–482 (1998)
- A.J. Calise, R.T. Rysdyk, Nonlinear adaptive flight control using neural networks. *IEEE Control Syst. Mag.* **18**(6), 14–25 (1998)
- A.J. Calise, T. Yucelen, Adaptive loop transfer recovery. *J. Guid. Control Dyn.* **35**(3), 807–815 (2012). doi:10.2514/1.55835
- E.F. Camacho, C. Bordons, *Model Predictive Control* (Springer, London, 1999)
- C. Cao, N. Hovakimyan, Design and analysis of a novel adaptive control architecture with guaranteed transient performance. *IEEE Trans. Autom. Control* **53**(2), 586–591 (2008)
- L. Chisci, P. Falugi, G. Zappa, Predictive control for constrained systems with polytopic uncertainty, in *Proceedings of the 2001 American Control Conference, 2001*, vol. 4 (American Automatic Control Council, Evanston, 2001), pp. 3073–3078
- G. Chowdhary, E.N. Johnson, Concurrent learning for convergence in adaptive control without persistency of excitation, in *49th IEEE Conference on Decision and Control* (Institute of Electrical and Electronics Engineers, Piscataway, 2010)
- G. Chowdhary, E.N. Johnson, Theory and flight test validation of a concurrent learning adaptive controller. *J. Guid. Control Dyn.* **34**(2), 592–607 (2011)
- G. Chowdhary, S. Lorenz, Non-linear model identification for a miniature rotorcraft, preliminary results, in *American Helicopter Society 61st Annual Forum* (American Helicopter Society, 2005)
- G. Chowdhary, E.N. Johnson, S.M. Kimbrell, R. Chandramohan, A.J. Calise, Flight test results of adaptive controllers in the presence of significant aircraft faults, in *AIAA Guidance Navigation and Control Conference*, Toronto, Canada, 2010. Invited
- G. Chowdhary, E.N. Johnson, R. Chandramohan, M.S. Kimbrell, A. Calise, H. Jeong, Autonomous guidance and control of an airplane under severe damage, in *AIAA Infotech@Aerospace* (American Institute of Aeronautics and Astronautics, Reston, 2011). AIAA-2011-1428
- G. Chowdhary, E.N. Johnson, R. Chandramohan, S.M. Kimbrell, A. Calise, Autonomous guidance and control of airplanes under actuator failures and severe structural damage. *J. Guid. Control Dyn.* (2012, in-press)
- L.D. Crider, Control of Commercial Aircraft with Vertical Tail Loss, in *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, Chicago, IL, USA, 20–22 Sept. 2004. pp. 1–11 (2004)
- M. Cutler, J.P. How, Actuator constrained trajectory generation and control for variable-pitch quadrotors, in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Minneapolis, Minnesota, August 2012 (submitted)
- F.A. Cuzzola, J.C. Geromel, M. Morari, An improved approach for constrained robust model predictive control. *Automatica* **38**(7), 1183–1189 (2002)

- H. Demircioglu, E. Karasu, Generalized predictive control. a practical application and comparison of discrete- and continuous-time versions. *IEEE Control Syst. Mag.* **20**(5), 36–47 (2000)
- W. Dunbar, Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Robot.* **52**, 1249–1263 (2007)
- W.B. Dunbar, R. Murray, Model predictive control of coordinated multi-vehicle formations, in *IEEE Conference on Decision and Control (CDC)* (IEEE, 2002), pp. 4631–4636
- W.B. Dunbar, R.M. Murray, Receding horizon control of multi-vehicle formation: a distributed implementation, in *IEEE Conference on Decision and Control (CDC)* (2004)
- D. Ernst, M. Glavic, F. Capitanescu, L. Wehenkel, Model predictive control and reinforcement learning as two complementary frameworks, in *Proceedings of the 13th IFAC Workshop on Control Applications of Optimisation*, Cachan, France, 2006
- W. Favoreel, B. De Moor, P. Van Overschee, M. Gevers, Model-free subspace-based lqg-design, in *Proceedings of the 1999 American Control Conference, 1999*, vol. 5 (IEEE, Piscataway, 1999), pp. 3372–3376
- R. Findeisen, L. Imsland, F. Allgower, B.A. Foss, Output feedback stabilization of constrained systems with nonlinear predictive control. *Int. J. Robust Nonlinear Control* **13**, 211–227 (2003)
- H. Fukushima, T.-H. Kim, T. Sugie, Adaptive model predictive control for a class of constrained linear systems based on the comparison model. *Automatica* **43**(2), 301–308 (2007)
- P. Gahinet, A. Nemirovskii, A.J. Laub, M. Chilali, The lmi control toolbox, in *IEEE Conference on Decision and Control (CDC)*, vol. 3 (IEEE, Piscataway, 1994), pp. 2038–2041
- V. Gavrilits, B. Mettler, E. Feron, Human-inspired control logic for automated maneuvering of miniature helicopter. *AIAA J. Guid. Control Dyn.* **27**(5), 752–759 (2004)
- J. Gertler, U.S. unmanned aerial systems. Technical report, Congressional Research Service, January 2012. Prepared for Members and Committees of Congress (2012)
- W.M. Haddad, V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach* (Princeton University Press, Princeton, 2008)
- C. Hajiyeve, F. Caliskan, *Fault Diagnosis and Reconfiguration in Flight Control Systems* (Kluwer Academic, Boston, 2003)
- R. Haoouzi, M. Verhaegen, Fault-tolerant subspace predictive control applied to a boeing 747 model. *J. Guid. Control Dyn.* **31**(4) (2008)
- J. Hauser, R. Hindman, Aggressive flight maneuvers, in *Proceedings of the 36th Conference on Decision and Control*, San Diego, CA, December 1997 (IEEE, 1997)
- X. He, J. Zhao, Parameter-dependent h8 filter design for lpv systems and an autopilot application. *Appl. Math. Comput.* **218**(9), 5508–5517 (2012)
- Y. Hitachi, Damage-tolerant control system design for propulsion-controlled aircraft. Master's thesis, University of Toronto, 2009
- M. Idan, M.D. Johnson, A.J. Calise, A hierarchical approach to adaptive control for improved flight safety. *AIAA J. Guid. Control Dyn.* **25**(6), 1012–1020 (2002)
- P.A. Ioannou, J. Sun, *Robust Adaptive Control* (Prentice-Hall, Upper Saddle River, 1996)
- R. Isermann, *Fault-Diagnosis Systems: An Introduction from Detection to Fault Tolerance* (Springer, Berlin/New York, 2006)
- A. Isidori, *Nonlinear Control Systems*, 3rd edn. (Springer, New York, 1995)
- R.V. Jategaonkar, *Flight Vehicle System Identification a Time Domain Approach*. Volume 216 of Progress in Astronautics and Aeronautics (American Institute of Aeronautics and Astronautics, Reston, 2006)
- E.N. Johnson, Limited authority adaptive flight control. PhD thesis, Georgia Institute of Technology, Atlanta GA, 2000
- E.N. Johnson, A.J. Calise, Limited authority adaptive flight control for reusable launch vehicles. *AIAA J. Guid. Control Dyn.* **26**(6), 906–913 (2003)
- E. Johnson, S. Kannan, Adaptive trajectory control for autonomous helicopters. *J. Guid. Control Dyn.* **28**(3), 524–538 (2005)
- E. Johnson, M. Turbe, A. Wu, S. Kannan, Flight results of autonomous fixed-wing uav transitions to and from stationary hover, in *Proceedings of the AIAA GNC Conference*, August 2006 (AIAA, 2006)

- D.B. Jourdan, M.D. Piedmonte, V. Gavrillets, D.W. Vos, *Enhancing UAV Survivability Through Damage Tolerant Control*. Number August (AIAA, 2010) pp. 1–26. AIAA-2010-7548
- S. Kannan, Adaptive control of systems in cascade with saturation. PhD thesis, Georgia Institute of Technology, Atlanta GA, 2005
- S. Karra, R. Shaw, S.C. Patwardhan, S. Noronha, Adaptive model predictive control of multivariable time-varying systems. *Ind. Eng. Chem. Res.* **47**(8), 2708–2720 (2008)
- E.C. Kerrigan, *Robust Constraint Satisfaction Invariant Sets and Predictive Control*. PhD thesis, University of Cambridge, Department of Engineering, Nov 2000
- E.C. Kerrigan, J.M. Maciejowski, Robust feasibility in model predictive control: necessary and sufficient conditions, in *Proceedings of the 40th IEEE Conference on Decision and Control, 2001*, vol. 1, pp. 728–733 (IEEE, Piscataway, 2001)
- E.C. Kerrigan, D.Q. Mayne, Optimal control of constrained, piecewise affine systems with bounded disturbances, in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, vol. 2 (2002), pp. 1552–1557
- H.K. Khalil, *Nonlinear Systems* (Macmillan, New York, 2002)
- N. Kim, Improved methods in neural network based adaptive output feedback control, with applications to flight control. PhD thesis, Georgia Institute of Technology, Atlanta GA, 2003
- H.A. Kingravi, G. Chowdhary, P.A. Vela, E.N. Johnson, Reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(7), 1130–1141 (2012)
- P.V. Kokotović, The joy of feedback: nonlinear and adaptive. *IEEE Control Syst. Mag.* **12**(3), 7–17 (1992)
- S.R. Kou, D.L. Elliott, T.J. Tarn, Observability of nonlinear systems. *Inf. Control* **22**(1), 89–99 (1973)
- E. Koyuncu, N. Ure, G. Inalhan, Integration of path/maneuver planning in complex environments for agile maneuvering ucavs. *J. Intell. Robot. Syst.* **57**, 143–170 (2010). doi:10.1007/s10846-009-9367-1
- M. Krstić, I. Kanellakopoulos, P. Kokotović, *Nonlinear and Adaptive Control Design* (Wiley, New York, 1995)
- A. Kutay, G.V. Chowdhary, A. Calise, E.N. Johnson, A comparison of two novel direct adaptive control methods under actuator failure accommodation, in *Proceedings of the AIAA GNC Conference, Honolulu, HI* (AIAA, 2008)
- E. Lavretsky, Combined/composite model reference adaptive control. *IEEE Trans. Autom. Control* **54**(11), 2692–2697 (2009)
- E. Lavretsky, Reference dynamics modification in adaptive controllers for improved transient performance, in *Guidance Navigation and Control Conference*, Portland, OR, August 2011 (AIAA, 2011). AIAA-2011-620
- E. Lavretsky, K. Wise, Flight control of manned/unmanned military aircraft, in *Proceedings of American Control Conference* (AIAA/IEEE, Evanston, 2005)
- J.H. Lee, Modeling and identification for nonlinear model predictive control: requirements, current status and future research needs, in *Nonlinear Model Predictive Control*. Volume 26 of Progress in Systems and Control Theory (Birkhäuser, Basel-Boston-Berlin, 2000), pp. 269–294
- Y.I. Lee, B. Kouvaritakis, Constrained receding horizon predictive control for systems with disturbances. *Int. J. control* **72**(11), 1027–1032 (1999)
- Y.I. Lee, B. Kouvaritakis, A linear programming approach to constrained robust predictive control. *IEEE Trans. Autom. Control* **45**(9), 1765–1770 (2000)
- C.H. Lee, M.H. Shin, M.J. Chung, A design of gain-scheduled control for a linear parameter varying system: an application to flight control. *Control Eng. Pract.* **9**(1), 11–21 (2001)
- K.H. Lee, J.H. Lee, W.H. Kwon, A stabilizing low-order output feedback receding horizon control for linear discrete time-invariant systems, in *Proceedings of the 2002 American Control Conference, 2002*, vol. 3 (IEEE, Piscataway, 2002), pp. 2412–2417
- A. Levant, A. Pridor, R. Gitizadeh, I. Yaesh, J.Z. Ben-Asher, Aircraft pitch control via second order sliding techniques. *J. Guid. Control Dyn.* **23**, 586–594 (2000)

- F.L. Lewis, Nonlinear network structures for feedback control. *Asian J. Control* **1**, 205–228, 1999. Special Issue on Neural Networks for Feedback Control
- X. Li, H.H.T. Liu, A passive fault tolerant flight control for maximum allowable vertical tail damaged aircraft. *J. Dyn. Syst. Meas Control* **134**(3), 031006. 10.1115/1.4005512
- D. Liberzon, *Switching in Systems and Control* (Birkhäuser, Boston, 2003)
- D. Liberzon, Switched systems, in *Handbook of Networked and Embedded Control Systems*, (Birkhauser, Boston, 2005), pp. 559–574
- T.J.J. Lombaerts, H.O. Huisman, Q.P. Chu, J.A. Mulder, D.A. Joosten, Nonlinear reconfiguring flight control based on online physical model identification, in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 18–21 Aug 2008
- J. Lunze, T. Steffen, Control reconfiguration after actuator failures using disturbance decoupling methods. *IEEE Trans. Autom. Control* **51**(10), 1590–1601 (2006)
- J.M. Maciejowski, *Predictive Control with Constraints* (Prentice Hall Publications, New York, 2002)
- J.-F. Magni, S. Bennani, J. Terlouw, *Robust Flight Control: A Design Challenge*. Lecture Notes in Control and Information Sciences 224 (Springer, London/New York, 1997)
- V. Manikonda, P.O. Arambel, M. Gopinathan, R.K. Mehra, F.Y. Hadaegh, A model predictive control-based approach for spacecraft formation keeping and attitude control in *Proceedings of the 1999 American Control Conference*, 1999, vol. 6 (IEEE, Piscataway, 1999), pp. 4258–4262
- D.L. Marruedo, T. Alamo, E.F. Camacho, Stability analysis of systems with bounded additive uncertainties based on invariant sets: stability and feasibility of mpc, in *Proceedings of the 2002. American Control Conference*, 2002, vol. 1 (Piscataway, IEEE, 2002), pp. 364–369
- I. Masubuchi, J. Kato, M. Saeki, A. Ohara, Gain-scheduled controller design based on descriptor representation of lpv systems: application to flight vehicle control, in *43rd IEEE Conference on Decision and Control*, 2004. CDC, vol. 1 (IEEE, Piscataway, 2004), pp. 815–820
- D.Q. Mayne, Nonlinear model predictive control: challenges and opportunities, in *Nonlinear Model Predictive Control*. Volume 26 of Progress in Systems and Control Theory (Birkhäuser, Basel-Boston-Berlin, 2000), pp. 23–44
- D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: Stability and optimality. *Automatica* **36**, 789–814 (2000)
- B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft* (Kluwer Academic, Norwell, 2003)
- H. Michalska, D.Q. Mayne, Moving horizon observers and observer-based control. *IEEE Trans. Autom. Control* **40**(6), 995–1006 (1995)
- E.A. Morelli, Real time parameter estimation in the frequency domain. *J. Guid. Control Dyn.* **23**(5), 812–818 (2000)
- F. Nardi, Neural network based adaptive algorithms for nonlinear control. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, Nov 2000
- K. Narendra, A. Annaswamy, A new adaptive law for robust adaptation without persistent excitation. *IEEE Trans. Autom. Control* **32**(2), 134–145 (1987)
- K.S. Narendra, A.M. Annaswamy, *Stable Adaptive Systems* (Prentice-Hall, Englewood Cliffs, 1989)
- N. Nguyen, Asymptotic linearity of optimal control modification adaptive law with analytical stability margins, in *Infotech@AIAA Conference*, Atlanta, GA, 2010
- N. Nguyen, K. Krishnakumar, J. Kaneshige, Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft, in *AIAA Guidance, Navigation, and Control Conference and Exhibits*, 21–24 Aug 2006 (AIAA, 2006a). AIAA-2006-6049
- N. Nguyen, K. Krishnakumar, J. Kaneshige, P. Nespeca, Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft, in *AIAA Guidance Navigation and Control Conference*, Keystone, CO, 2006b
- N. Nguyen, K. Krishnakumar, J. Kaneshige, Flight dynamics and hybrid adaptive control of damaged aircraft. *AIAA J. Guid. Control Dyn.* **31**(3), 751–764 (2008)

- G.D. Nicolao, L. Magi, R. Scattolini, Stability and robustness of nonlinear receding horizon control, in *Nonlinear Model Predictive Control*. Volume 26 of Progress in Systems and Control Theory (Birkhäuser, Basel-Boston-Berlin, 2000), pp. 3–22
- H. Nijmeijer, A.J. van der Schaft, *Nonlinear Dynamical Control Systems* (Springer, New York, 1990)
- Office of the Secretary of Defense, Unmanned aircraft systems roadmap 2005–2030. Technical report, Department of Defense, Aug 2005
- Office of the Secretary of Defense, Unmanned aircraft systems roadmap. Technical report, OSD, 2007
- J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks. *Neural Comput.* **3**, 246–257 (1991)
- V. Patel, C. Cao, N. Hovakimyan, K. Wise, E. Lavretsky, Adaptive controller for tailless unstable aircraft in the presence of unknown actuator failures. *Int. J. Control* **82**(4), 705–720 (2009)
- R.J. Patton, J. Chen, Fault-tolerant control systems: the 1997 situation, in *IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes* (Kingston Upon Hull, United Kingdom, 1997), pp. 1033–1054
- I.R. Petersen, V.A. Ugrinovskii, A.V. Savkin, *Robust Control Design Using H_∞ Methods*. Communications and Control Engineering Series (Springer, London, 2000)
- J.-B. Pomet, L. Praly, Automatic Control, *IEEE Transactions on Adaptive nonlinear regulation: estimation from the Lyapunov equation*, (1992), **37**(6), 729–740, doi=10.1109/9.256328
- I.A. Raptis, K.P. Valavanis, *Linear and Nonlinear Control of Small-Scale Unmanned Helicopters*. Volume 45 of International Series on Intelligent Systems, Control, and Automation: Science and Engineering (Springer, Heidelberg, 2011)
- J.B. Rawlings, Tutorial overview of model predictive control. *IEEE Control Syst. Mag.* **20**(3), 38–52 (2000)
- A.G. Richards, Robust constrained model predictive control. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, February 2005
- A. Richards, J.P. How, Decentralized model predictive control of cooperating UAVs, in *IEEE Conference on Decision and Control (CDC)*, Paradise Island, Bahamas, Dec 2004, pp. 4286–4291
- A. Richards, J.P. How, Robust decentralized model predictive control. *Int. J. Control* **80**(9), 1517–1531 (2007)
- L. Rodrigues, J.P. How, Observer-based control of piecewise-affine systems. *Int. J. Control* **76**(5), 459–477 (2003)
- R.M. Sanner, J.-J.E. Slotine, Gaussian networks for direct adaptive control. *IEEE Trans. Neural Netw.* **3**(6), 837–863 (1992)
- M.A. Santillo, D.S. Bernstein, Adaptive control based on retrospective cost optimization. *J. Guid. Control Dyn.* **33**(2), 289–304 (2010). doi: 10.2514/1.46741
- G.H. Shah, Aerodynamic effects and modeling of damage to transport aircraft. Technical report, NASA Langley Research Center, Hampton, VA, 2008
- J.S. Shamma, M. Athans, Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica* **27**(3), 559–564 (1991)
- I.A. Shkolnikov, Y.B. Shtessel, Aircraft nonminimal phase control in dynamic sliding manifolds. *J. Guid. Control Dyn.* **24**, 566–573 (2001)
- S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control – Analysis and Design* (Wiley, Reading, 2005)
- J.E. Slotine, W. Li, *Applied Nonlinear Control* (Prentice-Hall, Englewood Cliffs, 1991)
- M. Steinberg, Historical overview of research in reconfigurable flight control. *Proc. Inst. Mech. Eng. G* **219**(4), 263–275 (2005)
- N. Sundararajan, P. Saratchandran, L. Yan, *Fully Tuned Radial Basis Function Neural Networks for Flight Control* (Kluwer academic publishers, Norwell, 2002)

- G.J. Sutton, R.R. Bitmead, Performance and Computational Implementation of Nonlinear Model Predictive Control on a Submarine, in *Nonlinear Model Predictive Control*. Volume 26 of Progress in Systems and Control Theory (Birkhäuser, Basel-Boston-Berlin, 2000), pp. 461–472
- G. Tao, *Adaptive Control Design and Analysis* (Wiley, Hoboken, 2003)
- U.S. Army UAS Center of Excellence, “Eyes of the Army” U.S. Army Unmanned Aircraft Systems Roadmap 2010–2035, 2010
- K.Y. Volyanskyy, W.M. Haddad, A.J. Calise, A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: beyond σ and e -modifications. *IEEE Trans. Neural Netw.* **20**(11), 1707–1723 (2009)
- B.R. Woodley, J.P. How, R.L. Kosut, Model free subspace based hscr; infin; control, in *Proceedings of the 2001 American Control Conference, 2001*, vol. 4 (2001), pp. 2712–2717
- L. Yu, *Robust Control: Linear Matrix Inequality Approach*. Tshinghua University Press, in Chinese edition, 2002
- T. Yucelen, A. Calise, A derivative-free model reference adaptive controller for the generic transport model, in *AIAA Guidance, Control and Navigation Conference*, Toronto, Canada, Aug 2010. Invited
- Y. Zhang, J. Jang, Bibliographical review on reconfigurable fault-tolerant control systems. *Elsevier Annu. Rev. Control* **32**(1), 229–252 (2008)
- Zhi-sheng Zhang, Huai-min Chen, Cheng-fu Wu, Ma Song-hui, An effective terminal condition ensuring stability of mpc, in *2010 International Conference on Computer Application and System Modeling (ICCA SM)*, vol. 7 (IEEE, Piscataway, 2010), pp. V7–202–V7–206
- A. Zheng, Some practical issues and possible solutions for nonlinear model predictive control, in *Nonlinear Model Predictive Control*. Volume 26 of Progress in Systems and Control Theory (Birkhäuser, Basel-Boston-Berlin, 2000), pp. 129–144
- A. Zheng, F. Allgower, Towards a practical nonlinear predictive control algorithm with guaranteed stability for large-scale systems, in *Proceedings of the 1998 American Control Conference, 1998*, vol. 4 (1998), pp. 2534–2538
- K. Zhou, J.C. Doyle, K. Glover, *Robust and Optimal Control* (Prentice-Hall, Englewood Cliffs, 1996)

Suresh K. Kannan, Girish Vinayak Chowdhary, and Eric N. Johnson

Contents

| | | |
|--------|---|-----|
| 29.1 | Introduction | 614 |
| 29.1.1 | Approach | 616 |
| 29.2 | Control of an Air Vehicle | 619 |
| 29.2.1 | Vehicle Dynamics | 619 |
| 29.2.2 | Control Design | 622 |
| 29.2.3 | Reference Model and Hedging | 625 |
| 29.2.4 | Tracking Error Dynamics | 627 |
| 29.2.5 | Boundedness | 629 |
| 29.3 | Concurrent Learning | 630 |
| 29.4 | Helicopter Specific Design | 634 |
| 29.4.1 | Approximate Model | 634 |
| 29.4.2 | Reference Model | 636 |
| 29.4.3 | Choice of Gain Linear Dynamics | 637 |
| 29.4.4 | Imposing Response Characteristics | 639 |
| 29.5 | Experimental Results | 639 |
| 29.5.1 | Parameter Selections | 640 |
| 29.5.2 | Flight Test | 641 |
| 29.5.3 | Application to a Ducted Fan | 655 |

S.K. Kannan (✉)

Controls Group, Systems Department, United Technologies Research Center, East Hartford, CT, USA

e-mail: kannan@gatech.edu

G.V. Chowdhary

Department of Aeronautics and Astronautics Aerospace Controls Laboratory, Massachusetts Institute of Technology Laboratory for Information and Decision Systems, Cambridge, MA, USA

e-mail: girishc@MIT.EDU

E.N. Johnson

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

e-mail: eric.johnson@ae.gatech.edu

29.5.4 Application to a Fixed-Wing Aircraft 656

29.5.5 Implementation of Concurrent-Learning Adaptive Controller on a
VTOL UAV 657

29.6 Conclusion 666

A Adaptive Element 670

References 671

Abstract

Typically, unmanned aerial vehicles are underactuated systems, that is, they have fewer independent control inputs than degrees of freedom. In a helicopter, for example, the body axes roll, pitch, yaw, and altitude are fully actuated. However, lateral and longitudinal translational motion is only possible by tilting the thrust vector. This chapter develops a six degree-of-freedom flight control algorithm that can track both position and attitude trajectories. Approximate inverse models for vehicle attitude and position dynamics are used for feedback linearization leading to an inner-loop that tracks attitude and angular rate and an outer-loop that tracks position and velocity commands. A single adaptive element is used to compensate for inversion errors (uncertainty) in both loops. A key challenge in realizing an adaptive control design on real aircraft is dealing with actuator magnitude and rate saturation. Such saturation elements cannot be easily captured in inverse models and leads to incorrect learning in the adaptive element during periods of saturation. A mechanism to exactly remove such incorrect learning is provided. Additionally, nonlinear reference models are introduced to mitigate the risks of the closed-loop system entering regions of the flight envelope that result in loss-of-controllability. The resulting adaptive controller accepts trajectory commands comprising of desired position, velocity, attitude, and angular velocity and produces normalized actuator signals required for flight control. A modification to the baseline adaptive control system is also provided that enables long-term retention of the uncertainty approximation within the adaptive element. This architecture is validated through flight tests on several fixed wing and rotorcraft UAVs, including a 145-lb helicopter UAV (Yamaha RMAX or GTMax), a scale model fixed-wing aircraft (GTEdge), and a small ducted fan (GTSpy).

29.1 Introduction

The U.S. Department of Defense Integrated Unmanned Systems Roadmap (2011) defines four levels of autonomy for unmanned systems. Level 1 involves manual operator control. Level 2 assumes automatic control while using humans to delegate waypoints. Level 3 assumes the UAV is capable of performing high-level activities using sensed data when given some directions by a human. Level 4 assumes that the UAV is capable of taking a top-level goal, breaking it into tasks and executing them along with the contingency replanning necessary to accomplish a top-level goal. Level 4 approaches the abstract and high-level goals that are provided to human soldiers in today’s battlefield.

Level 2 behavior is available in almost all UAVs today. However, this accomplishment is after at least two decades of development with multiple efforts for each type of vehicle. The current U.S. Department of Defense focus is on research, development, and procurement of technologies that encompass Level 3 and Level 4 autonomous operations while assuming Level 2 is available. The key to developing fully autonomous Level 4-type behaviors is the ability to plan and also deal with contingencies. At the flight control system level this translates to the closed-loop system being robust and adaptive to changes in the environment and vehicle dynamics. To motivate further, future unmanned systems will involve verification and validation using formal approaches where an amount of trust in autonomy is generated. Current conventional FAA certification practices will be superseded with certification methods that allow integration of UAV's into the civil airspace. One straightforward metric that may be used for *trust in flight control* is the variance in trajectory tracking with uncertainty in the dynamics and environment. Consequently, the vehicles may be required to perform at their limits in order to maintain the required performance. Most current control systems still do not leverage the full flight envelope of small helicopters, at least, unless significant and expensive system identification and validation has been conducted.

Currently, fast algorithms to plan routes around obstacles are available (Frazzoli et al. 2002; Karaman and Frazzoli 2011). To be truly useful, these routes would include high-speed dashes, tight turns around buildings, avoiding dynamic obstacles and other required aggressive maneuvers. Allowing control saturation and adaptation allows higher-level planning algorithms to provide optimistic trajectories which are then tracked by the flight controller to the best of the vehicles ability. The following is a description of some key elements that affect flight control stability and performance.

Parametric uncertainty arises from uncertainty in the mass, inertia properties, and aerodynamic characteristics. This limits the safe operational envelope of the vehicle to flight regimes where control designs are valid and parametric uncertainty is small. The effects of parametric uncertainty and unmodeled dynamics can be handled using a combination of system identification (Gavrilets et al. 2001; Civita et al. 2002a; Mettler 2002) and robust control techniques (Civita et al. 2002b, 2003; Gavrilets et al. 2002). However, system identification is expensive and when changes happen in real-time, for e.g., an unexpected payload is attached or deployed, the available previously-identified models may not fit the current aircraft configuration. In this chapter, parametric uncertainty arises due to approximate inversion and appears as an unknown nonlinear function of the states and controls (unstructured). An adaptive element (neural network) is then used as a nonlinear function approximator to instantaneously cancel the inversion error.

Unmodeled dynamics arise when the vehicle model used for control design neglects parts of the real system's dynamics. Examples include the Bell-Hillier stabilizer bar and the flapping dynamics of the helicopter rotor blades. For most autonomous flight control work, the flapping dynamics may be safely neglected, whereas the Bell-Hillier stabilizer bar found on small rotorcraft cannot be ignored when high-bandwidth attitude control is desired. This chapter assumes state

feedback of rigid-body states keeping the control design simpler by leveraging the control design's robustness to unmodeled dynamics. Related adaptive designs that use an output-feedback controller formulation to explicitly deal with unmodeled dynamics are described in Calise et al. (2001) with experimental results in Corban et al. (2003).

Actuator magnitude and rate saturation limit control authority and thus the closed-loop stability and performance. Addressing input dynamics constitutes an important class of control design methods for systems with bounded actuation and includes Sontag's universal formula approach using control Lyapunov functions (Lin and Sontag 1991) and others (Bemporad et al. 1997; Pappas 1996). Avoiding saturation, however, usually results in either conservative or highly complex control laws leading to possibly very conservative domains of attraction and slow convergence. See Bernstein and Michel (1995) and the references therein for a survey of early work on constrained control.

Other related works on rotorcraft control may be found in Kendoul et al. (2006, 2007, 2010) and a recent survey on guidance, navigation and control methods for rotorcraft is available in Kendoul (2012).

29.1.1 Approach

Helicopters have six degrees of freedom when considering just the rigid body modes and 4 independent controls are available to control them. Traditionally, the control variables lateral stick, δ_{lat} ; longitudinal stick, δ_{lon} ; and pedal, δ_{ped} , control moments around the roll, pitch, and yaw axes, respectively. Finally, the collective input, δ_{coll} , produces thrust along the main rotor shaft. The rotational dynamics are fully actuated, whereas the translational dynamics are underactuated, but controllable. The rotor thrust has to be oriented using the aircraft's pitch and roll attitude to produce translational accelerations.

An overall architecture of the approach is shown in Fig. 29.1 with details in Fig. 29.6. The outer loop is responsible for tracking desired translational accelerations. It generates δ_{coll} to vary rotor thrust along the main shaft and also generates the desired roll and pitch angles to orient the thrust vector to generate linear accelerations in these two underactuated degrees of freedom. Note here that the desired pitch and roll angles are commands to the inner-loop controller. In this respect the inner loop acts like a (virtual) actuator as far as the outer loop is concerned. Similarly, the inner loop generates the actuator deflections necessary to control the rotational dynamics. Of course, here the inner-loop's output actuation signal is subject to the real actuator dynamics of the physical aircraft. In both loops, approximate models of the rotational (inner loop) and translational (outer loop) dynamics are dynamically inverted to produce the actuator deflections (and desired pitch and roll) necessary to achieve the desired angular and linear accelerations. These desired accelerations are generated using reference models dictating the desired ideal closed-loop response. The cascaded inner-outer loop architecture used here is commonly employed in aerospace control applications due to the different

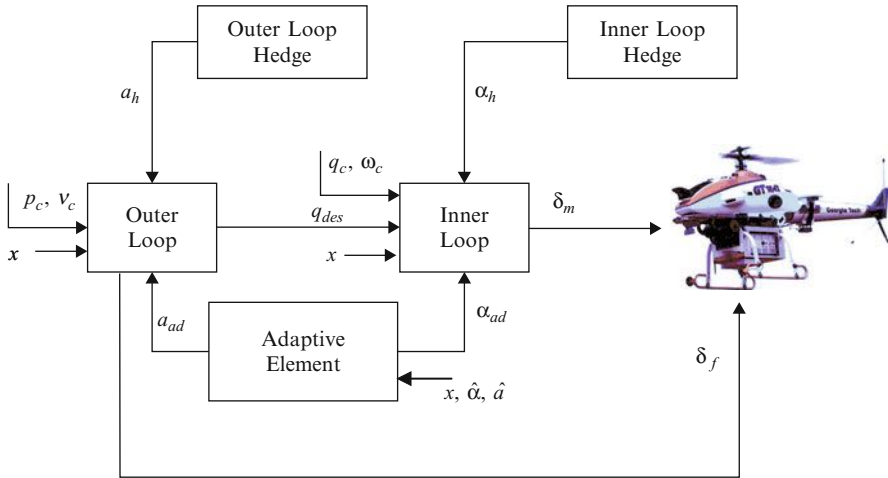


Fig. 29.1 Overall architecture

dynamical time-scales of the two loops. Chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book discusses relevant details of cascaded control systems for UAVs.

Adaptation is introduced in all six degrees of freedom to account for inversion errors arising from the approximate models used for inversion purposes. There is no particular restriction on the inversion that results in *desired* actuator deflections to be bounded. Hence, at large desired accelerations, large actuator deflections may be commanded. Such saturation and dynamics will now appear in the adaptation training signal. This is also true in the case of the outer loop because the commanded pitch and roll attitudes are now subject to the closed-loop dynamics of the inner loop in addition to the actuator dynamics of the δ_{coll} actuator.

These nonlinearities appear in the stability analysis by way of their appearance in the error dynamics. The pseudocontrol hedging signal (PCH) is introduced in the outer-loop and inner-loop reference models in a manner that exactly removes elements of actuator saturation from the training signal for the adaptive element. The reference models themselves are nonlinear and prescribe the aggressiveness with which external commands are achieved. Thus, a comprehensive nonlinear, adaptive, trajectory tracking controller capable of adapting to uncertainties in all six degrees of freedom is developed. It must be noted that although the concrete example used throughout this chapter is one of a helicopter, the controller is not specific to a helicopter UAS. The development is generic; the only difference between a helicopter, a fixed-wing aircraft, or other esoteric aircraft is the manner in which the available controls are categorized and the approximate models used for dynamic inversion purposes.

An underlying assumption of this work is that the nonlinear modeling error uncertainty can be approximated by a continuous function over the flight domain of an aircraft. The goal is to capture an approximation of the uncertainty using

universal approximators such as neural networks. This universal approximation property guarantees that given a sufficient number of neurons, there exists an optimal set of (a priori unknown) weights that can approximate the uncertainty to a desired minimum approximation error. Once these weights are found, the learned dynamics can be used for online planning and health-monitoring purposes. The baseline adaptive laws developed in later sections of this chapter are designed to cancel instantaneous model error but do not necessarily guarantee convergence to the ideal weights during normal course of operation (Johnson and Kannan 2005; Kannan and Johnson 2010b; Kannan 2005). To alleviate this restriction, a modification, the *concurrent-learning adaptive control method*, is introduced that greatly improves the convergence of weights to their ideal values in real-world conditions (Chowdhary and Johnson 2011b). The method can in fact guarantee exponential convergence of the neural network weights to a neighborhood of their ideal values for linearly parameterized neural networks (Chowdhary 2010).

The adaptive controller described in this chapter has been extensively validated in flight on several aircrafts regularly since 2002. The range of aircraft types include the Yamaha RMAX (GTMax) helicopter (Fig. 29.2); an 11-in. ducted fan, the GTSpy (Fig. 29.3); a tailless fixed-wing aircraft, the D6; and a high-thrust-to-weight-ratio aircraft, the GTEdge (Fig. 29.4). The GTEdge is a tilt-body fixed-wing aircraft and capable of hovering on its propeller and flying like a regular fixed-wing aircraft. An interesting set of maneuvers performed by the GTEdge is the hover \Rightarrow forward flight \Rightarrow hover, all using the same adaptive control system. The methods discussed here have also been implemented on smaller aircraft such as



Fig. 29.2 The GTMax helicopter



Fig. 29.3 The GTSpy 11-in. ducted fan

the GT Twinstar (Fig. 29.5), a foam-built twin-engine aircraft; the GT Logo, a small rotorcraft of about 1-m rotor diameter; and the GTQ (Chowdhary et al. 2011b), a miniature quadrotor. On the GT Twinstar, a variant of the algorithms presented here was used for flight with 25 % right wing missing (Chowdhary et al. 2010, 2011a, 2012).

29.2 Control of an Air Vehicle

29.2.1 Vehicle Dynamics

Consider an air vehicle modeled as a nonlinear system of the form

$$\dot{p} = v, \quad (29.1)$$

$$\dot{v} = a(p, v, q, \omega, \delta_f, \delta_m), \quad (29.2)$$

$$\dot{q} = \dot{q}(q, \omega), \quad (29.3)$$

$$\dot{\omega} = \alpha(p, v, q, \omega, \delta_f, \delta_m), \quad (29.4)$$



Fig. 29.4 The GTEdge aircraft with a high (greater than 1) thrust-to-weight ratio

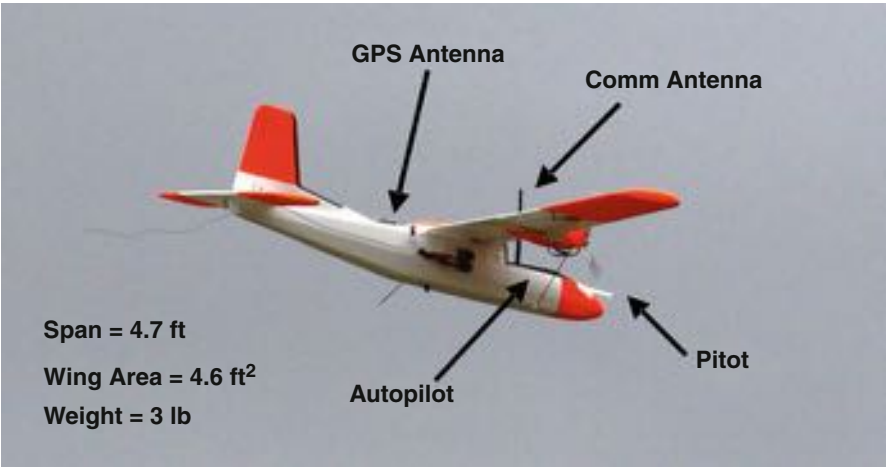


Fig. 29.5 The GT Twinstar foam-built twin-engine aircraft equipped for fault-tolerant control work (see, e.g., Chowdhary et al. [2011a](#))

where $p \in \mathcal{R}^3$ is the position vector, $v \in \mathcal{R}^3$ is the velocity of the vehicle, $q \in \mathcal{R}^4$ is the attitude quaternion, and $\omega \in \mathcal{R}^3$ is the angular velocity. Equation (29.2) represents translational dynamics and Eq. (29.4) represents the attitude dynamics. Together, they represent rigid body dynamics and flat-earth kinematics as given in Etkin (1972) and Stevens and Lewis (2003) and discussed in detail in the chapter ► [Linear Flight Control Techniques for Unmanned Aerial Vehicles](#) in this book. Equation (29.3) represents the quaternion propagation equations (Stevens and Lewis 2003). The use of quaternions, though not a minimal representation of attitude, avoids numerical and singularity problems that Euler-angle-based representations have. This enables the control system to be all attitude capable as required for aggressive maneuvering. The state vector x may now be defined as $x \triangleq [p^T \ v^T \ q^T \ \omega^T]^T$.

Remark 29.1. The objective is to design a control system that can track a given position, velocity, attitude, and angular rate trajectory. The consolidated trajectory command is given by $[p_c^T \ v_c^T \ q_c^T \ \omega_c^T]^T$.

The control vectors are denoted by δ_f and δ_m and represent actual physical actuators on the aircraft, where δ_f denotes the primary force-generating actuators and δ_m denotes the primary moment generating actuators. For a helicopter, the main force effector is the rotor thrust which is controlled by changing main rotor collective δ_{coll} . Hence, $\delta_f \in \mathcal{R} = \delta_{coll}$. There are three primary moment control surfaces, the lateral cyclic δ_{lat} , longitudinal cyclic δ_{lon} , and tail rotor pitch, also called the pedal input δ_{ped} . Hence, $\delta_m \in \mathcal{R}^3 = [\delta_{lat} \ \delta_{lon} \ \delta_{ped}]^T$. In this chapter, the primary moment producing controls are treated as the inner-loop control effector, whereas the $\delta_f = \delta_{coll}$ is treated as an outer-loop control effector. In general, both control inputs, δ_f and δ_m , may each produce both forces and moments. The helicopter is an underactuated system, and hence, the aircraft attitude, q , is treated like a *virtual actuator* used to tilt the main rotor thrust in order to produce desired translational accelerations in the longitudinal and lateral directions. Thus, it is not possible to track the commanded pitch and roll for a helicopter independently. It is only possible to track the heading component of the attitude q_c and body-yaw rate ω_3 independently. Direct control over the translational accelerations in the *body* – *z* – *axis* is possible using δ_{coll} .

The consolidated control vector δ is defined as

$$\delta \triangleq [\delta_f^T \ \delta_m^T]^T;$$

the actuators themselves may have dynamics represented by

$$\dot{\delta} = \begin{bmatrix} \dot{\delta}_m \\ \dot{\delta}_f \end{bmatrix} = \begin{bmatrix} g_m(x, \delta_m, \delta_{m_{des}}) \\ g_f(x, \delta_f, \delta_{f_{des}}) \end{bmatrix} = g(x, \delta, \delta_{des}), \quad (29.5)$$

where $g(\cdot)$ is generally unknown.

Remark 29.2. It is possible to extend the architecture in order to treat actuator dynamics as simply another system in cascade with the translational and attitude dynamics, and the control design would include an outer, inner, and actuator loop with the actuator loop being the lowest block in the cascade. However, unless the physical actuators need to be stabilized, their internal dynamics may be assumed to be asymptotically stable. In this chapter, rate and higher-order dynamics are ignored, but magnitude saturation will be handled explicitly. It can be shown that such an assumption is possible because the control design is robust to the unmodeled dynamics (Kannan 2005).

29.2.2 Control Design

The control architecture is based on a model reference adaptive control architecture (see Fig. 29.6). Noting that Eqs. (29.1) and (29.3) represent exactly known kinematics, approximate models for translational acceleration, \hat{a} , and a model for angular acceleration, $\hat{\alpha}$, need to be established.

$$\begin{bmatrix} a_{des} \\ \alpha_{des} \end{bmatrix} = \begin{bmatrix} \hat{a}(p, v, q_{des}, \omega, \delta_{f_{des}}, \hat{\delta}_m) \\ \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \delta_{m_{des}}) \end{bmatrix}.$$

Here, a_{des} and α_{des} are commonly referred to as the pseudocontrol and represent desired accelerations. Additionally, $\delta_{f_{des}}, \delta_{m_{des}}, q_{des}$ are the control inputs and attitude expected to achieve the desired pseudocontrol. This form assumes that translational dynamics are coupled strongly with attitude dynamics, as is the case for a helicopter. From the outer-loop's point of view, q (attitude) is like a *virtual actuator* that generates translational accelerations and q_{des} is the desired attitude that the outer-loop inversion expects will contribute toward achieving the desired translational acceleration, a_{des} . The dynamics of q appear like actuator dynamics to the outer loop.

Remark 29.3. Although the models are approximate, their functional dependence on vehicle rigid body and actuator states are stated accurately for completeness. It is likely that a specific approximate model that is introduced might drop some of this dependency.

Remark 29.4. The attitude quaternion q_{des} will be used to augment the externally commanded attitude q_c to achieve the desired translational accelerations. Ideally, the trajectory generator would generate a commanded attitude q_c that is consistent with the translational acceleration profile needed to track $x_c(t)$ and $v_c(t)$. If not, the outer-loop inverse takes care of correcting it by an amount necessary to achieve the desired translational accelerations in the longitudinal and lateral directions.

The models have a functional dependence on current actuator position. Because actuator positions are often not measured on small unmanned aerial

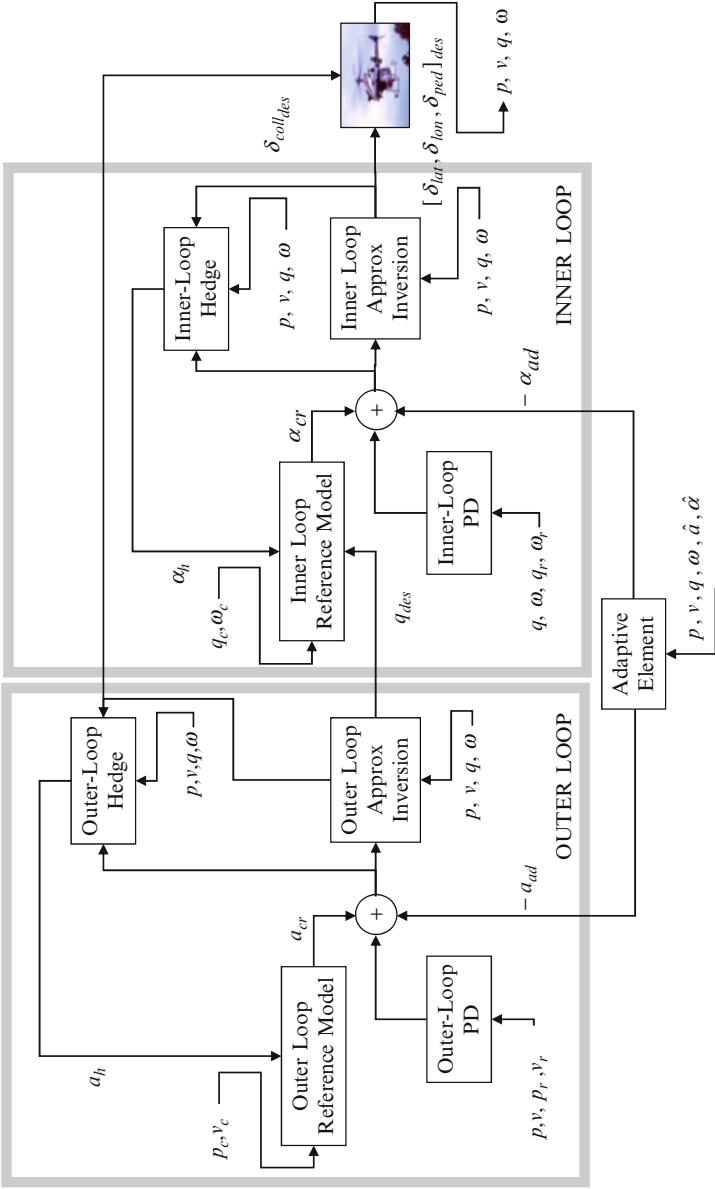


Fig. 29.6 Detailed inner- and outer-loop controller architecture for an autonomous helicopter

vehicles, estimates of the actuator positions $\hat{\delta}_m, \hat{\delta}_f$ can be used. When the actuator positions are directly measured, they may be regarded as known $\hat{\delta}_m = \delta_m$ and $\hat{\delta}_f = \delta_f$. In fact, in the outer-loop's case, good estimates of the roll and pitch attitude virtual actuators are available using inertial sensors and a navigation filter. The approximate models may now be inverted to obtain the desired control and attitude

$$\begin{bmatrix} \delta_{f_{des}} \\ q_{des} \end{bmatrix} = \begin{bmatrix} \hat{a}_{\delta_f}^{-1}(p, v, a_{des_{\delta_f}}, \omega, \hat{\delta}_m) \\ \hat{a}_q^{-1}(p, v, a_{des_q}, \omega, \hat{\delta}_m) \end{bmatrix} \quad (29.6)$$

$$\delta_{m_{des}} = \hat{\alpha}^{-1}(p, v, q, \omega, \hat{\delta}_f, \alpha_{des}),$$

with $a_{des_{\delta_f}} + a_{des_q} = a_{des}$, $\hat{a}_{\delta_f}, \hat{a}_q$ formulated to be consistent with Eq. (29.6) and where actuator estimates are given by actuator models

$$\dot{\hat{\delta}} = \begin{bmatrix} \dot{\hat{\delta}}_f \\ \dot{\hat{\delta}}_m \end{bmatrix} = \begin{bmatrix} \hat{g}_f(x, \hat{\delta}_f, \delta_{f_{des}}) \\ \hat{g}_m(x, \hat{\delta}_m, \delta_{m_{des}}) \end{bmatrix} = \hat{g}(x, \hat{\delta}, \delta_{des}). \quad (29.7)$$

Introducing the inverse control law Eq. (29.6) into Eqs. (29.2) and (29.4) results in the following closed-loop translational and attitude dynamics:

$$\begin{aligned} \dot{v} &= a_{des} + \bar{\Delta}_a(x, \delta, \hat{\delta}) - a_h \\ \dot{\omega} &= \alpha_{des} + \bar{\Delta}_\alpha(x, \delta, \hat{\delta}) - \alpha_h, \end{aligned} \quad (29.8)$$

where

$$\bar{\Delta}(x, \delta, \hat{\delta}) = \begin{bmatrix} \bar{\Delta}_a(x, \delta, \hat{\delta}) \\ \bar{\Delta}_\alpha(x, \delta, \hat{\delta}) \end{bmatrix} = \begin{bmatrix} a(x, \delta) - \hat{a}(x, \hat{\delta}) \\ \alpha(x, \delta) - \hat{\alpha}(x, \hat{\delta}) \end{bmatrix}, \quad (29.9)$$

are static nonlinear functions (model error) that arise due to imperfect model inversion and errors in the actuator model \hat{g} . The main discrepancy between $g(\cdot)$ and $\hat{g}(\cdot)$ is the lack of a magnitude-saturation function in \hat{g} . This is required in order to maintain invertibility. The signals, a_h and α_h , represent the pseudocontrol that *cannot be achieved* due to actuator input characteristics such as saturation. If the model inversion were perfect and no magnitude saturation were to occur, $\bar{\Delta}$, a_h and α_h would vanish leaving only the pseudocontrols a_{des} and α_{des} .

Two tasks now remain, (1) stabilize the feedback linearized dynamics and (2) address the effects of model error. The desired accelerations may be designed as

$$\begin{aligned} a_{des} &= a_{cr} + a_{pd} - \bar{a}_{ad} \\ \alpha_{des} &= \alpha_{cr} + \alpha_{pd} - \bar{\alpha}_{ad}, \end{aligned} \quad (29.10)$$

where a_{cr} and α_{cr} are outputs of reference models for the translational and attitude dynamics, respectively. a_{pd} and α_{pd} are outputs of proportional-derivative (PD) compensators; and finally, \tilde{a}_{ad} and $\tilde{\alpha}_{ad}$ are the outputs of an adaptive element designed to cancel model error $\tilde{\Delta}$. The effects of input dynamics, represented by a_h, α_h , will first be addressed in the following section by designing the reference model dynamics such that they do not appear in the tracking error dynamics. The reference model, tracking error dynamics, and boundedness are discussed in the following sections with details of the adaptive element left to Appendix Sect. A.

29.2.3 Reference Model and Hedging

Any dynamics and nonlinearities associated with the actuators δ_m, δ_f have not yet been considered in the design. If they become saturated (position or rate), the reference models will continue to demand tracking as though full authority were still available. Furthermore, the inner loop appears like an actuator with dynamics to the outer loop. Practical operational limits on the maximum attitude of the aircraft may have also been imposed in the inner-loop reference model. This implies that the outer-loop desired attitude augmentation q_{des} may not actually be achievable or at the very least is subject to the inner-loop dynamics.

If the reference model is designed as

$$\begin{aligned}\dot{v}_r &= a_{cr}(p_r, v_r, p_c, v_c) \\ \dot{\omega}_r &= \alpha_{cr}(q_r, \omega_r, q_c \oplus q_{des}, \omega_c),\end{aligned}\tag{29.11}$$

where p_r and v_r are the outer-loop reference model states and q_r, ω_r are the inner-loop reference model states. The external command signal is $x_c = [p_c^T \ v_c^T \ q_c^T \ \omega_c^T]^T$. The attitude rotation desired by the outer loop is now added to the commands for the inner-loop controller. Here, $q_c \oplus q_{des}$ denotes quaternion multiplication (Stevens and Lewis 2003) and effectively concatenates the two rotations.

If tracking error dynamics is computed by subtracting Eq.(29.10) from Eq.(29.11), the un-achievable acceleration a_h, α_h will appear in the tracking error dynamics. When an adaptive element such as a neural network or integrator is introduced, these effects of input dynamics propagate into the training signal and eventually result in the adaptive element attempting to correct for them, leading to incorrect adaptation.

Tackling this issue involves redesigning the reference model by subtracting the deficit accelerations (pseudocontrol hedging):

$$\dot{v}_r = a_{cr}(p_r, v_r, p_c, v_c) - a_h,\tag{29.12}$$

$$\dot{\omega}_r = \alpha_{cr}(q_r, \omega_r, q_c \oplus q_{des}, \omega_c) - \alpha_h, l\tag{29.13}$$

where a_h and α_h are the differences between commanded pseudocontrol and an estimate of the achieved pseudocontrol. It is an estimate because actual actuator positions may not be known. Additionally, the aircraft state vectors p, v, q, ω are estimated using a Kalman filter (Christophersen et al. 2006; Chowdhary et al. 2011b). However, for purposes of control design, they are assumed to be known, and thus the virtual actuators such as attitude may be assumed to be known in the PCH computation. This assumption may have to be revisited in the case where the control/observer pair is not assumed to be separable, perhaps in a tough localization problem where the control inputs directly affect the observability of the aircraft states.

The PCH signals are given by

$$\begin{aligned} a_h &= \hat{a}(p, v, q_{des}, \omega, \delta_{f_{des}}, \hat{\delta}_m) - \hat{a}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m) \\ &= a_{des} - \hat{a}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m), \end{aligned} \quad (29.14)$$

$$\begin{aligned} \alpha_h &= \hat{\alpha}(p, v, q, \omega, \delta_{f_{des}}, \delta_{m_{des}}) - \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m) \\ &= \alpha_{des} - \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m). \end{aligned} \quad (29.15)$$

The hedge signals a_h, α_h , do not directly affect the reference model output a_{cr}, α_{cr} , but do so only through subsequent changes in the reference model states.

The command tracking error may now be defined as e_r

$$e_r \triangleq \begin{bmatrix} p_c - p_r \\ v_c - v_r \\ \tilde{Q}(q_c, q_r) \\ \omega_c - \omega_r \end{bmatrix}, \quad (29.16)$$

with corresponding command tracking error dynamics given by

$$\dot{e}_r = \begin{bmatrix} v_c - v_r \\ a_c - (a_{cr} - a_h) \\ \omega_c - \omega_r \\ \alpha_c - (\alpha_{cr} - \alpha_h) \end{bmatrix}. \quad (29.17)$$

The particular form of the reference model dynamics chosen for the translational dynamics, a_{cr} , and attitude dynamics, α_{cr} , has profound effects on the overall response and controllability of the system. This is fully expounded in Chap. 4 of Kannan (2005) and in Kannan and Johnson (2010a). Also see Kannan and Johnson (2003) for a discussion on the effects of reference model poles when various elements saturate.

As a summary, three reference models were considered:

- A *linear reference model* will attempt to elicit a linear response in the plant when no such response is possible (peaking) as the plant is nonlinear, especially with the magnitude saturation of actuators.
- The *nested saturation-based reference model* is an alternative to the linear reference model containing saturations functions appearing in a nested form and is based on the work by Teel (1996, 1997). This form allows one to restrict the evolution of states in a prescribable manner.
- The *constrained linear reference model* is a special case of the nested saturation-based reference model that is locally linear near the origin.

For the quadratic candidate Lyapunov functions chosen in Kannan (2005), only the nested-saturation and constrained linear reference models have their Lyapunov derivative bounds on the PCH signals a_h, α_h . In this chapter, the constrained reference model is used with equations given later in Sect. 29.4.2.

29.2.4 Tracking Error Dynamics

The tracking error vector is defined as, e as

$$e \triangleq \begin{bmatrix} p_r - p \\ v_r - v \\ \tilde{Q}(q_r, q) \\ \omega_r - \omega \end{bmatrix}, \quad (29.18)$$

where $\tilde{Q} : \mathcal{R}^4 \times \mathcal{R}^4 \mapsto \mathcal{R}^3$ is a function (Johnson 2000) that, given two quaternions, results in an error angle vector with three components. An expression for \tilde{Q} is given by

$$\begin{aligned} \tilde{Q}(p, q) &= 2 \operatorname{sgn}(q_1 p_1 + q_2 p_2 + q_3 p_3 + q_4 p_4) \times \\ &\quad \begin{bmatrix} -q_1 p_2 + q_2 p_1 + q_3 p_4 - q_4 p_3 \\ -q_1 p_3 - q_2 p_4 + q_3 p_1 + q_4 p_2 \\ -q_1 p_4 + q_2 p_3 - q_3 p_2 + q_4 p_1 \end{bmatrix}. \end{aligned} \quad (29.19)$$

The output of the PD compensators may be written as

$$\begin{bmatrix} a_{pd} \\ \alpha_{pd} \end{bmatrix} = \begin{bmatrix} R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d \end{bmatrix} e, \quad (29.20)$$

where $R_p, R_d \in \mathcal{R}^{3 \times 3}$, $K_p, K_d \in \mathcal{R}^{3 \times 3}$ are linear gain positive-definite matrices whose choice is discussed below. The tracking error dynamics may be found by directly differentiating Eq. (29.18):

$$\dot{e} = \begin{bmatrix} v_r - v \\ \dot{v}_r - \dot{v} \\ \omega_r - \omega \\ \dot{\omega}_r - \dot{\omega} \end{bmatrix}.$$

Considering \dot{e}_2 ,

$$\begin{aligned} \dot{e}_2 &= \dot{v}_r - \dot{v} \\ &= a_{cr} - a_h - a(x, \delta) \\ &= a_{cr} - a_{des} + \hat{a}(x, \hat{\delta}) - a(x, \delta) \\ &= a_{cr} - a_{pd} - a_{cr} + \bar{a}_{ad} + \hat{a}(x, \hat{\delta}) - a(x, \delta) \\ &= -a_{pd} - (a(x, \delta) - \hat{a}(x, \hat{\delta}) - \bar{a}_{ad}) \\ &= -a_{pd} - (\bar{\Delta}_a(x, \delta, \hat{\delta}) - \bar{a}_{ad}), \end{aligned}$$

\dot{e}_4 may be found similarly. Then, the overall tracking error dynamics may now be expressed as

$$\dot{e} = Ae + B \left[\bar{v}_{ad} - \bar{\Delta}(x, \delta, \hat{\delta}) \right], \quad (29.21)$$

where $\bar{\Delta}$ is given by Eq. (29.9),

$$\bar{v}_{ad} = \begin{bmatrix} \bar{a}_{ad} \\ \bar{\alpha}_{ad} \end{bmatrix}, A = \begin{bmatrix} 0 & I & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -K_p & -K_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix}, \quad (29.22)$$

and so the linear gain matrices must be chosen such that A is Hurwitz's. Now, \bar{v}_{ad} remains to be designed in order to cancel the effect of $\bar{\Delta}$.

Note that commands, δ_{mdes} , δ_{fdes} , q_{des} , do not appear in the tracking error dynamics. PCH allows adaptation to continue when the actual control signal has been replaced by any arbitrary signal and thus allows switching between manual and automatic flight during flight tests without any transients. Furthermore, if the actuator is considered ideal and the actual position and the commanded position are equal, addition of the PCH signals a_h , α_h has no effect on any system signal.

The adaptive signal \bar{v}_{ad} contains two terms

$$\bar{v}_{ad} = v_{ad} + v_r = \begin{bmatrix} a_{ad} + a_r \\ \alpha_{ad} + \alpha_r \end{bmatrix},$$

where v_{ad} is the output of the single hidden layer (SHL) neural network (NN) described in Appendix Sect. A. For an air vehicle with adaptation in all degrees of freedom, $v_{ad} \in \mathcal{R}^6$, where the first three outputs, a_{ad} , approximate Δ_a and the

last three outputs, α_{ad} , approximate Δ_α , is consistent with the definition of the error in Eq. (29.18). The term $v_r = [a_r^T, \alpha_r^T]^T \in \mathcal{R}^6$ is a robustifying signal that arises in the proofs of boundedness found in Kannan (2005).

29.2.5 Boundedness

Noting that the plant states are given by

$$x(t) = x_r(t) - e(t), \quad (29.23)$$

boundedness of the reference model states $x_r(t)$ is sufficient to establish boundedness of the plant states $x(t)$. However, the reference model dynamics now include the PCH signal which could be arbitrary and large. The problem of actuator saturation has effectively been moved from affecting the tracking error dynamics to affecting the command tracking error dynamics of the reference model. If boundedness of 29.17 can be established, then an assumption that the external command $x_c(t)$ is bounded is sufficient to establish boundedness of the overall system. The following assumptions are required to guarantee boundedness:

Assumption 1 *The external command x_c is bounded:*

$$\|x_c\| \leq \bar{x}_c.$$

Assumption 2 *The NN approximation $\Delta(x, \hat{\delta}) = v_{ad}(x, \hat{\delta}) + \epsilon$ holds in a compact domain \mathcal{D} , which is large enough such that $\mathcal{D}_{x_c} \times \mathcal{D}_{e_r} \times \mathcal{D}_e \times \mathcal{D}_{\bar{z}}$ maps into \mathcal{D} . This assumption is required to leverage the universal approximation property of SHL NN (Hornik et al. 1989a).*

Assumption 3 *The norm of the ideal weights (V^*, W^*) is bounded by a known positive value,*

$$0 < \|Z^*\|_F \leq \bar{Z},$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This is justified due to the universal approximation property of SHL NN if the previous assumption holds (Hornik et al. 1989a).

Assumption 4 *Note that Δ depends on v_{ad} through the pseudocontrol v , whereas \bar{v}_{ad} has to be designed to cancel Δ . Hence, the existence and uniqueness of a fixed-point solution for $v_{ad} = \Delta(x, v_{ad})$ is assumed. Sufficient conditions (Calise et al. 2001) for this assumption are also available.*

Assumption 5 *Noting that the null-controllable region of the plant \mathcal{C}_x is not necessarily a connected or closed set, assume that $\mathcal{D} \subseteq \mathcal{C}_x$ and that \mathcal{D} in addition to being compact is also convex.*

The adaptive element training signal, r ; adaptive element output, v_{ad} ; and robustifying term, v_r , are given by

$$\begin{aligned} r &= (e^T P B)^T \\ \bar{v}_{ad} &= v_{ad} + v_r \\ v_{ad} &= W^T \sigma(V^T \bar{x}) \\ v_r &= -K_r (\|Z\|_F + \bar{Z}) r \frac{\|e\|}{\|r\|}. \end{aligned}$$

Theorem 29.1. Consider the system given by (29.1)–(29.4), with the inverse law (29.6), reference models (29.35) and (29.36) which is consistent with (29.12) and (29.13), where the gains are the same as those selected such that the system matrix in (29.21) is Hurwitz and Assumptions 1–5 are met. If $K_r > 0 \in \mathcal{R}^{k \times k}$ is chosen sufficiently large with lower limit stated in the proof, and adaptive element weights W, V satisfy the adaptation laws

$$\begin{aligned} \dot{W} &= -[(\sigma - \sigma' V^T \bar{x}) r^T + \kappa \|e\| W] \Gamma_W \\ \dot{V} &= -\Gamma_V [\bar{x} (r^T W^T \sigma') + \kappa \|e\| V], \end{aligned} \quad (29.24)$$

with $\Gamma_W, \Gamma_V > 0$, $\kappa > 0$ with lower limit stated in the proof, and the external command $x_c(t)$ is such that $e_r(t) \in \Omega(P_r, \rho)$, for some $\rho > 0$, then, the command tracking error, e_r ; the reference model tracking error, e ; and adaptive element weights (\tilde{W}, \tilde{V}) are uniformly ultimately bounded. Further, the plant states, x , are ultimately bounded.

Proof. See proof of Theorem 4 in Kannan (2005). □

Remark 29.5. The update laws $\dot{W}(t), \dot{V}(t)$ closely resembles the backpropagation method of tuning neural network weights (Rumelhart and Williams 1986; Suykens et al. 1996; Haykin 1998; Kim and Lewis 1998). However, it is important to note that the training signal r is different from that of the backpropagation-based learning laws.

29.3 Concurrent Learning

The single hidden layer neural-network-based adaptive elements used in this chapter are known to have the universal approximation property (Haykin 1998; Hornik et al. 1989a), that is, given sufficient number of hidden layer neurons there exists a set of ideal weights W^*, V^* that brings the neural network output to within an ϵ neighborhood of the modeling error $\bar{\Delta}(x, \delta)$ (uncertainty). The adaptive laws in Eq. (29.24) are designed to minimize the instantaneous tracking error e . Although Theorem 29.1 guarantees boundedness of the tracking error e , it cannot

be guaranteed that the adaptive weights will approach the ideal weights over the long term during a normal course of operation. It is useful to drive the weights closer toward their ideal values, as the resulting NN representation forms a good approximation of the uncertainty, which can result in improved performance and can be used for planning and health-monitoring purposes.

One limitation of the adaptive laws in Eq. (29.24) (without the e -modification term) is that at any instant of time, they are constrained to search for the ideal weights only in the direction of instantaneous tracking error reduction. In that sense these adaptive laws are equivalent to a gradient-descent or a greedy update. Therefore, the adaptive weights may not approach the ideal weights unless all directions in which the weights can evolve to reduce the tracking error are explored infinitely often during the course of operation. Intuitively, this explains why *persistence of excitation* is required to guarantee weight convergence for most adaptive laws (see, e.g., Boyd and Sastry 1986). The idea in concurrent learning is to use specifically selected and online recorded data to ensure parameter convergence without requiring persistent excitation. If data is recorded when the system states are exciting and if invariant system properties, such as modeling error information, can be inferred from the recorded data, then weight convergence can be guaranteed without requiring persistent excitation (Chowdhary 2010). In an implementation of a concurrent-learning adaptive controller, each measured data point is evaluated to determine whether it should be added to a *history stack*. The maximum number of recorded data points is limited, and when this number is reached, new data points replace old points. Note that the history stack is not intended to be a buffer of last p states. The approximation modeling error at a recorded data point, which is an invariant system property, is inferred from the recorded data point by noting that $\Delta(x_i, \delta_i) \approx \hat{x}_i - v(x_i, \delta_i)$ where \hat{x}_i is the smoothed estimate of \dot{x}_i (Chowdhary and Johnson 2011b; Gelb 1974). Adaptation happens concurrently on recorded and current data such that the instantaneous tracking error and the modeling error at all recorded data points simultaneously reduces (Chowdhary and Johnson 2010, 2011b; Chowdhary 2010).

It was shown in Chowdhary (2010) and Chowdhary and Johnson (2010) that for linearly parameterized uncertainties, the requirement on persistency of excitation can be relaxed if online recorded data is used concurrently with instantaneous data for adaptation. If the uncertainty can be linearly parameterized, then

$$\bar{\Delta}(x, \delta) = W^{*T} \phi(x, \delta) + \epsilon(x, \delta) \quad (29.25)$$

where $W^* \in \mathbb{R}^l$ denotes the ideal weights that guarantee for a given basis function $\phi(x, \delta) \in \mathbb{R}^l$ $\sup_{\delta} \|\epsilon(x, \delta)\| \leq \bar{\epsilon}$ for some positive constant $\bar{\epsilon}$. In this case, the adaptive element can also be linearly parameterized in the form $v_{ad} = W^T \phi(x, \delta)$. In certain UAV applications, the basis functions for the modeling error are known (see, e.g., the problem of wing-rock control (Singh et al. 1995)), in which case, the existence of an unknown ideal weight vector W^* can be established such that $\bar{\epsilon} = 0$. The representation in (29.25) can also be guaranteed for any continuous modeling error approximated over a compact domain if elements of ϕ consist of set of Gaussian radial basis functions and a scalar bias term b_w (see Park and Sandberg

1991; Haykin 1998). For either of these linearly parameterized representations of the uncertainty, the following theorem can be proven (Chowdhary 2010; Chowdhary and Johnson 2010, 2011a):

Theorem 29.2. *Consider the system given by (29.1)–(29.4), with the inverse law (29.6), reference models (29.35) and (29.36) which is consistent with (29.12) and (29.13), where the gains are the same as those selected such that the system matrix in (29.21) is Hurwitz. Assume further that the uncertainty is linearly parameterizable using an appropriate set of bases over a compact domain D and that Assumptions 4 and 5 hold. For each recorded data point j , let $\epsilon_i(t) = W^T(t)\phi(x_i, \delta_i) - \hat{\Delta}(x_i, \delta_i)$, with $\hat{\Delta}(x_i, \delta_i) = \hat{x}_i - v(x_i, \delta_i)$. Now consider the following update law for the weights of the RBF NN:*

$$\dot{W} = -\Gamma_W \sigma(z) e^T P B - \sum_{j=1}^p \Gamma_W \sigma(x_i, \delta_i) \epsilon_j^T, \quad (29.26)$$

and assume that $Z = [\phi(z_1), \dots, \phi(z_p)]$ and $\text{rank}(Z) = l$. Let B_α be the largest compact ball in D , and assume $\zeta(0) \in B_\alpha$, define $\delta = \max(\beta, \frac{2\|PB\|\bar{\epsilon}}{\lambda_{\min}(Q)} + \frac{p\bar{\epsilon}\sqrt{l}}{\lambda_{\min}(\Omega)})$, and assume that D is sufficiently large such that $m = \alpha - \delta$ is a positive scalar. If the states x_{rm} of the bounded-input bounded-output reference model of (29.11) remains bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m\}$ for all $t \geq 0$, then the tracking error e and the weight error $\tilde{W} = W - W^*$ are uniformly ultimately bounded. Furthermore, if the representation in (29.25) is exact over the entire operating domain, that is, $\bar{\epsilon} = 0$, then the tracking error and weight error converge exponentially fast to a compact ball around the origin for arbitrary initial conditions, with the rate of convergence directly proportional to the minimum singular value of the history stack matrix Z .

Remark 29.6. The size of the compact ball around the origin where the weight and tracking error converge is dependent on the representation error $\bar{\epsilon}$ and the estimation error $\check{\epsilon} = \max_i \|\dot{x}_i - \hat{x}_i\|$. The former can be reduced by choosing appropriate number of RBFs across the operating domain, and the latter can be reduced by an appropriate implementation of a fixed point smoother. Note that $\hat{x}(t)$ is not needed at a current instant t . Therefore, an appropriate implementation of a fixed-point smoother alleviates several issues faced in estimating $\hat{x}(t)$ by using recorded data before and after a data point is recorded to form very accurate estimates of \hat{x}_i (Gelb 1974; Chowdhary and Johnson 2011b).

The history stack matrix $Z = [\phi(z_1), \dots, \phi(z_p)]$ is not a buffer of last p states. It can be updated online by including data points that are of significant interest over the course of operation. In the linearly parameterized case, convergence is guaranteed as soon as the history stack becomes full ranked. New data points could replace existing data points once the history stack reaches a predetermined size. It was shown in Chowdhary and Johnson (2011a) that the rate of convergence of the tracking error and weights is directly proportional to the minimum singular

value of Z . This provides a useful metric to determine which data points are most useful for improving convergence. Consequently, an algorithm for adding points that improve the minimum singular value of Z for the case of linearly parameterizable uncertainty was presented in Chowdhary and Johnson (2011a). The main limitation of the linearly parameterized RBF NN representation of the uncertainty is that the RBF centers need to be preallocated over an estimated compact domain of operation D . Therefore, if the system evolves outside of D , all benefits of using adaptive control are lost. This can be addressed by evolving the RBF basis to reflect the current domain of operation; a reproducing kernel Hilbert space approach for accomplishing this was presented in Kingravi et al. (2012).

On the other hand, the nonlinearly parameterized NN described in Appendix Sect. A is more flexible: it only requires the uncertainties to be bounded over a compact set, but does not require that the domain of operation be known. However, it is typically more difficult to analyze due to the nonlinear parameterizations. In Chowdhary and Johnson (2011b), a concurrent-learning adaptive law was proposed for SHL NN and was validated in flight on the GTMax rotorcraft (see Sect. 29.5.5). In particular, the following theorem can be proven (Chowdhary and Johnson 2011b; Chowdhary 2010):

Theorem 29.3. *Consider the system given by (29.1)–(29.4), with the inverse law (29.6), reference models (29.35) and (29.36) which is consistent with (29.12) and (29.13), where the gains are the same as those selected such that the system matrix in (29.21) is Hurwitz and Assumptions 1–5 are met. Let $i \in \mathfrak{S}$ denote the index of an online recorded data point z_i , define $r_{b_i}(t) = v_{ad}(z_i) - \hat{\Delta}(z_i)$, where $\hat{\Delta}(z) = \hat{\dot{x}}_i - v_i$ and $\hat{\dot{x}}_i$ is the smoothed estimate of \dot{x}_i , and consider the following adaptive law:*

$$\begin{aligned} \dot{W}(t) = & -(\sigma(V^T(t)\bar{x}(t)) - \sigma'(V^T(t)\bar{x}(t))V^T(t)\bar{x}(t))r^T(t)\Gamma_w - k\|e(t)\|W(t) \\ & - W_c(t) \sum_{i=1}^p (\sigma(V^T(t)\bar{x}_i) - \sigma'(V^T(t)\bar{x}_i)V^T(t)\bar{x}_i)r_{b_i}^T(t)\Gamma_w, \end{aligned} \quad (29.27)$$

$$\begin{aligned} \dot{V}(t) = & -\Gamma_V \bar{x}(t)r^T(t)W^T(t)\sigma'(V^T(t)\bar{x}(t)) - k\|e(t)\|V(t) - \\ & V_c(t) \sum_{i=1}^p \Gamma_V \bar{x}_i r_{b_i}^T(t)W^T(t)\sigma'(V^T(t)\bar{x}_i), \end{aligned} \quad (29.28)$$

where W_c , V_c are orthogonal projection operators that restrict the update based on the recorded data in the null-space of update based on current data:

$$\begin{aligned} W_c = & I - \frac{(\sigma(V^T \bar{x}) - \sigma'(V^T \bar{x})V^T \bar{x})(\sigma(V^T \bar{x}) - \sigma'(V^T \bar{x})V^T \bar{x})^T}{(\sigma(V^T \bar{x}) - \sigma'(V^T \bar{x})V^T \bar{x})^T (\sigma(V^T \bar{x}) - \sigma'(V^T \bar{x})V^T \bar{x})}, \\ V_c = & I - \frac{\Gamma_V \bar{x} \bar{x}^T \Gamma_V}{\bar{x}^T \Gamma_V \Gamma_V \bar{x}}, \end{aligned} \quad (29.29)$$

with $\Gamma_W, \Gamma_V > 0$, $\kappa > 0$ with lower limit stated in the proof, and the external command $x_c(t)$ is such that $e_r(t) \in \Omega(P_r, \rho)$, for some $\rho > 0$, then, the command tracking error, e_r ; the reference model tracking error, e ; and adaptive element weights (\tilde{W}, \tilde{V}) are uniformly ultimately bounded. Further, the plant states, x , are ultimately bounded.

For the nonlinearly parameterized neural network, the simplest way to record a data point $x(t)$ online is to ensure that for a given $\bar{\theta} \in \mathbb{R}^+$,

$$\frac{\|x(t) - x_k\|^2}{\|x(t)\|} \geq \bar{\theta}, \quad (29.30)$$

where x_k is the last recorded data point. The points can be stored in an online history stack which contains a maximum of \bar{p} points. Once the maximum number of recorded points are reached, points are added such that the newest point replaces the oldest one.

29.4 Helicopter Specific Design

Consider the application of the combined inner-outer-loop adaptive architecture to the trajectory control of a helicopter. The dynamics (Munzinger 1998; Mettler 2002; Gavrillets et al. 2001) of the helicopter may be modeled in the same form as Eqs. (29.1)–(29.4). Most small helicopters include a Bell-Hiller stabilizer bar, which provides lagged rate feedback and is a source of unmodeled dynamics. The nonlinear model used for simulation in this work included the stabilizer bar dynamics. Additionally, blade flapping and other aspects such as gear and engine dynamics were also modeled.

29.4.1 Approximate Model

An approximate model for the attitude dynamics of the helicopter was generated by linearizing the nonlinear model around hover and neglecting coupling between the attitude and translational dynamics as well as the stabilizer bar:

$$\alpha_{des} = \hat{A}_1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \hat{A}_2 \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \hat{B} \left(\underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{des} - \underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{trim} \right), \quad (29.31)$$

or

$$\alpha_{des} = \hat{A}_1 \omega_B + \hat{A}_2 v_B + \hat{B}(\delta_{m_{des}} - \delta_{m_{trim}}),$$

where \hat{A}_1 and \hat{A}_2 represent the attitude and translational dynamics, respectively, and ω_B represents the angular velocity of the body with respect to the earth expressed in the body frame. The body velocity vector with respect to the earth expressed in the body frame is given by v_B , and $\delta_{m_{trim}}$ is the trim control vector that is consistent with the linear model. Choosing the control matrix \hat{B} such that it is invertible, the moment controls may be evaluated as

$$\delta_{m_{des}} = \hat{B}^{-1}(\alpha_{des} - \hat{A}_1\omega_B - \hat{A}_2v_B) + \delta_{m_{trim}}.$$

The translational dynamics may be modeled as a point mass with a thrust vector that may be oriented in a given direction as illustrated in Fig. 29.7. More involved inverses (Lipp and Prasad 1993) may be used, but the simple relationships between thrust, attitude, and accelerations suffice when used with adaptation

$$a_{des} = \begin{bmatrix} 0 \\ 0 \\ Z_{\delta_{coll}} \end{bmatrix} (\delta_{coll_{des}} - \delta_{coll_{trim}}) + L_{bv}g, \quad (29.32)$$

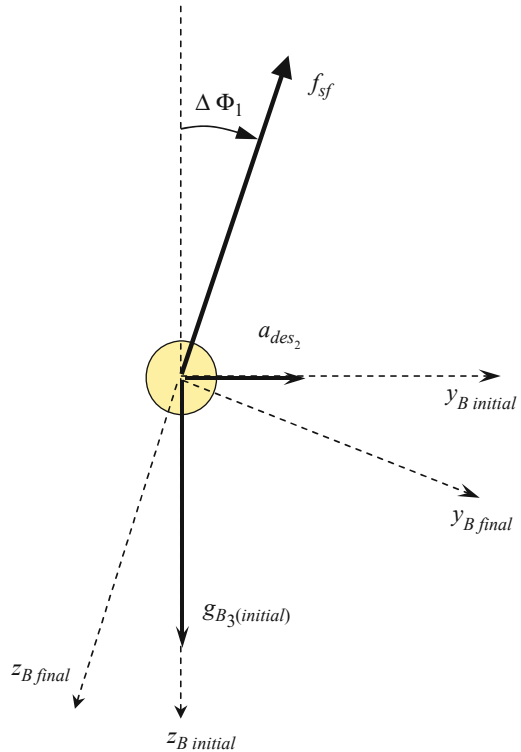


Fig. 29.7 Point mass model for outerloop inversion

where $Z_{\delta_{coll}}$ is the control derivative for acceleration in the vertical axis. L_{bv} is the direction cosine matrix that transforms a vector from the vehicle (or local) frame to the body frame and g is an assumed gravity vector. The desired specific force along the body z axis may be evaluated as

$$f_{sf} = (a_{des} - L_{bv}g)_3.$$

The required collective input may be evaluated as

$$\delta_{coll_{des}} = \frac{f_{sf}}{Z_{\delta_{coll}}} + \delta_{coll_{trim}}.$$

The attitude augmentation required in order to orient the thrust vector to attain the desired translational accelerations are given by the following small angle corrections from the current reference body attitude and attitude command:

$$\Delta\Phi_1 = \frac{a_{des2}}{f_{sf}}, \quad \Delta\Phi_2 = -\frac{a_{des1}}{f_{sf}}, \quad \Delta\Phi_3 = 0, \quad (29.33)$$

For this simplified helicopter model, heading change has no effect on accelerations in the x, y plane, and hence $\Delta\Phi_3 = 0$. These three correction angles may now be used to generate the attitude quaternion correction desired by the outer loop. Thus,

$$q_{des} = q(\Delta\Phi_1, \Delta\Phi_2, \Delta\Phi_3), \quad (29.34)$$

where $q(\cdot)$ is a function (Stevens and Lewis 2003) that expresses an Euler-angle-based rotation as a quaternion. The overall detailed controller architecture is shown in Fig. 29.6.

Remark 29.7. If the desired specific force f_{sf} is close to zero, which occurs when the desired acceleration in the body z axis is the same as the component of gravity vector along that axis, then, Eq. (29.33) is undefined. To overcome this problem, one can impose a restriction where 29.33 is only computed if $|f_{sf}| > \bar{f}_{sf}$, where $\bar{f}_{sf} > 0$ and is a lower limit. Essentially it means do not bother using attitude unless the desired specific force is greater than \bar{f}_{sf} .

29.4.2 Reference Model

Using a linear a_{cr} and α_{cr} in Eqs. (29.12) and (29.13) results in the following reference model dynamics:

$$\begin{aligned} \dot{v}_r &= R_p(p_c - p_r) + R_d(v_c - v_r) - a_h \\ \dot{\omega}_r &= K_p(\tilde{Q}(q_c \oplus q_{des}, q_r)) + K_d(\omega_c - \omega_r) - \alpha_h, \end{aligned}$$

where R_p, R_d, K_p, K_d are the same gains used for the PD compensator in Eq. (29.20). If limits on the angular rate or translational velocities are to be imposed, then they may be easily included in the reference model dynamics by choosing the following constrained linear reference for a_{cr} and α_{cr} :

$$a_{cr} = R_d[v_c - v_r + \sigma(R_d^{-1}R_p(p_c - p_r), v_{lim})], \quad (29.35)$$

$$\alpha_{cr} = K_d[\omega_c - \omega_r + \sigma(K_d^{-1}K_p\tilde{Q}(q_c \oplus q_{des}, q_r), \omega_{lim})]. \quad (29.36)$$

This reference model has prescribable aggressiveness, where $\sigma(\cdot)$ is a saturation function and v_{lim}, ω_{lim} are the translational and angular rate limits, respectively.

Remark 29.8. Note that there are no limits placed on the externally commanded position, velocity, angular rate, or attitude. For example, in the translational reference model, if a large position step is commanded, $p_c = [1,000, 0, 0]^T$ ft and $v_c = [0, 0, 0]^T$ ft/s, the speed at which this large step will be achieved is v_{lim} . On the other hand if $p_c = \int v_c dt$ and $v_c = [60, 0, 0]^T$ ft/s, the speed of the vehicle will be 60 ft/s. Similarly, ω_{lim} dictates how fast large attitude errors will be corrected. Additionally, aggressiveness with which translational accelerations will be pursued by tilting the body may be governed by limiting the magnitude of q_{des} to the scalar limit q_{lim} .

29.4.3 Choice of Gain Linear Dynamics

When the combined adaptive inner-outer-loop controller for position and attitude control is implemented, the poles for the combined error dynamics must be selected appropriately. The following analysis applies to the situation where inversion model error is compensated for accurately by the NN, and it is assumed that the system is exactly feedback linearized. The inner loop and outer loop each represent a second-order system, and the resulting position dynamics $p(s)/p_c(s)$ are fourth order in directions perpendicular to the rotor spin axis.

When the closed-loop longitudinal dynamics, near hover, are considered, and with an acknowledgment of an abuse of notation, it may be written as

$$\ddot{x} = a_{des} = \ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x), \quad (29.37)$$

$$\ddot{\theta} = \alpha_{des} = \ddot{\theta}_g + K_d(\dot{\theta}_g - \dot{\theta}) + K_p(\theta_g - \theta), \quad (29.38)$$

where R_p, R_d, K_p , and K_d are the PD compensator gains for the inner loop (pitch angle) and outer loop (fore-aft position). Now x is the position, θ the attitude, and θ_g the attitude command. Normally, $\theta_g = \theta_c + \theta_{des}$ where θ_c is the external command and θ_{des} the outer-loop-generated attitude command. Here, it is assumed that the external attitude command and its derivatives are zero; hence, $\theta_g = \theta_{des}$. In the following development, the transfer function $x(s)/x_c(s)$ is found and used to place

the poles of the combined inner-outer-loop system in terms of the PD compensator gains.

When contributions of $\dot{\theta}_g(s)$ and $\ddot{\theta}_g(s)$, are ignored, the pitch dynamics Eq. 29.38 may be rewritten in the form of a transfer function as

$$\theta(s) = \frac{\theta(s)}{\theta_g(s)} \theta_g(s) = \frac{K_p}{s^2 + K_d s + K_p} \theta_g(s). \quad (29.39)$$

If the outer-loop linearizing transformation used to arrive at Eq. 29.37 has the form $\ddot{x} = f\theta$, where $f = -g$ and g is gravity, it may be written as

$$s^2 x(s) = f\theta(s). \quad (29.40)$$

The outer-loop attitude command may be generated as

$$\theta_{des} = \frac{\ddot{x}_{des}}{f} = \frac{a_{des}}{f}. \quad (29.41)$$

Note that $\theta_g = \theta_{des}$; if $\theta_c = 0$,

$$\theta_g = \theta_{des} = \frac{1}{f} [\ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x)]. \quad (29.42)$$

When Eqs. (29.39) and (29.42) are used in Eq. 29.40,

$$s^2 x(s) = \frac{K_p [s^2 x_c + R_d s(x_c - x) + R_p(x_c - x)]}{s^2 + K_d s + K_p}. \quad (29.43)$$

Rearranging the above equation results in the following transfer function:

$$\frac{x(s)}{x_c(s)} = \frac{K_p s^2 + K_p R_d s + K_p R_p}{s^4 + K_d s^3 + K_p s^2 + K_p R_d s + K_p R_p}. \quad (29.44)$$

One way to choose the gains is by examining a fourth-order characteristic polynomial written as the product of two second-order systems

$$\begin{aligned} \Upsilon(s) &= (s^2 + 2\zeta_o \omega_o + \omega_o^2)(s^2 + 2\zeta_i \omega_i + \omega_i^2) \\ &= s^4 + (2\zeta_i \omega_i + 2\zeta_o \omega_o) s^3 \\ &\quad + (\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2) s^2 + (2\zeta_o \omega_o \omega_i^2 + 2\omega_o^2 \zeta_i \omega_i) s + \omega_o^2 \omega_i^2, \end{aligned} \quad (29.45)$$

where the subscripts i, o represent the inner- and outer-loop values, respectively.

Comparing the coefficients of the poles of Eqs. (29.44) and (29.45) allows the gains to be expressed as a function of the desired pole locations for each axis in turn

$$\begin{aligned}
R_p &= \frac{\omega_o^2 \omega_i^2}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \\
R_d &= 2 \frac{\omega_o \omega_i (\zeta_o \omega_i + \omega_o \zeta_i)}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \\
K_p &= \omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2 \\
K_d &= 2\zeta_i \omega_i + 2\zeta_o \omega_o.
\end{aligned} \tag{29.46}$$

Additionally, the zeros of the transfer function given by Eq. 29.44 affect the transient response. Thus, ω_i , ζ_i , ω_o , ζ_o must be selected such that performance is acceptable.

29.4.4 Imposing Response Characteristics

The methods presented in this chapter do not contain assumptions that limit its application to unmanned helicopters. Manned rotorcraft normally have to meet standards, such as those specified in the Aeronautical Design Standard-33 (2000) handling qualities specifications. Control system performance (Civita et al. 2002b; Rysdyk and Calise 2005) may be evaluated by imposing response requirements and computing metrics prescribed in the ADS-33. When there is no saturation, the hedging signals a_h , α_h are zero. When it is assumed that the adaptation has reached its ideal values of (V^*, W^*) , then

$$\begin{aligned}
\dot{v} &= a_{cr} + a_{pd} + \epsilon_a \\
\dot{\omega} &= \alpha_{cr} + \alpha_{pd} + \epsilon_\alpha,
\end{aligned}$$

where ϵ_a and ϵ_α are bounded by $\bar{\epsilon}$. Additionally, the Lyapunov analysis provides guaranteed model following, which implies a_{pd} and α_{pd} are small. Thus, $\dot{v} \approx a_{cr}$ and $\dot{\omega} \approx \alpha_{cr}$. Hence, as long as the preceding assumptions are valid over the bandwidth of interest, the desired response characteristics may be encoded into the reference model a_{cr} and α_{cr} .

29.5 Experimental Results

The proposed guidance and control architecture was applied to the Georgia Institute of Technology Yamaha RMAX helicopter (GTMax) shown in Fig. 29.2. The GTMax helicopter weighs about 157 lb and has a main rotor radius of 5.05 ft. Nominal rotor speed is 850 revolutions per minute. Its practical payload capability is about 66 lb with a flight endurance of greater than 60 min. It is also equipped with a Bell-Hillier stabilizer bar. Its avionics package includes a Pentium 266 flight control computer, an inertial measurement unit (IMU), a global positioning system, a 3-axis magnetometer, and a sonar altimeter. The control laws presented in this

chapter were first implemented in simulation (Kannan et al. 2004) using a nonlinear helicopter model that included flapping and stabilizer bar dynamics. Wind and gust models were also included. Additionally, models of sensors with associated noise characteristics were implemented. Many aspects of hardware such as the output of sensor model data as serial packets was simulated. This introduced digitization errors as would exist in real life and also allowed testing of many flight-specific components such as sensor drivers. The navigation system (Christophersen et al. 2006) consists of a 17-state Kalman filter to estimate variables such as attitude and terrain altitude. The navigation filter was executed at 100 Hz and corresponds to the highest rate at which the IMU is able to provide data. Controller calculations occurred at 50 Hz. The control laws were first implemented as C-code and tested in simulation. Because almost all aspects specific to flight testing were included in the simulation environment, a subset of the code from the simulation environment was implemented on the main flight computer. During flight, Ethernet and serial-based data links provided a link to the ground station computer that allowed monitoring and uploading of waypoints. A simple kinematics-based trajectory generator (with limits on accelerations) was used to generate smooth consistent trajectories (p_c, v_c, q_c, ω_c) for the controller. Various moderately aggressive maneuvers were performed during flight to test the performance of the trajectory tracking controller. Controller testing began with simple hover followed by step responses and waypoint navigation. Following initial flight tests, aggressiveness of the trajectory was increased by relaxing acceleration limits in the trajectory generator and relaxing ω_{lim} and v_{lim} in the reference models. Tracking error performance was increased by increasing the desired bandwidth of the controllers. Selected results from these flight tests are provided in the following sections.

29.5.1 Parameter Selections

The controller parameters for the inner loop involved choosing K_p, K_d based on a natural frequency of 2.5, 2, 3 rad/s for the roll, pitch, and yaw channels, respectively, and damping ratio of 1.0. For the outer loop, R_p, R_d were chosen based on a natural frequency of 2, 2.5, 3 rad/s for the x, y, and z body axes, all with a damping ratio of unity. The NN was chosen to have five hidden-layer neurons. The inputs to the network included body axis velocities and rates as well as the estimated pseudocontrols, that is, $x_{in} = [v_B^T, \omega_B^T, \hat{a}^T, \hat{\alpha}^T]$. The output layer learning rates Γ_W were set to unity for all channels, and a learning rate of $\Gamma_V = 10$ was set for all inputs. Limits on maximum translation rate and angular rate in the reference model dynamics were set to $v_{lim} = 10$ ft/s and $\omega_{lim} = 2$ rad/s. Additionally, attitude corrections from the outer loop, q_{des} were limited to 30° .

With regard to actuator magnitude limits, the helicopter has a radio-control transmitter that the pilot may use to fly the vehicle manually. The full deflections available on the transmitter sticks in each of the channels were mapped as $\delta_{lat}, \delta_{lon}, \delta_{ped} \in [-1, 1]$, corresponding to the full range of lateral tilt and longitudinal tilt of the swash plate and full range of tail rotor blade pitch. The collective

was mapped as $\delta_{coll} \in [-2.5, 1]$, corresponding to the full range of main rotor blade pitch available to the human pilot. The dynamic characteristics of the actuators were not investigated in detail. Instead, conservative rate limits were artificially imposed in software. Noting that $\delta = [\delta_{coll}, \delta_{lat}, \delta_{lon}, \delta_{ped}]^T$, the actuator model used for PCH purposes as well as artificially limiting the controller output has form

$$\dot{\delta} = \lim_{\lambda \rightarrow +\infty} \sigma \left(\lambda (\sigma(\delta_{des}, \delta_{\min}, \delta_{\max}) - \hat{\delta}), \dot{\delta}_{\min}, \dot{\delta}_{\max} \right), \quad (29.47)$$

where $\hat{\delta}$ is limited to lie in the interval $[\delta_{\min}, \delta_{\max}]$. The discrete implementation has the form

$$\begin{aligned} \hat{\delta}[k+1] = \sigma \left(\hat{\delta}[k] + \sigma \left(\sigma(\delta_{des}, \delta_{\min}, \delta_{\max}) - \hat{\delta}[k], \Delta T \dot{\delta}_{\min}, \Delta T \dot{\delta}_{\max} \right), \right. \\ \left. \delta_{\min}, \delta_{\max} \right), \end{aligned} \quad (29.48)$$

where ΔT is the sampling time. The magnitude limits were set to

$$\begin{aligned} \delta_{\min} &= [-2.5, -1, -1, -1]^T \\ \delta_{\max} &= [1, 1, 1, 1]^T \end{aligned} \quad (29.49)$$

units, and the rate limits were set to

$$\begin{aligned} \dot{\delta}_{\min} &= [-4, -2, -2, -2]^T \\ \dot{\delta}_{\max} &= [4, 2, 2, 2]^T \end{aligned} \quad (29.50)$$

units per second.

29.5.2 Flight Test

Finally, the controller was flight tested on the GTMax helicopter shown in Fig. 29.2. A lateral-position step response is shown in Fig. 29.8. The vehicle heading was regulated due north during this maneuver. Lateral control deflections during the maneuver were recorded and are also shown. A step heading command response and pedal control history is shown in Fig. 29.9. It should be noted that during flight tests, states were sampled at varying rates in order to conserve memory and data link bandwidth. The trajectory commands p_c, v_c, q_c, ω_c were sampled at 1 Hz; actuator deflections $\delta_{coll}, \delta_{lon}, \delta_{lat}$, and δ_{ped} were sampled at 50 Hz; and vehicle position and speed was sampled at 50 Hz. Since the command vector is sampled at a low rate (1 Hz), a step command appears as a fast ramp in figures.

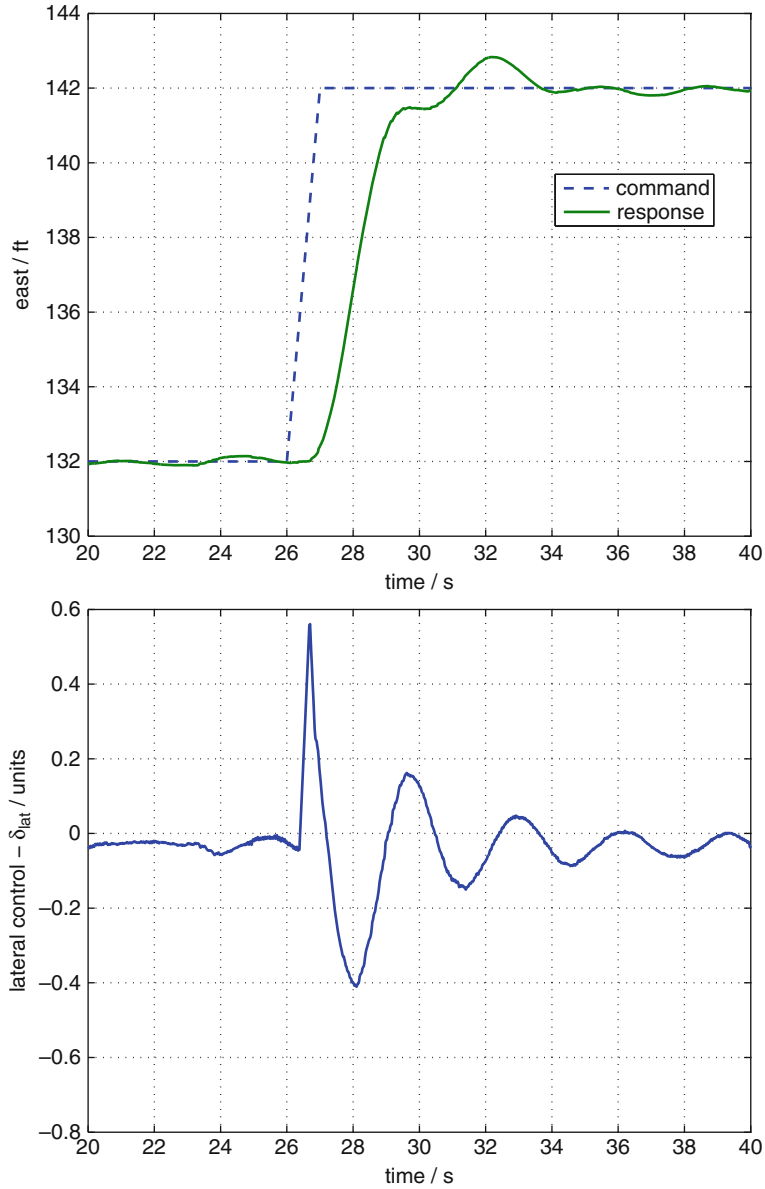


Fig. 29.8 Response to a 20-ft step in the lateral direction

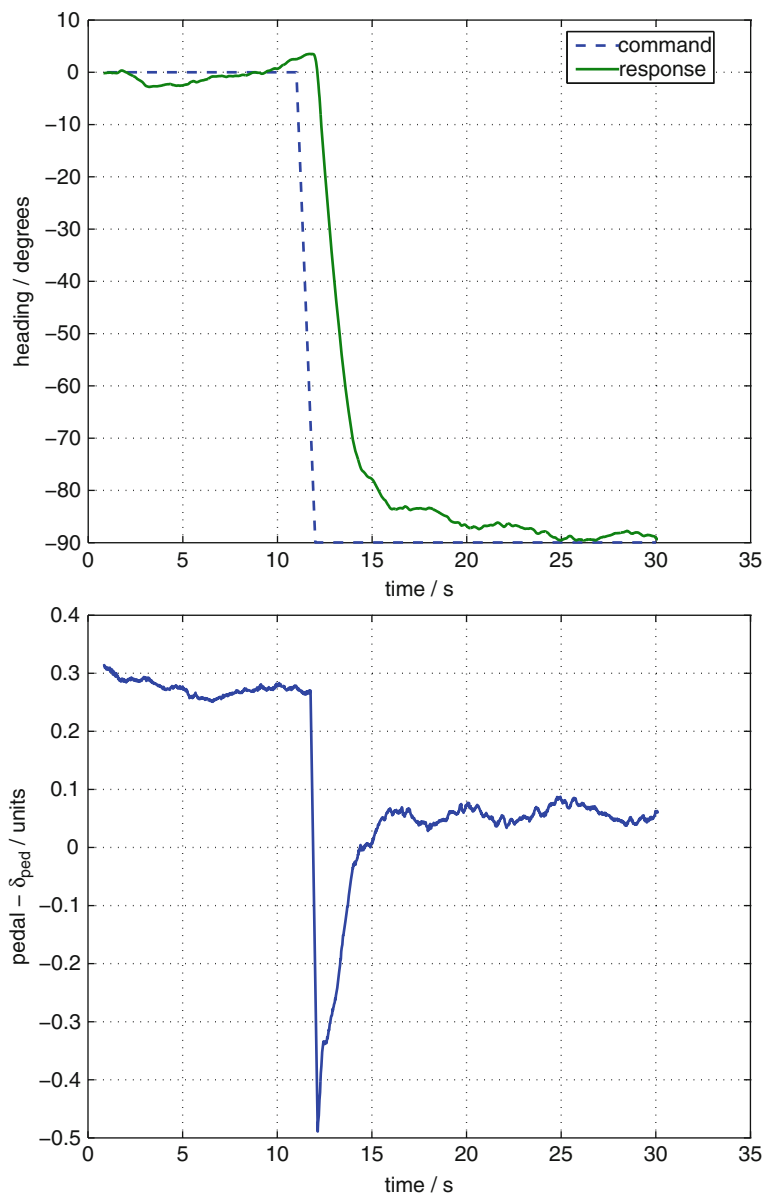


Fig. 29.9 Response to a 90° heading command

During takeoff and landing phases, a range sensor (sonar) is used to maintain and update the estimated local terrain altitude in the navigation system. The sonar is valid up to 8 ft above the terrain, sufficient for landing and takeoff purposes. Figure 29.10 illustrates the altitude and collective profile during a landing. The vehicle starts at an initial hover at 300 ft, followed by a descent at 7 ft/s until the vehicle is 15 ft above the estimated terrain. The vehicle then descends at 0.5 ft/s until weight on skids is automatically detected at which point the collective is slowly ramped down. Automatic takeoff (Fig. 29.11) is similar where the collective is slowly ramped up until weight on skids is no longer detected. It should be noted that NN adaptation is active at all times except when weight on skids is active. Additionally, when weight is on skids, the collective ramp-up during takeoff and ramp-down during landing is open loop.

The approximate model used to compute the dynamic inverse (Eqs. (29.32) and (29.31)) is based on a linear model of the dynamics in hover. To evaluate controller performance at different points of the envelope, the vehicle was commanded to track a trajectory that accelerated up to a speed of 100 ft/s. To account for wind, an upwind and downwind legs were flown. In the upwind leg, the vehicle accelerated up to 80 ft/s, and during the downwind leg, the vehicle accelerated up to a speed of 97 ft/s as shown in Fig. 29.12. Collective and longitudinal control deflections are also shown. In the upwind leg, the collective is saturated and the vehicle is unable to accelerate further. The longitudinal control deflections behave nominally as the vehicle accelerates and decelerates through a wide range of the envelope. The NN is able to adapt to rapidly changing flight conditions, from the baseline inverting design at hover through to the maximum speed of the aircraft. A conventional proportional-integral-derivative design would have required scheduling of gains throughout the speed range. More significantly, classical design would require accurate models at each point, unlike this design which does not. In addition to flight at high speeds, tracking performance was evaluated at moderate speeds, where a square pattern was flown at 30 ft/s for which position tracking is shown in Fig. 29.13. External command position tracking errors are shown in Fig. 29.14 with a peak total position error 3.3 ft and standard deviation of 0.8 ft.

Many maneuvers such as high-speed flight are quasi steady, in the sense that once in the maneuver, control deflection changes are only necessary for disturbance rejection. To evaluate performance where the controls have to vary significantly in order to track the commanded trajectory, the helicopter was commanded to perform a circular maneuver in the north-east plane with constant altitude and a constantly changing heading. The trajectory equations for this maneuver are given by

$$p_c = \begin{bmatrix} \frac{V}{\omega} \cos(\omega t) \\ \frac{V}{\omega} \sin(\omega t) \\ -h \end{bmatrix}, \quad v_c = \begin{bmatrix} -V \sin(\omega t) \\ V \cos(\omega t) \\ 0 \end{bmatrix},$$

$$\psi_c = \omega t f,$$

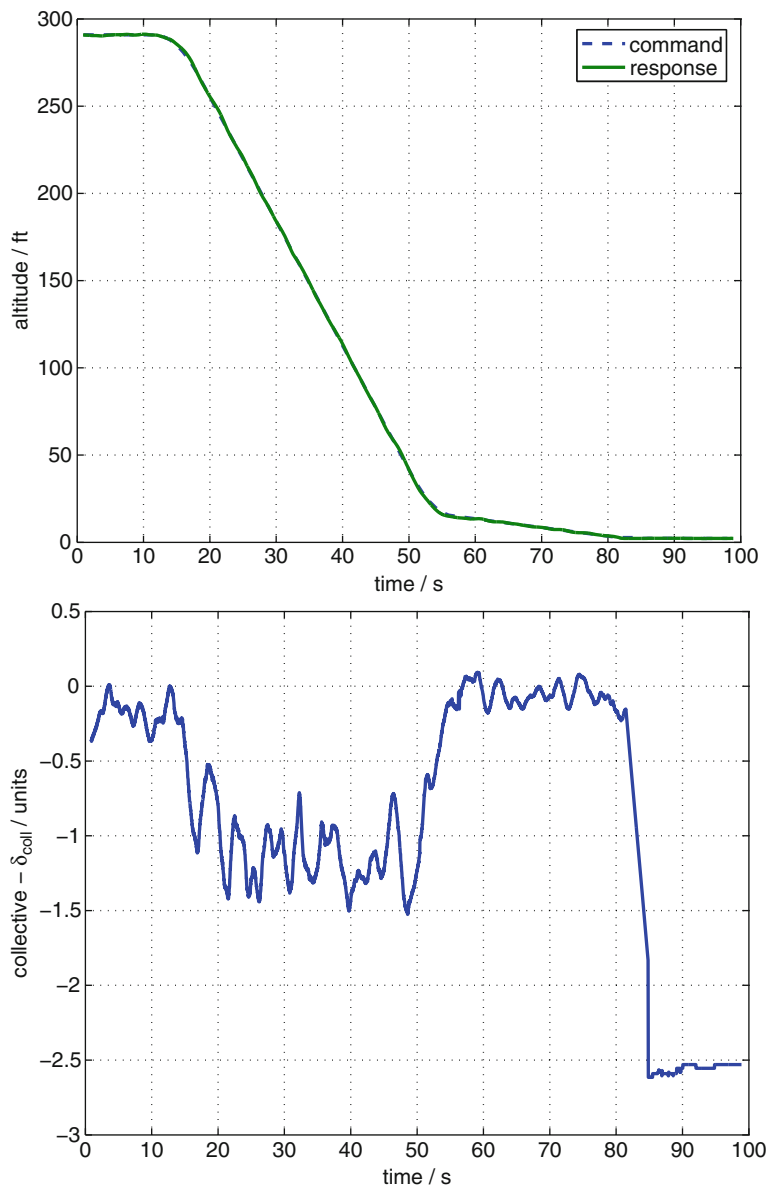


Fig. 29.10 Automatic landing maneuver

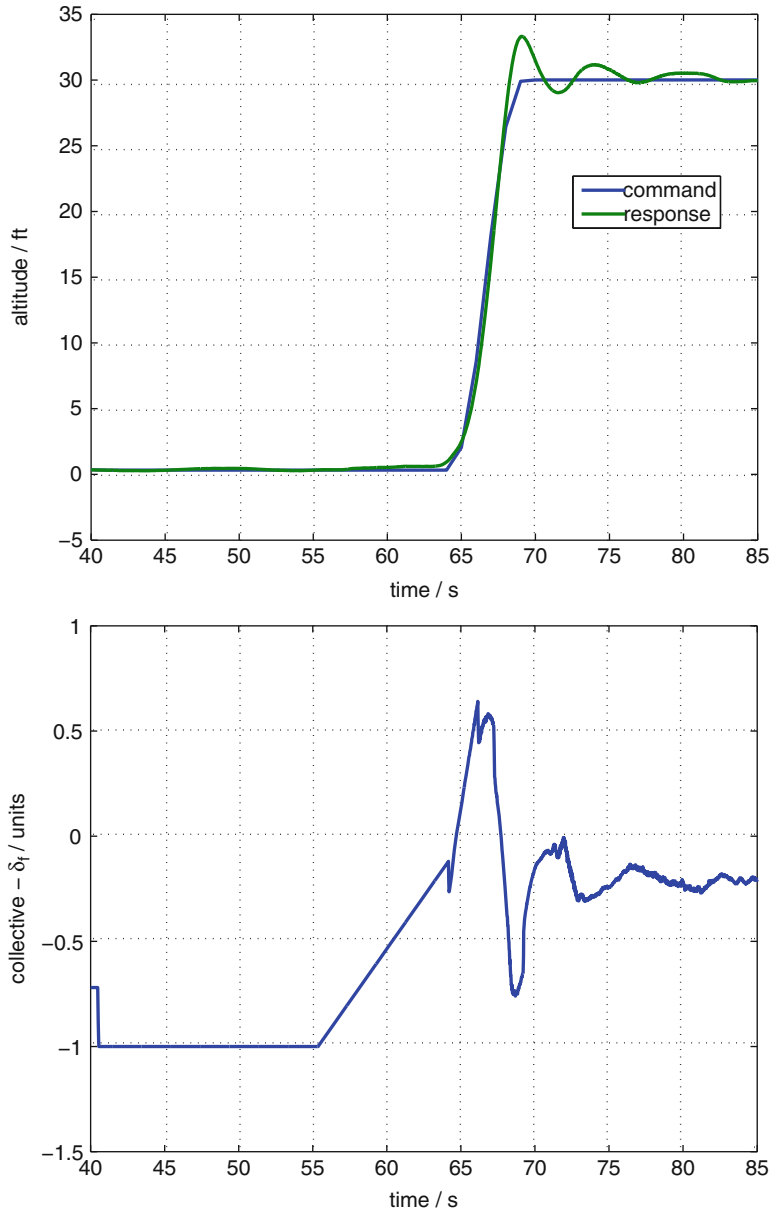
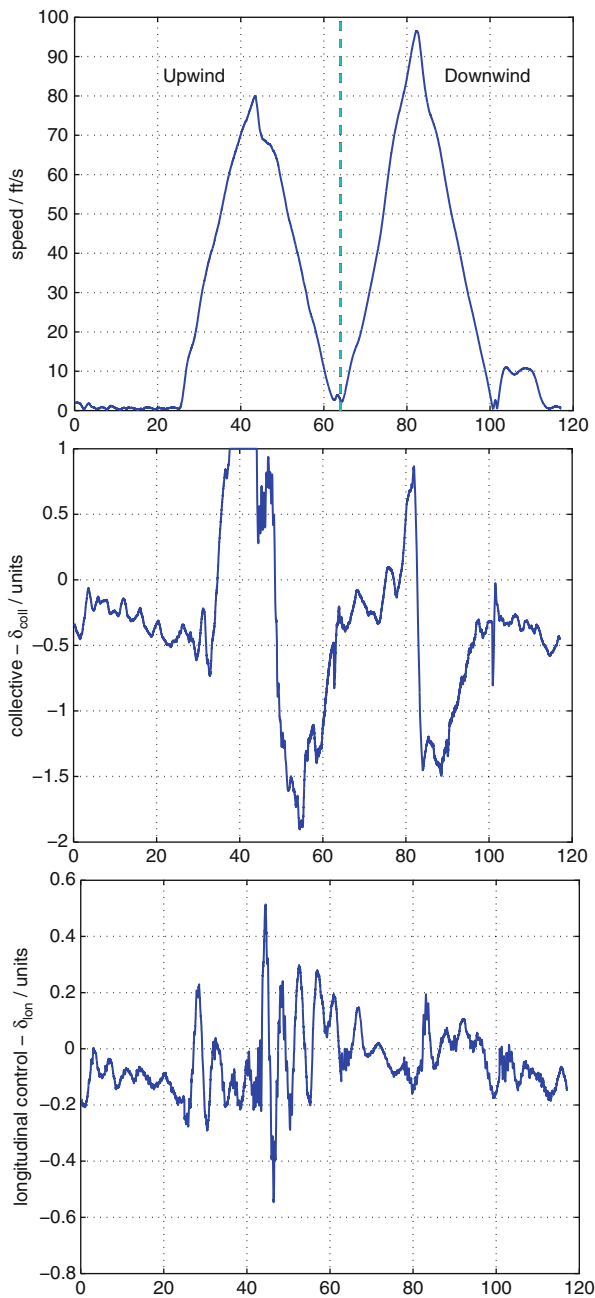


Fig. 29.11 Automatic takeoff maneuver

Fig. 29.12 High-speed forward flight up to 97 ft/s



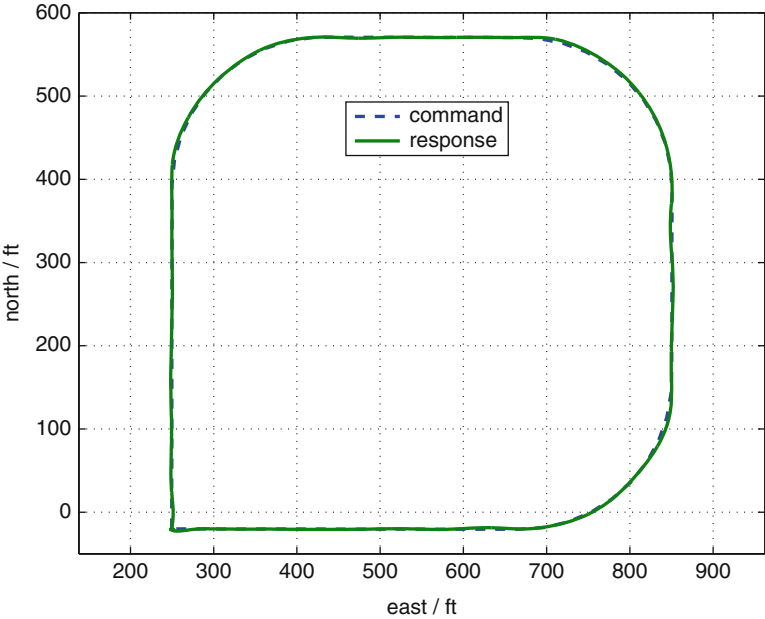


Fig. 29.13 Flying a square pattern at 30 ft/s

where t is current time and h is a constant altitude command. V is speed of the maneuver, ω is angular speed of the helicopter around the maneuver origin, and f is number of 360° changes in heading to be performed per circuit. If $\omega = \pi/2$ rad/s, the helicopter will complete the circular circuit once every 4 s. If $f = 1$, the helicopter will rotate anticlockwise 360° once per circuit. Figure 29.15 shows the response to such a trajectory with parameters $\omega = 0.5$ rad/s, $f = 1$, $V = 10$ ft/s. After the initial transition into the circular maneuver, the tracking is seen to be within 5 ft. To visualize the maneuver easily, superimposed still images of the vehicle during the circular maneuver are shown. Both anticlockwise and clockwise heading changes during the maneuver were tested by changing the parameter from $f = 1$ (anticlockwise) to $f = -1$ (clockwise) at $t = 55$ s. Figure 29.16 shows that heading tracking is good in both cases. The time history of the pedal input δ_{ped} and all other controls during the maneuver is also shown and illustrates how the vehicle has to exercise all of its controls during this maneuver.

Next, the ability of the controller to track a previous manually flown maneuver was tested. First, a human pilot flew a figure-eight, three-dimensional pattern with the vehicle. Vehicle state was recorded and was then played back as commands to the adaptive controller. A 3D plot of the pilot and controller flown trajectories are shown in Fig. 29.17 along with projected ground track. Overall, the tracking in position was measured to be within 11.3 ft of the desired pilot-flown trajectory with a standard deviation of 4.7 ft.

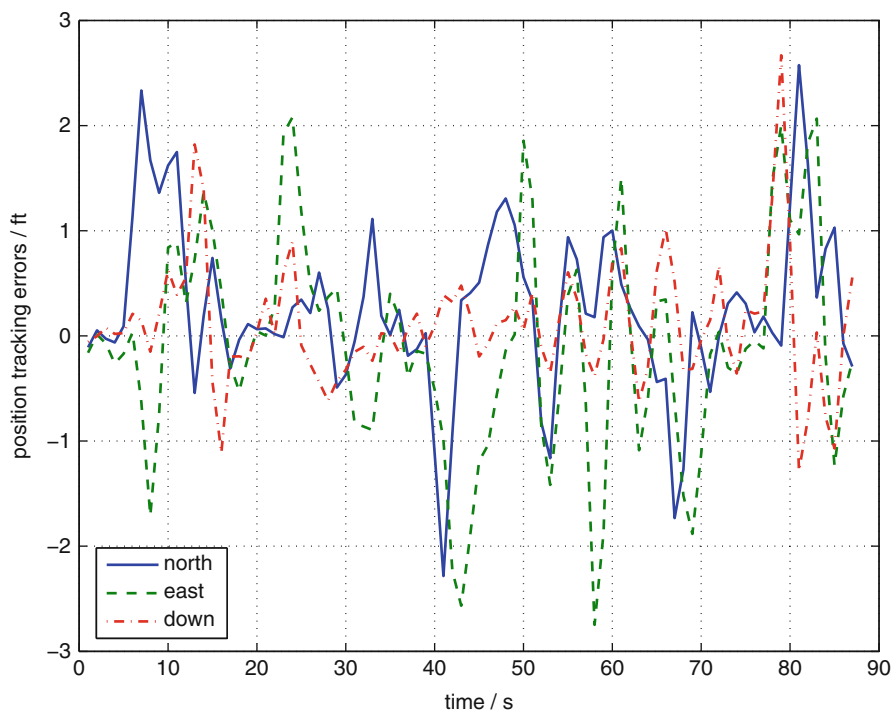


Fig. 29.14 Command tracking errors while flying a square pattern at 30 ft/s

Finally, a tactically useful maneuver was flown to test controller performance at high speeds and pitch attitudes. The objective of the maneuver is to make a 180-degree velocity change from a forward flight condition of 70 ft/s north to a 70 ft/s forward flight going south. The trajectory command and response in the north-altitude plane is shown in Fig. 29.18 along with the pitch angle. A time history of the altitude and the collective control deflection is shown in Fig. 29.19. During the maneuver, the helicopter is commanded to increase altitude by up to 50 ft in order to minimize saturation of the down collective. In the deceleration phase, the vehicle is able to track the command trajectory well; however, in accelerating to 70 ft/s going south, tracking performance suffers. In both the acceleration and deceleration phases, poor tracking corresponds with saturation of the collective control. The oscillations in altitude in Fig. 29.19 are expected and are due to control saturation which limits the vehicle's descent rate. The large pitch attitudes experienced are what the outer-loop inversion evaluates as being required to perform such rapid decelerations and accelerations. This experiment is an example of maneuvering where the commanded trajectory is more aggressive than the capability of the vehicle and is reflected by the extended periods of saturation. It is possible to operate at the limits of the vehicle primarily due to PCH which protects the adaptation process.

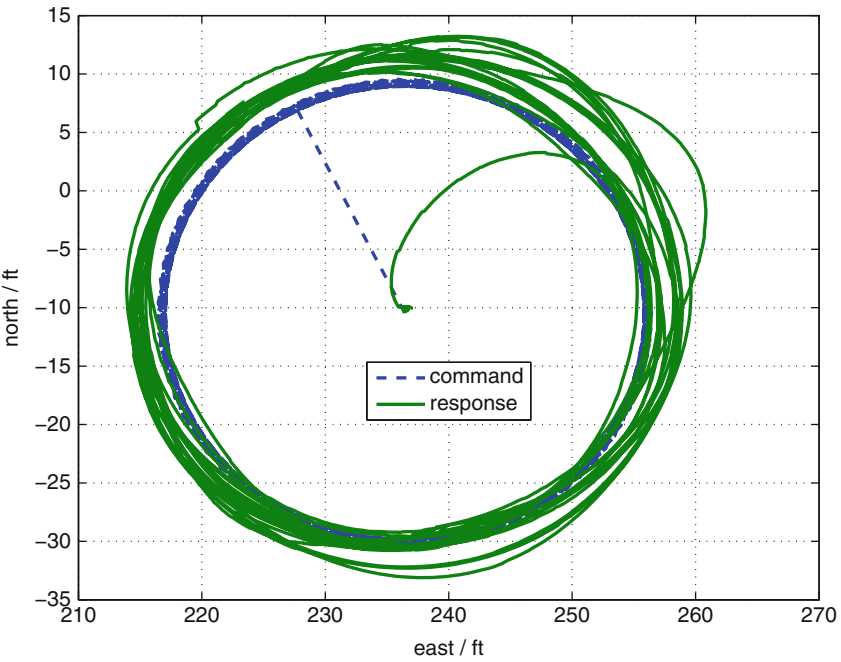


Fig. 29.15 Circular maneuver, with 360° heading changes during the circuit

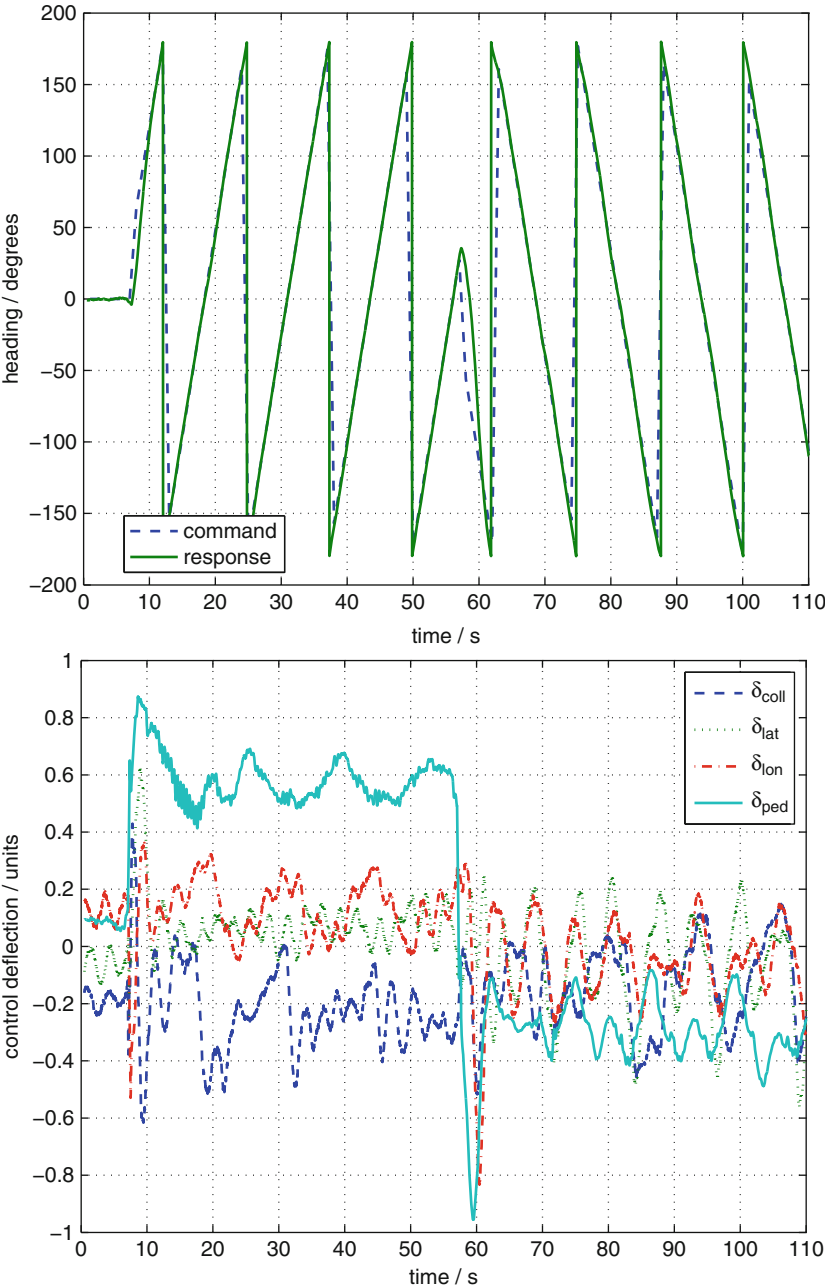


Fig. 29.16 Heading tracking during circular maneuver and control time history

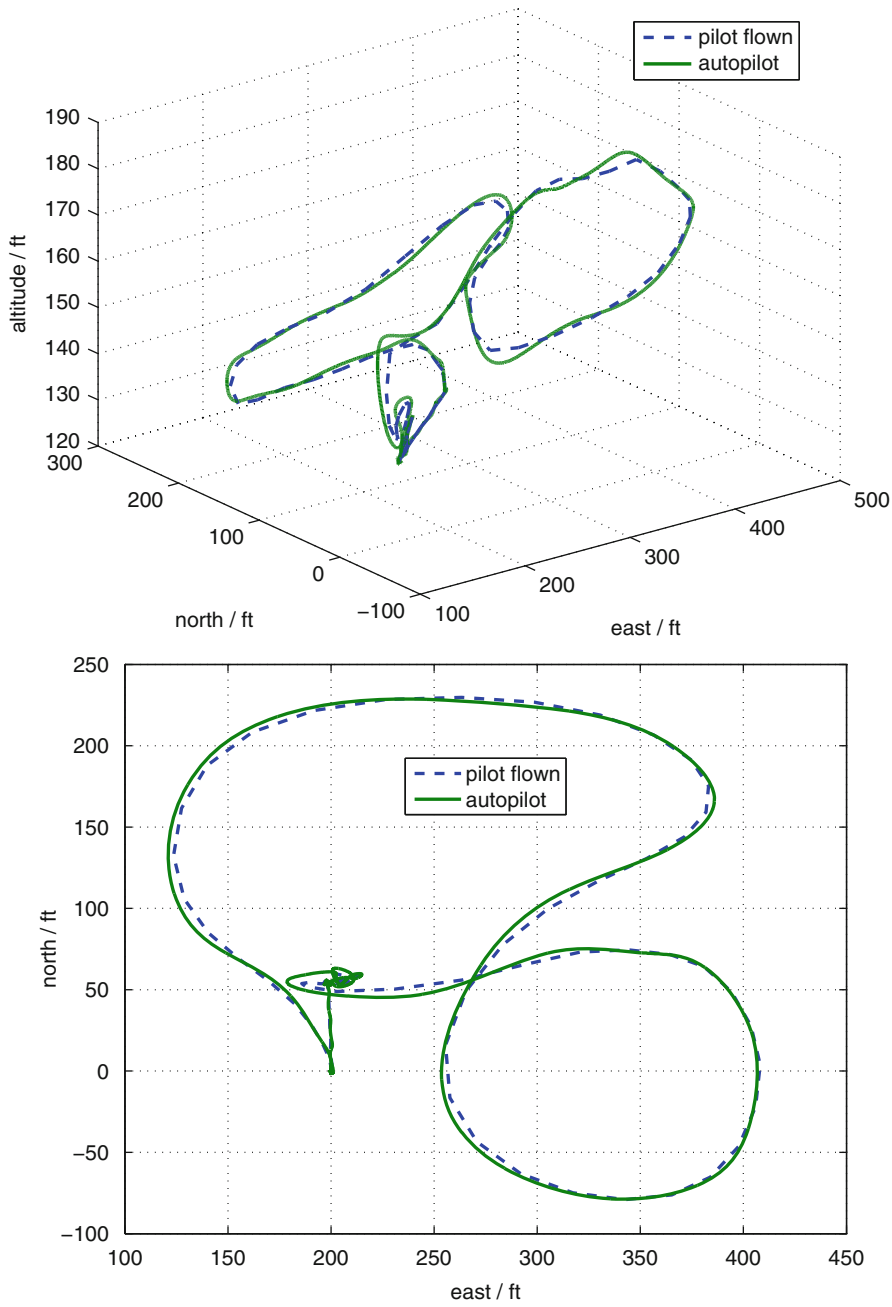


Fig. 29.17 A 3-D view and ground track view, of a trajectory initially flown manually by a pilot and then tracked by the controller

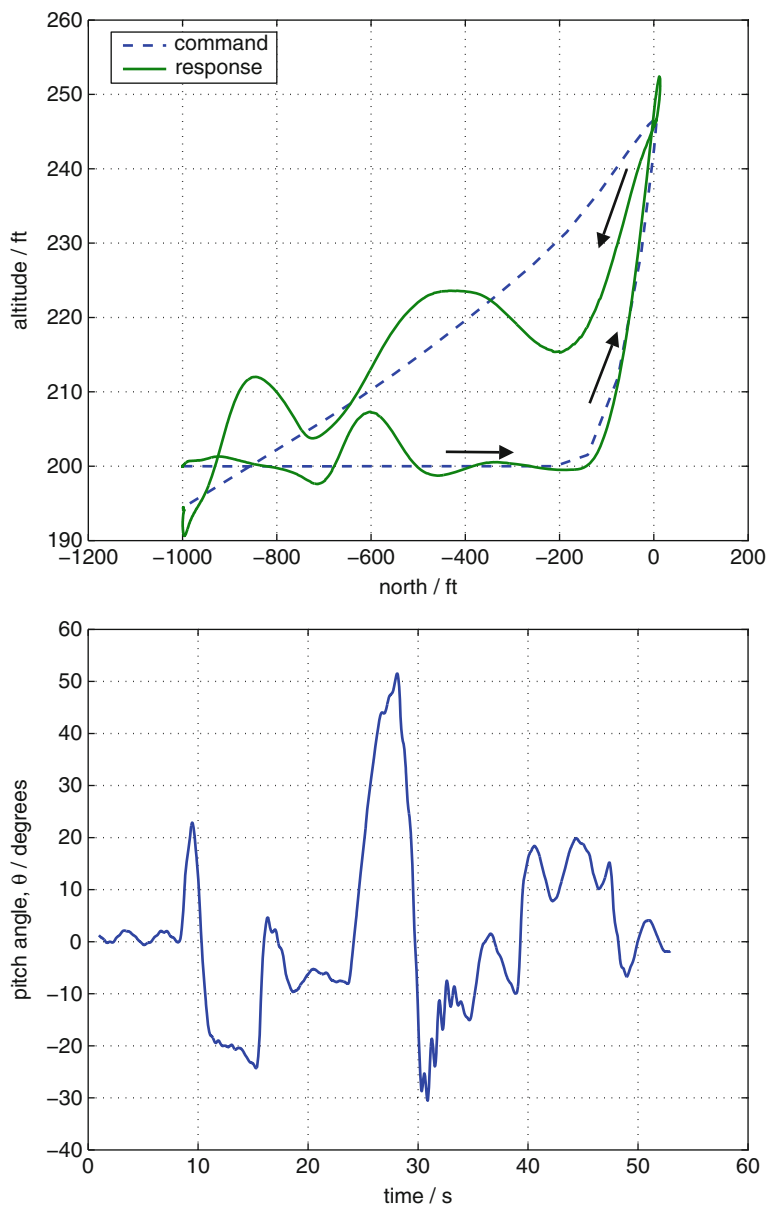


Fig. 29.18 North-altitude and pitch angle profile during a 180° velocity change maneuver. *Note: North axis and Altitude axis scales are not equal*

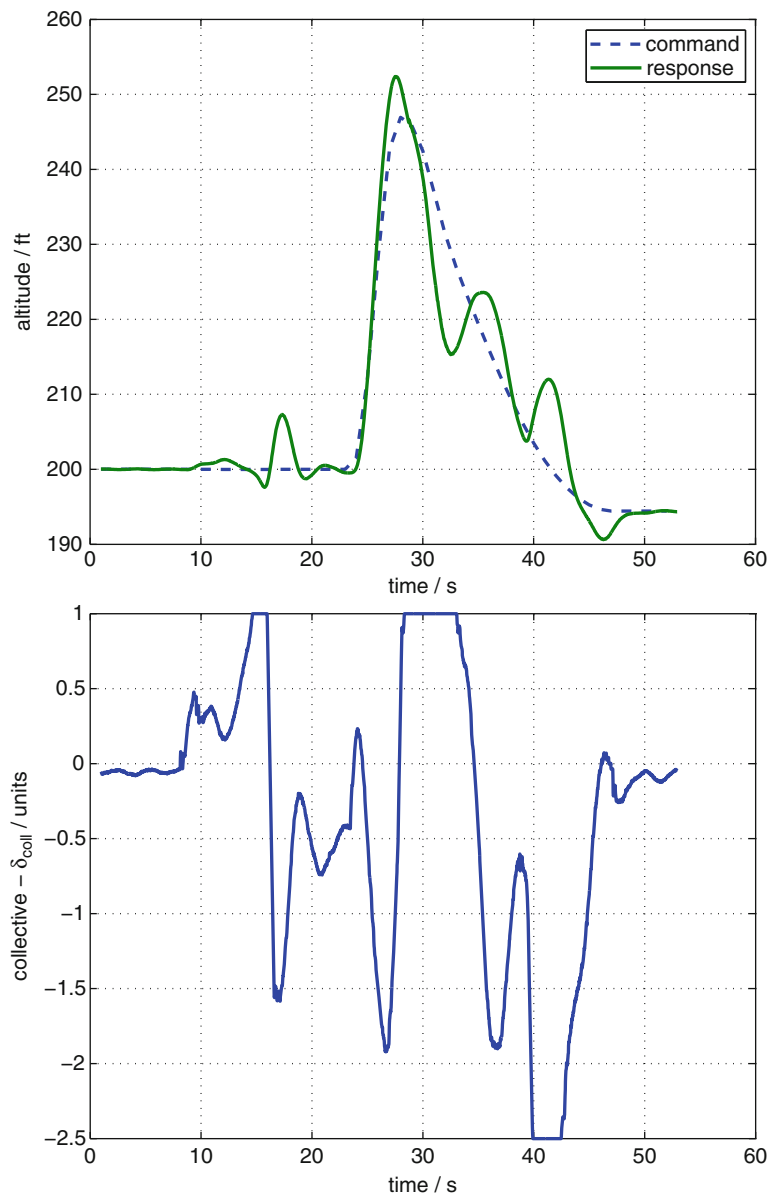


Fig. 29.19 Altitude and collective control history during a 180° velocity change maneuver

29.5.3 Application to a Ducted Fan

Following tests on the GTMax helicopter, the control method presented in this chapter was applied to other smaller aircraft. The algorithms were ported to a custom DSP/FPGA hardware device (the FCS20) along with a small sensor board that contained gyroscopes and accelerometers for inertial sensing and a GPS. The avionics package weighed less than 1 lb and fell within the payload capacity of the 11-in. ducted fan (GTSpy). The GTSpy has a maximum takeoff weight of 5.5 lb and is driven by a two-bladed fixed-pitch propeller. The propeller is enclosed in an annular wing duct with an outer diameter of 11 in. Vanes located directly beneath the propeller move in order to provide yaw control about the propeller axis. Two sets of control surfaces located further below the propeller move in order to provide pitch and roll moments. Maneuvering is accomplished by tilting the thrust vector with the control surfaces relying primarily on inflow for dynamic pressure during hover. Following satisfactory tethered tests, the vehicle was untethered and allowed to fly simple missions. Figure 29.20 shows a plan view of a small 50-ft box maneuver and the GTSpy's tracking. The large deviation on the eastern side of the box is most likely due to a wind gust. Another maneuver performed was the mid-air deployment of the GTSpy. The GTSpy was mounted on the GTMax helicopter with its engine on and then deployed from a safe altitude. The GTSpy was able to recover from the initial deployment transient and maintain attitude and position within 5 s of launch. Figure 29.21 shows the GTSpy and GTMax during

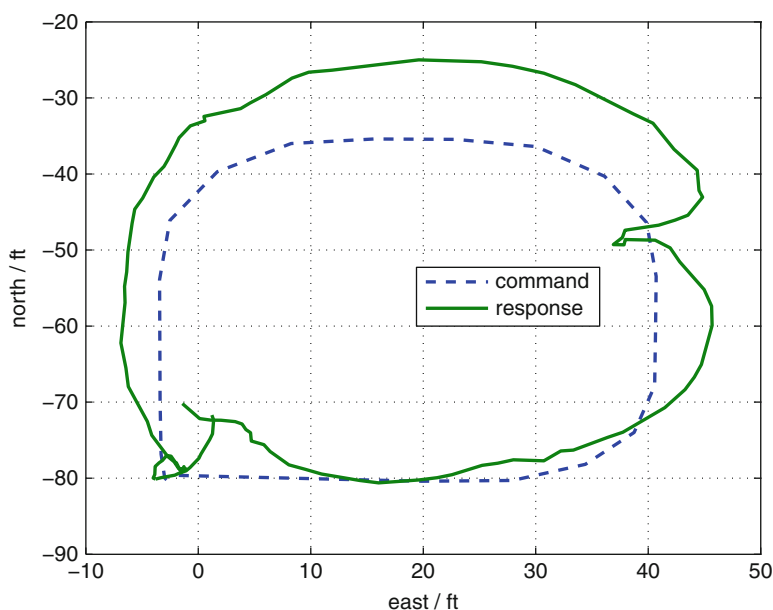


Fig. 29.20 The GTSpy performing a box maneuver

Fig. 29.21 Deployment of the GTSpy ducted fan from the GTMax helicopter



the deployment transient. Both the GTMax and GTSpy were under computer control during this maneuver and is the first known deployment of a rotorcraft from another rotorcraft.

29.5.4 Application to a Fixed-Wing Aircraft

The control method presented in this chapter was further applied to a high thrust-to-weight ratio, fixed-wing aircraft with conventional aircraft controls and a fixed-pitch two-bladed propeller. The dynamic inverse used for control purposes approximated the aircraft in hover mode where the body axis was defined as

$$x_{heli} = L_2(-\pi/2)x_{airplane}$$

where L_2 is a rotation matrix around the airplane's body y-axis. Hence, the ailerons control helicopter yaw, the rudder controls helicopter roll, and the elevators continue to control pitch. The external commands provided to the control algorithm contains a commanded pitch angle as a function of speed. Inner-loop gains were based on 2.5, 1.5, 2.5 rad/s for the (helicopter) roll, pitch and yaw axes, respectively. Outer-loop gains were based on 1.5, 1.0, 0.7 rad/s for the x, y, and z helicopter body axes, respectively. The output layer learning rates Γ_W were set to unity on all channels and a learning rate of Γ_V was set for all inputs. Reference model parameters were set to $v_{lim} = 10$ ft/s and $\omega_{lim} = 1.0$ rad/s. The control effectiveness B was scaled based on speed in order to reflect the reduced control authority of the control surfaces in hover. Flight tests were initiated with the airplane performing circular orbits and

gradually lowering airspeed until hover. The reverse transition to forward flight was accomplished by a more aggressive command into forward flight.

The following figures illustrate the response of the aircraft during transitions between hover and forward flight. Figure 29.22 shows the vehicle in forward flight at 80 ft/s performing a circular orbit. At $t = 26$ s a transition to hover is initiated by supplying external trajectory commands that lower the vehicle's speed. Transition is completed at $t = 35$ s with a low residual speed of approximately 5 ft/s. At $t = 55$ s a transition back to forward flight at 80 ft/s is initiated and completed at $t = 65$ s. During hover, $t \in [35, 55]$, the control deflections are seen to be significantly higher due to the lower effectiveness at lower speeds. The ailerons are saturated for significant intervals in a particular direction in order to counteract engine torque.

Figure 29.23 illustrates the (helicopter) pitch angle during transitions as well as the throttle control deflections. In forward flight, the pitch angle is approximately -75° and varies in hover due to reduced control effectiveness and the presence of a steady wind. Additionally, Fig. 29.24 shows the position trajectory during transitions, whereas Fig. 29.25 is a snapshot of the aircraft during the maneuver.

29.5.5 Implementation of Concurrent-Learning Adaptive Controller on a VTOL UAV

Flight-test results of the concurrent-learning adaptive law described in Sect. 29.3 are presented in this section. The test vehicle is the GTMax rotorcraft UAV. The modification to the adaptive controller described in Sect. 29.2.2 include the concurrent-learning adaptive law of equations (Eq. 29.27) for a nonlinearly parameterized SHL NN. Data points were selected online based on Eq. 29.30 and were stored in a history stack limited to carrying 20 points. Once the history stack was full, a new data point was added by replacing the oldest data point. A fixed point smoother was used to estimate \dot{x}_j for a recorded data point using both forward and a backward Kalman filter (Chowdhary and Johnson 2011b; Gelb 1974). Typically this induced a selectable time delay introduced by the time required for the smoother to converge, however, this does not affect the instantaneous tracking error.

29.5.5.1 Repeated Forward-Step Maneuvers

The repeated forward-step maneuvers are chosen in order to create a relatively simple situation in which the controller performance can be compared over several similar maneuvers. By using concurrent-learning NN, improved performance is expected through repeated maneuvers and a faster convergence of weights. Figure 29.26 shows the body frame states from recorded flight data for a chain of forward-step inputs. Figure 29.27a, b shows the evolution of inner- and outer-loop errors. These results assert the stability (in the ultimate boundedness sense) of the combined concurrent and online learning approach.

Figure 29.28b, d show the evolution of NN W and V weights as the rotorcraft performs repeated step maneuvers and the NN is trained using the

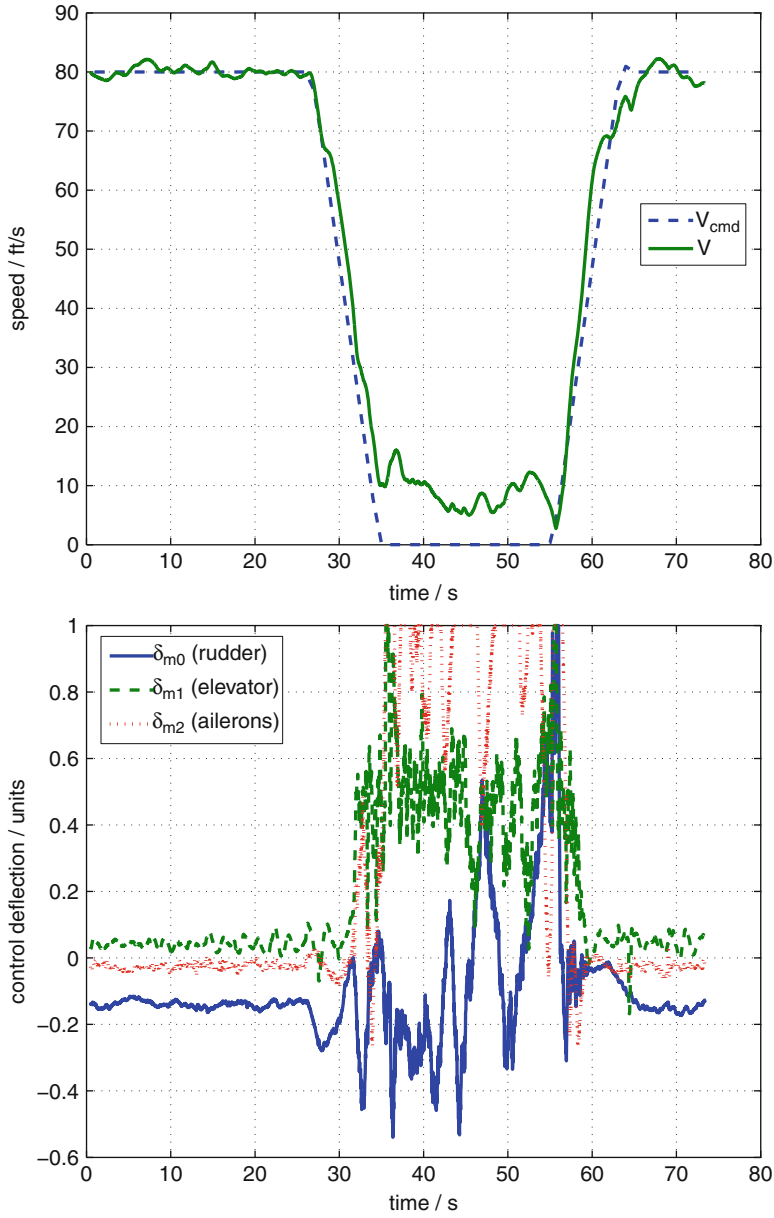


Fig. 29.22 GTEdge speed profile and control deflections during transitions between hover and forward flight

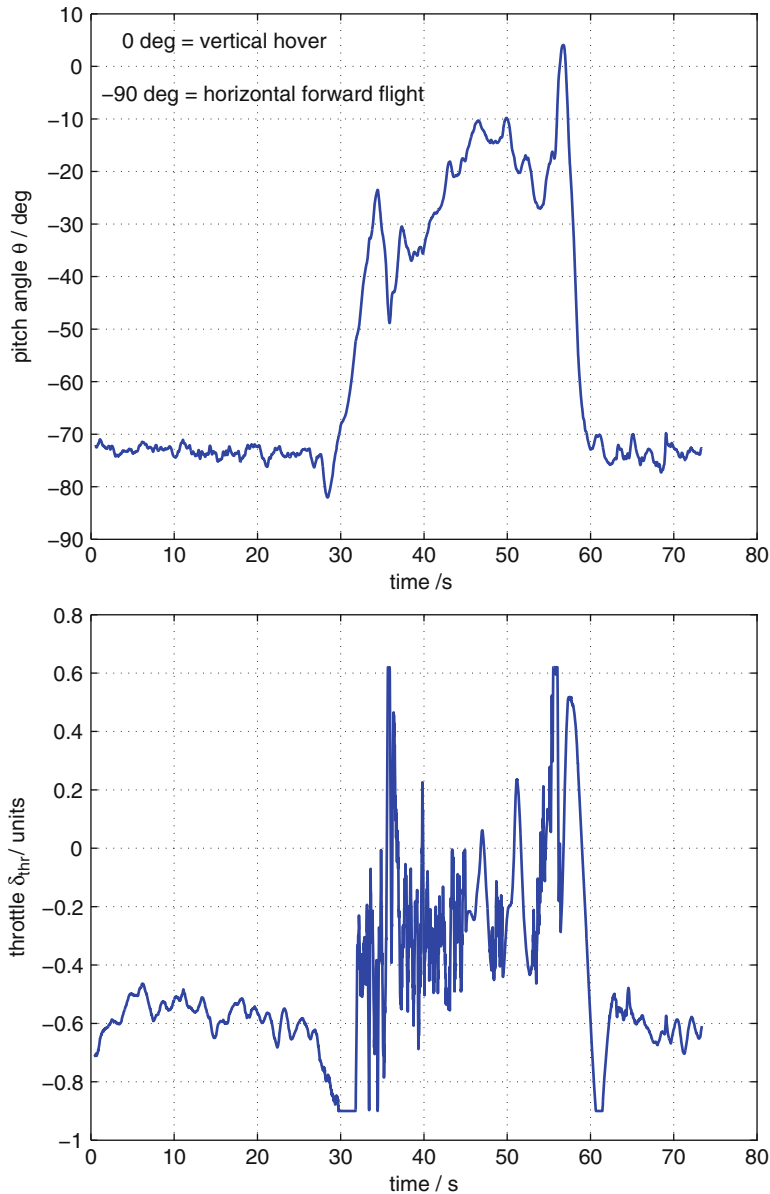


Fig. 29.23 GTEdge pitch angle, throttle profile during transitions between hover and forward flight

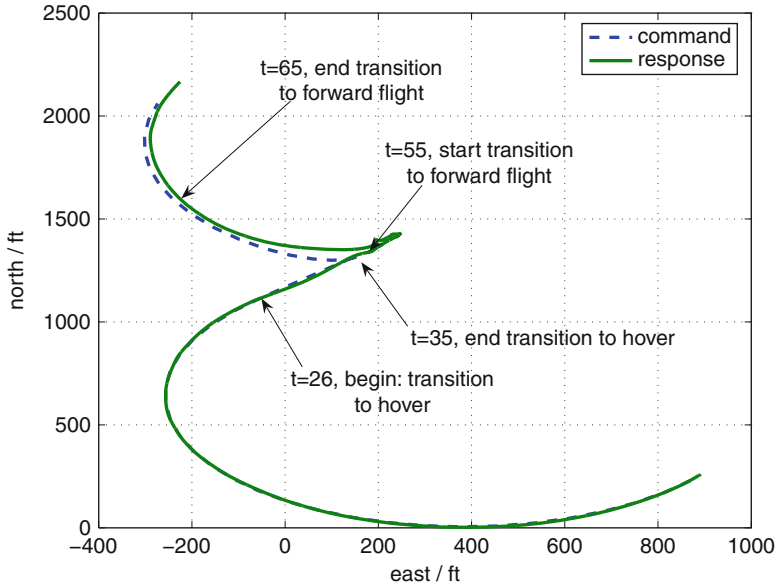


Fig. 29.24 GTEdge trajectory during transitions



Fig. 29.25 GTEdge during a transition

concurrent-learning method of Theorem 29.3. The NN V weights (Fig. 29.28b) appear to go to constant values when concurrent-learning adaptation is used, this can be contrasted with Fig. 29.28a which shows the V weight adaptation for a similar maneuver without concurrent-learning. NN W weights for both cases remain bounded, however it is seen that with concurrent learning adaptation the NN W weights seem to separate, this indicates alleviation of the rank-1 condition experienced by the baseline adaptive law relying only on instantaneous data (Chowdhary

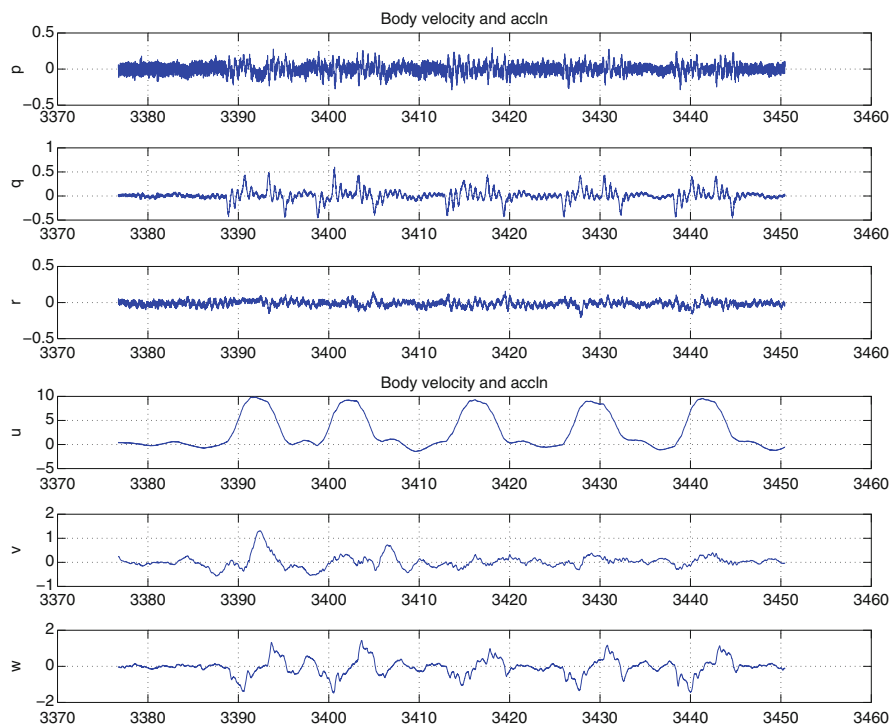


Fig. 29.26 Recorded body frame states for repeated forward steps

and Johnson 2011b). The flight test results indicate a noticeable improvement in the error profile. In Fig. 29.26 it is seen that the UAV tends not to have a smaller component of body lateral velocity (v) through each successive step. This is also seen in Fig. 29.27b where it is noted that the error in v (body y axis velocity) reduces through successive steps. These effects in combination indicate that the combined online and concurrent-learning system is able to improve performance over the baseline controller through repeated maneuvers, indicating long-term learning. These results are of particular interest, since the maneuvers performed were conservative, and the baseline adaptive MRAC controller had already been extensively tuned.

29.5.5.2 Aggressive Trajectory Tracking Maneuvers

Flight-test results are presented for concurrent-learning adaptive controllers while tracking repeatedly an elliptical trajectory with aggressive velocity (50 ft/s) and acceleration (20 ft/s^2) profile. Since these maneuvers involve state commands in more than one system state it is harder to visually inspect the data and see whether an improvement in performance is seen, therefore the Euclidian norm of

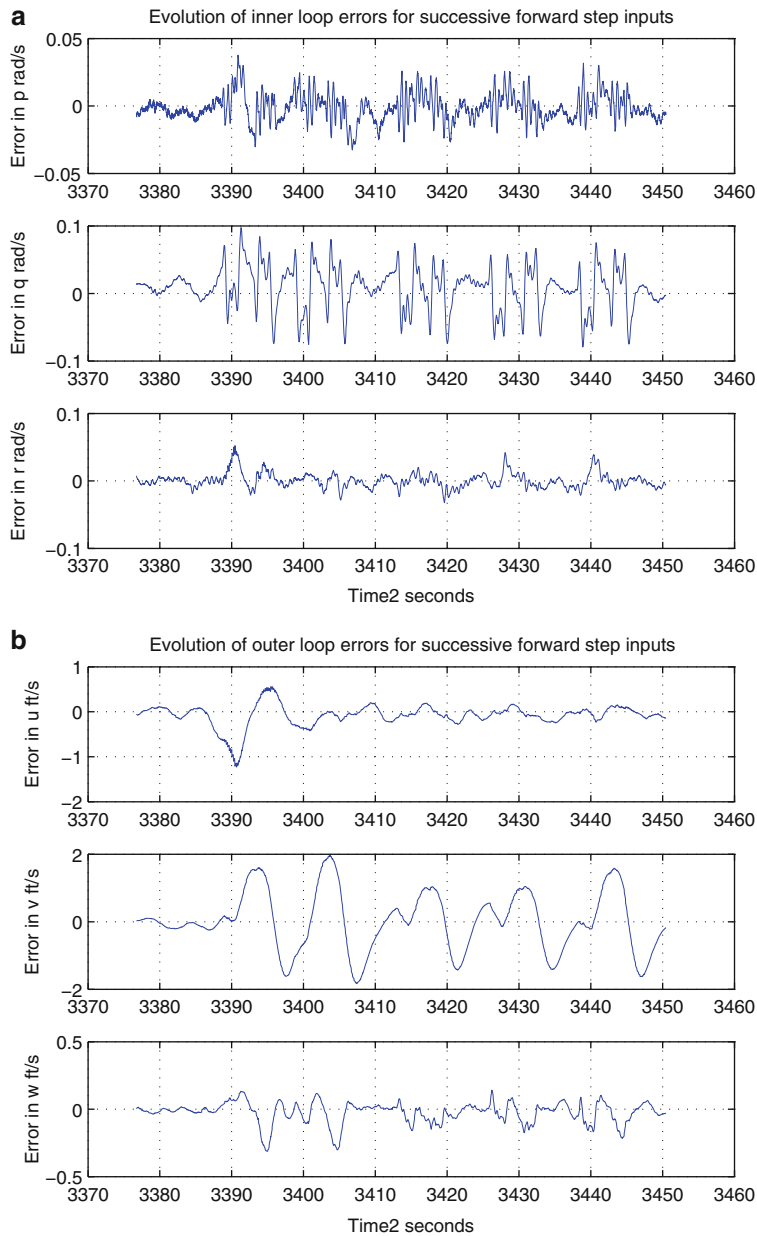


Fig. 29.27 GTMax recorded tracking errors for successive forward-step inputs with concurrent learning. **(a)** Evolution of inner loop errors with concurrent adaptation. **(b)** Evolution of outer loop errors with concurrent adaptation

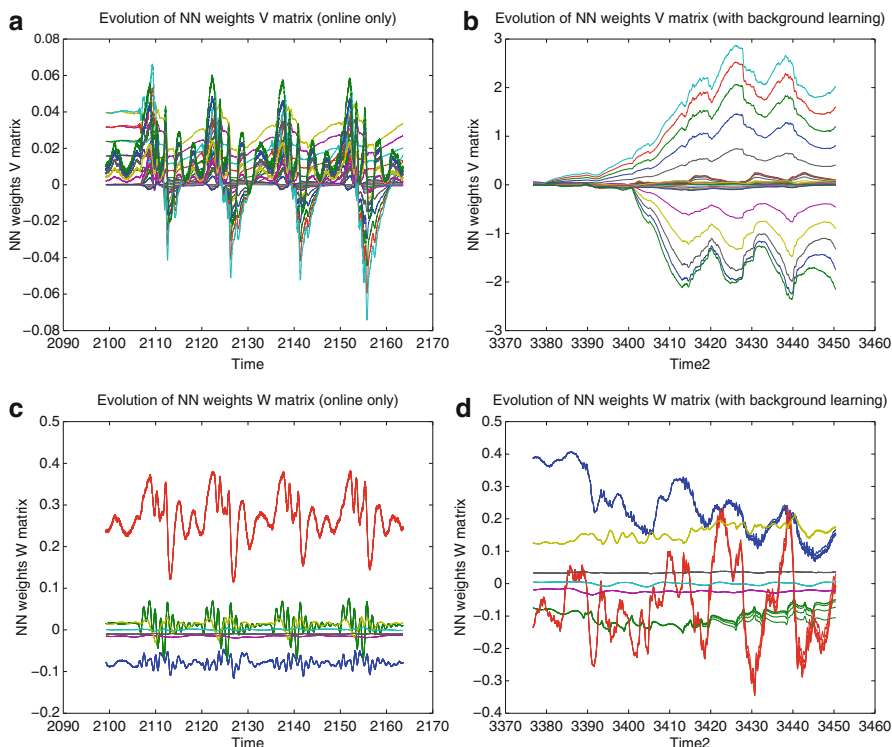


Fig. 29.28 Comparison of weight convergence on GTMax with and without concurrent learning. (a) Evolution of V matrix weights with only online adaptation. (b) Evolution of V matrix weights with concurrent adaptation. (c) Evolution of W matrix weights with only online adaptation. (d) Evolution of W matrix weights with concurrent adaptation

the error signal at each time step is used as a rudimentary metric. Figure 29.29 shows the recorded inner- and outer-loop states as the rotorcraft repeatedly tracks an oval trajectory pattern. In this flight, the first two ovals (until $t = 5415$ s) are tracked with a commanded acceleration of 30 ft/s^2 , while the rest of the ovals are tracked at 20 ft/s^2 . In the following both these parts of the flight test are discussed separately.

29.5.5.3 Aggressive Trajectory Tracking with Saturation in the Collective Channel

Due to the aggressive acceleration profile of 30 ft/s^2 the rotorcraft collective channels were observed to saturate while performing high velocity turns. This leads to an interesting challenge for the adaptive controller equipped with pseudocontrol hedging. Figure 29.30 shows the evolution of the innerloop and outerloop tracking error. It can be clearly seen that the tracking error in the u (body x axis velocity)

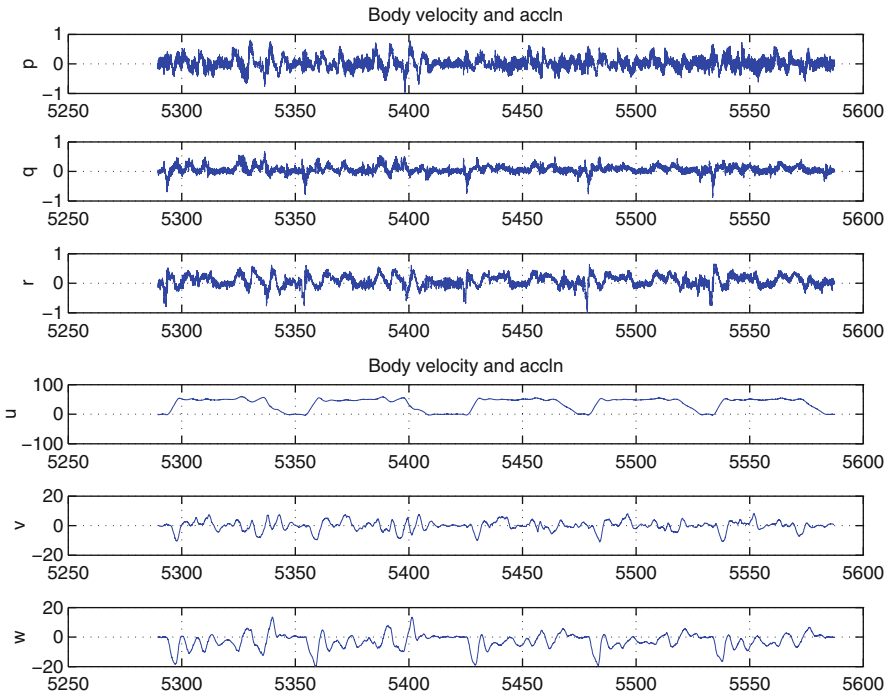


Fig. 29.29 Recorded body frame states for repeated oval maneuvers

channel reduces in the second pass through the ellipse indicating long-term learning by the combined online and concurrent-learning adaptive control system. This result is further characterized by the noticeable reduction in the norm of the tracking error at every time step as shown in Fig. 29.31.

29.5.5.4 Aggressive Trajectory Tracking Maneuver

For the results presented in this section, the acceleration profile was reduced to 20 ft/s^2 . At this acceleration profile, no saturation in the collective input was noted. Figure 29.32 shows the evolution of tracking error, and Fig. 29.33a shows the plot of the norm of the tracking error at each time step.

29.5.5.5 Aggressive Trajectory Tracking Maneuvers with Only Online Learning NN

The performance of the concurrent-learning adaptive controller is compared with the traditional instantaneous update-based adaptive controllers for the maneuvers described in Sect. 29.5.5.3.

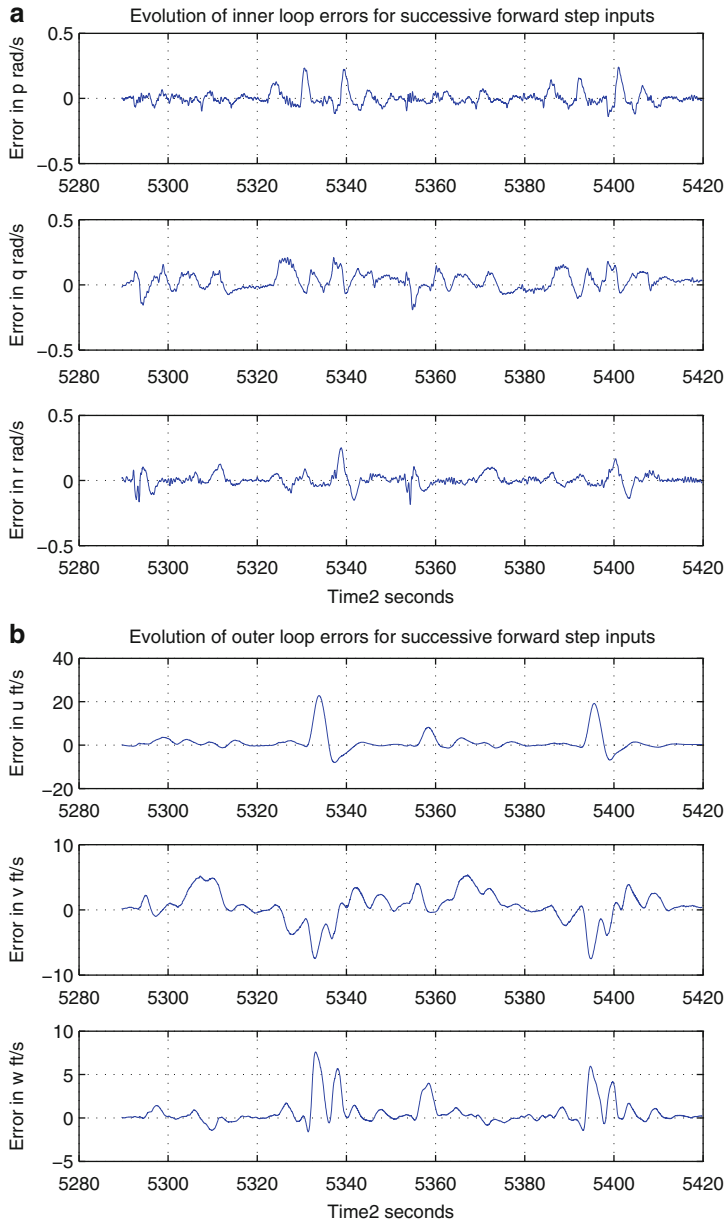
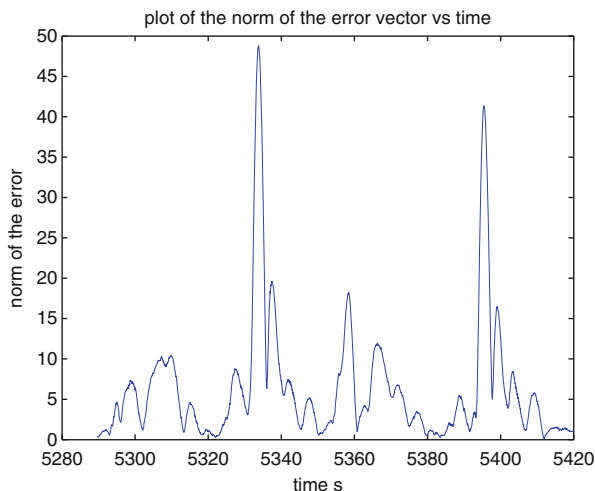


Fig. 29.30 GTMax recorded tracking errors for aggressive maneuvers with saturation in collective channels with concurrent learning. (a) Evolution of inner-loop errors with concurrent adaptation. (b) Evolution of outer-loop errors with concurrent adaptation

Fig. 29.31 Plot of the norm of the error at each time step for aggressive trajectory tracking with collective saturation



It is instructive to compare Fig. 29.34b, d which show the evolution of the NN weights with only instantaneous learning with Fig. 29.34a, c which show evolution of the NN weights with concurrent learning. Although absolute convergence of weights is not seen, as expected due to Theorem 29.3, it is interesting to see that when concurrent learning is on, the weights tend to be less oscillatory than when only instantaneous learning is used. Also, with combined online and concurrent learning, the weights do not tend to go to zero as the rotorcraft hovers between two successive tracking maneuvers. Figure 29.33b shows the plot of the tracking error norm as a function of time without concurrent learning. Comparing this figure with Fig. 29.33a, it can be clearly seen that the norm of the error vector is much higher when only online learning is used. This indicates that the combined online and concurrent-learning adaptive controller has improved trajectory tracking performance.

In summary, the flight test results ascertain an expected improvement in tracking performance. Furthermore, the evolution of the neural network W and V matrix weights were observed to have different characteristics when concurrent learning was employed, including weight separation, a tendency toward weight convergence in some cases, and different numerical values of the adaptive weights. This difference in neural network weight behavior demonstrates the effect of overcoming the rank 1 condition.

29.6 Conclusion

The objective in this chapter has been to provide an affordable control design solution that uses minimal prior knowledge of the vehicle dynamics. This is accomplished by relying on adaptation to cover the flight envelope of the helicopter

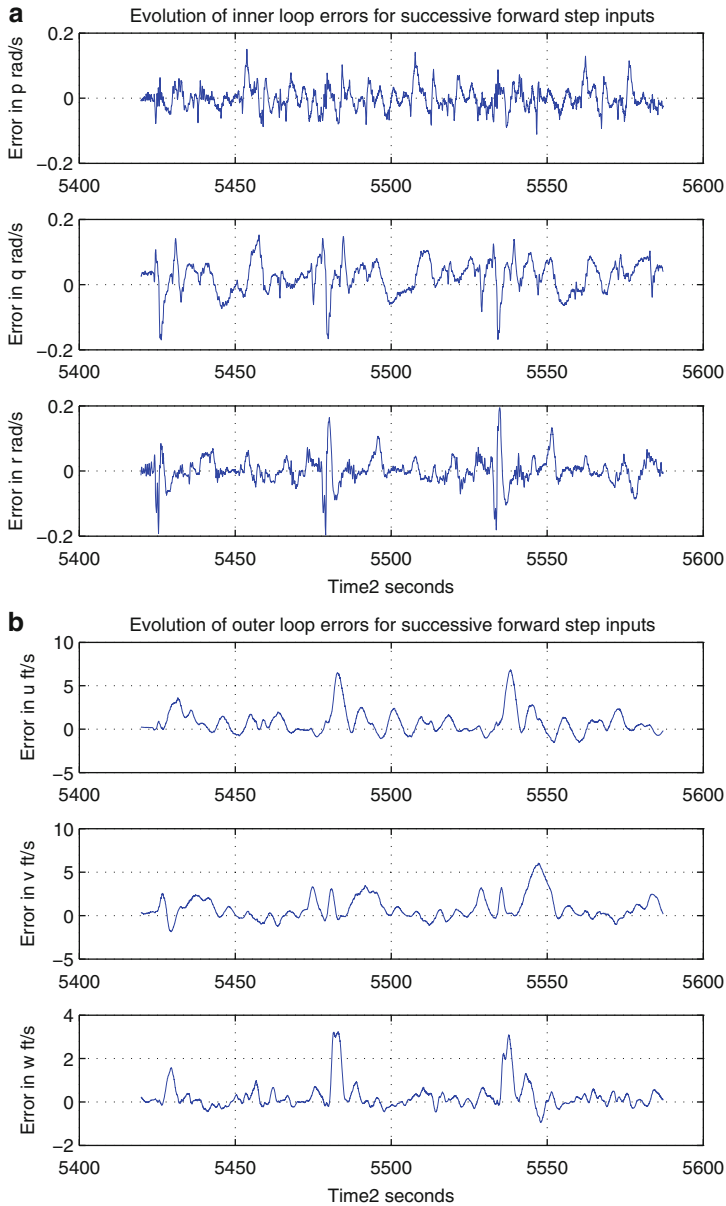


Fig. 29.32 GTMax recorded tracking errors for aggressive maneuvers with concurrent learning. (a) Evolution of inner-loop errors with concurrent adaptation. (b) Evolution of outer-loop errors with concurrent adaptation

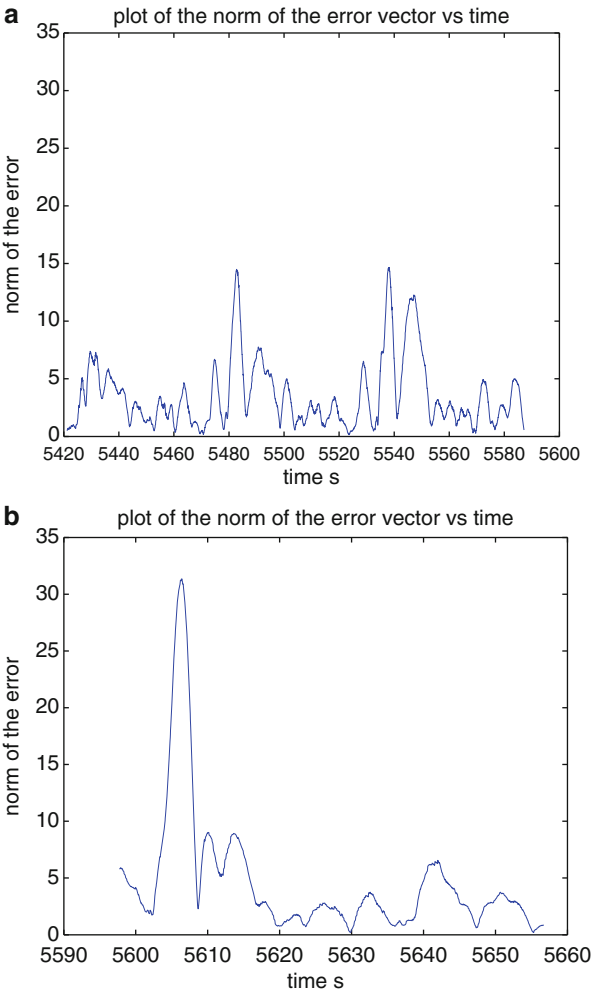


Fig. 29.33 Comparison of the norm of GTMax recorded tracking errors for aggressive maneuvers. **(a)** Evolution of the norm of the tracking error with concurrent adaptation. **(b)** Evolution of the norm of the tracking error with only online adaptation

under nominal conditions. Under mission-specific variations in the environment and system dynamics due to payload changes or damage, adaptation allows little or no human intervention after deployment. This approach is also in agreement with the DoD UAS Roadmap which subscribes to the following view on UAVs: “...affordability will be treated as a key performance parameter (KPP) equal to, if not more important than, schedule and technical performance...”.

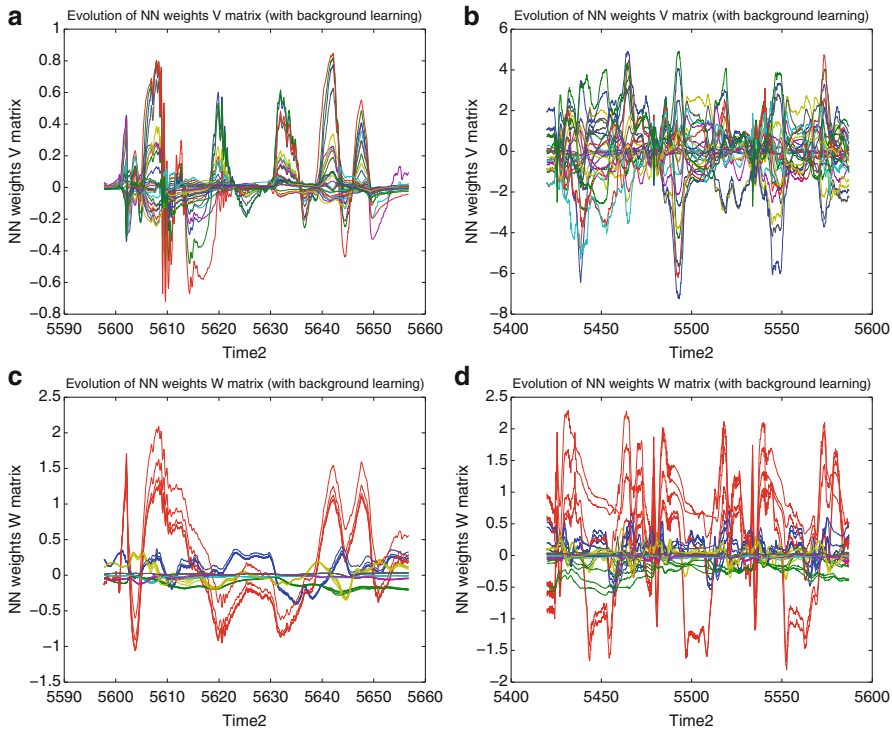


Fig. 29.34 Comparison of weight convergence as GTMax tracks aggressive trajectory with and without concurrent learning. (a) Evolution of V matrix weights with only online adaptation. (b) Evolution of V matrix weights with concurrent adaptation. (c) Evolution of W matrix weights with only online adaptation. (d) Evolution of W matrix weights with concurrent adaptation

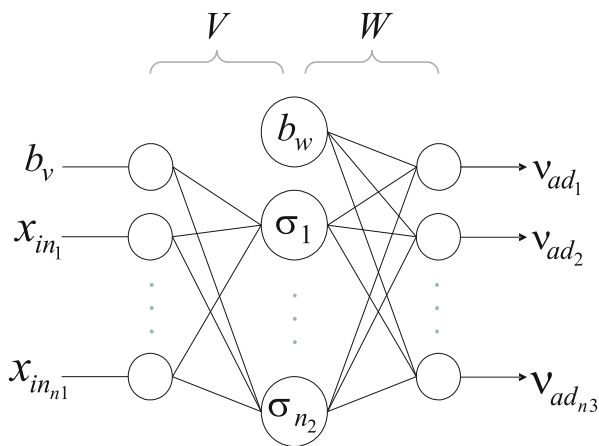


Fig. 29.35 Neural network with one hidden layer

A Adaptive Element

Single hidden layer (SHL) perceptron NNs are universal approximators (Hornik et al. 1989b; Spooner et al. 2002; Lewis 1999). Hence, given a sufficient number of hidden layer neurons and appropriate inputs, it is possible to train the network online to cancel model error.

Figure 29.35 shows the structure of a generic single hidden layer network whose input-output map may be expressed as

$$v_{ad_k} = b_w \theta_{w_k} + \sum_{j=1}^{n_2} w_{jk} \sigma_j(z_j), \quad (29.51)$$

where $k = 1, \dots, n_3$, b_w is the outer layer bias, θ_{w_k} is the k th threshold. w_{jk} represents the outer layer weights, z_j is the input to the neurons, and the scalar σ_j is a sigmoidal activation function

$$\sigma_j(z_j) = \frac{1}{1 + e^{-az_j}}, \quad (29.52)$$

where a is the so-called activation potential and may have a distinct value for each neuron. z_j is the input to the j th hidden layer neuron and is given by

$$z_j = b_v \theta_{v_j} + \sum_{i=1}^{n_1} v_{ij} x_{in_i}, \quad (29.53)$$

where b_v is the inner layer bias and θ_{v_j} is the j th threshold. Here, n_1, n_2 , and n_3 are the number of inputs, hidden layer neurons, and outputs, respectively. x_{in_i} , $i = 1, \dots, n_1$, denotes the inputs to the NN. For convenience, define the following weight matrices:

$$V \triangleq \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix}, \quad (29.54)$$

$$W \triangleq \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix}, \quad (29.55)$$

$$Z \triangleq \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix}. \quad (29.56)$$

Additionally, define the $\sigma(z)$ vector as

$$\sigma^T(z) \triangleq [b_w \sigma(z_1) \cdots \sigma(z_{n_2})], \quad (29.57)$$

where $b_w > 0$ allows for the thresholds, θ_w , to be included in the weight matrix W . Also, $z = V^T \bar{x}$, where

$$\bar{x}^T = [b_v x_{\text{in}}^T], \quad (29.58)$$

where $b_v > 0$ is an input bias that allows for thresholds θ_v to be included in the weight matrix V . The input-output map of the SHL network may now be written in concise form as

$$v_{ad} = W^T \sigma(V^T \bar{x}). \quad (29.59)$$

The NN may be used to approximate a nonlinear function, such as $\Delta(\cdot)$. The universal approximation property (Hornik et al. 1989b) of NNs ensures that given an $\bar{\epsilon} > 0$, then $\forall \bar{x} \in \mathcal{D}$, where \mathcal{D} is a compact set, \exists an \bar{n}_2 , and an ideal set of weights (V^*, W^*) brings the output of the NN to within an ϵ -neighborhood of the function approximation error. This ϵ is bounded by $\bar{\epsilon}$ which is defined by

$$\bar{\epsilon} = \sup_{\bar{x} \in \mathcal{D}} \|W^T \sigma(V^T \bar{x}) - \Delta(\bar{x})\|. \quad (29.60)$$

The weights (V^*, W^*) may be viewed as optimal values of (V, W) in the sense that they minimize $\bar{\epsilon}$ on \mathcal{D} . These values are not necessarily unique. The universal approximation property thus implies that if the NN inputs x_{in} are chosen to reflect the functional dependency of $\Delta(\cdot)$, then $\bar{\epsilon}$ may be made arbitrarily small given a sufficient number of hidden layer neurons, n_2 .

References

- Aeronautical Design Standard, *Handling Qualities Requirements for Military Rotorcraft, ADS-33E* (United States Army Aviation and Missile Command, Redstone Arsenal, AL, March 2000)
- A. Bemporad, A. Casavola, E. Mosca, Nonlinear control of constrained linear systems via predictive reference management. *IEEE Trans. Autom. Control* **42**(3), 340–349 (1997)
- D.S. Bernstein, A.N. Michel, A chronological bibliography on saturating actuators. *Int. J. Robust Nonlinear Control* **5**, 375–380 (1995)
- S. Boyd, S. Sastry, Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica* **22**(6), 629–639 (1986)
- A.J. Calise, N. Hovakimyan, M. Idan, Adaptive output feedback control of nonlinear systems using neural networks. *Automatica* **37**, 1201–1211 (2001)
- G. Chowdhary, Concurrent learning for convergence in adaptive control without persistency of excitation. Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, 2010
- G. Chowdhary, E.N. Johnson, Concurrent learning for convergence in adaptive control without persistency of excitation, in *49th IEEE Conference on Decision and Control*, Atlanta, 2010, pp. 3674–3679
- G. Chowdhary, E.N. Johnson, A singular value maximizing data recording algorithm for concurrent learning, in *American Control Conference*, San Francisco, June 2011a

- G. Chowdhary, E.N. Johnson, Theory and flight test validation of a concurrent learning adaptive controller. *J. Guid. Control Dyn.* **34**(2), 592–607 (2011b)
- G. Chowdhary, E.N. Johnson, S.M. Kimbrell, R. Chandramohan, A.J. Calise, Flight test results of adaptive controllers in the presence of significant aircraft faults, in *AIAA Guidance Navigation and Control Conference*, Toronto, Canada, 2010 Invited
- G. Chowdhary, R. Chandramohan, J. Hur, E.N. Johnson, A.J. Calise, Autonomous guidance and control of an airplane under severe damage, in *AIAA@INFOTECH*, St. Louis, MO, March 2011a. invited paper
- G. Chowdhary, M. Sobers, C. Pravitra, C. Christmann, A. Wu, H. Hashimoto, C. Ong, R. Kalghatgi, E.N. Johnson, Integrated guidance navigation and control for a fully autonomous indoor uas, in *AIAA Guidance Navigation and Control Conference*, Portland, OR, Aug 2011b
- G. Chowdhary, E.N. Johnson, R. Chandramohan, S.M. Kimbrell, A. Calise, Autonomous guidance and control of airplanes under actuator failures and severe structural damage. *J. Guid. Control Dyn.* (2012 in-press)
- H. Christophersen, R.W. Pickell, J.C. Neidhoefer, A.A. Koller, S.K. Kannan, E.N. Johnson, A compact guidance, navigation, and control system for unmanned aerial vehicles. *J. Aerosp. Comput. Inf. Commun.* **3**(5), 187–213 (2006)
- M. La Civita, W.C. Messner, T. Kanade, Modeling of small-scale helicopters with integrated first-principles and system-identification techniques, in *Proceedings of the 58th Forum of the American Helicopter Society*, Montreal, Canada, June 2002a, vol. 2, pp. 2505–2516
- M. La Civita, G. Papageorgiou, W.C. Messner, T. Kanade, Design and flight testing of a high bandwidth \mathcal{H}_∞ loop shaping controller for a robotic helicopter, in *AIAA Guidance, Navigation and Control Conference*, number AIAA-2002-4846, Monterey, CA, Aug 2002b
- M. La Civita, G. Papageorgiou, W.C. Messner, T. Kanade, Design and flight testing of a gain-scheduled \mathcal{H}_∞ loop shaping controller for wide-envelope flight of a robotic helicopter, in *Proceedings of the 2003 American Control Conference*, Denver, CO, June 2003, pp. 4195–4200
- J.E. Corban, A.J. Calise, J.V.R. Prasad, G. Heynen, B. Koenig, J. Hur, Flight evaluation of an adaptive velocity command system for unmanned helicopters, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, TX, Aug 2003
- B. Etkin, *Dynamics of Atmospheric Flight* (Wiley, New York, 1972)
- E. Frazzoli, M.A. Dahleh, E. Feron, Real-time motion planning for agile autonomous vehicles. *AIAA J. Guid. Control Dyn.* **25**(1), 116–129 (2002)
- V. Gavrillets, B. Mettler, E. Feron, Nonlinear model for a small-sized acrobatic helicopter, in *AIAA Guidance, Navigation and Control Conference*, number 2001-4333, Montréal, QC, Canada, Aug 2001
- V. Gavrillets, B. Mettler, E. Feron, Control logic for automated aerobatic flight of miniature helicopter, in *AIAA Guidance, Navigation and Control Conference*, number AIAA-2002-4834, Monterey, CA, Aug 2002
- A. Gelb, *Applied Optimal Estimation* (MIT, Cambridge, 1974)
- S. Haykin, *Neural Networks a Comprehensive Foundation*, 2nd edn. (Prentice Hall, Upper Saddle River, 1998)
- K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *IEEE Trans. Neural Netw.* **2**(5), 359–366 (1989a)
- K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**(5), 359–366 (1989b)
- E.N. Johnson, Limited authority adaptive flight control. Ph.D. thesis, Georgia Institute of Technology, 270 Ferst Drive, Atlanta GA 30332, USA, 2000
- E.N. Johnson, S.K. Kannan, Adaptive trajectory control for autonomous helicopters. *J. Guid. Control Dyn.* **28**(3), 524–538 (2005)
- S.K. Kannan, Adaptive control of systems in cascade with saturation. Ph.D. thesis, Georgia Institute of Technology, 270 Ferst Drive, Atlanta GA 30332, USA, Dec 2005
- S.K. Kannan, E.N. Johnson, Nested saturation with guaranteed real poles, in *American Control Conference*, Denver, CO, June 2003, pp. 497–502

- S.K. Kannan, E.N. Johnson, Adaptive control of systems in cascade with saturation, in *IEEE Conference on Decision and Control*, Atlanta, GA, Dec 2010a
- S.K. Kannan, E.N. Johnson, Model reference adaptive control with a constrained linear reference model, in *IEEE Conference on Decision and Control*, Atlanta, GA, Dec 2010b
- S.K. Kannan, A.A. Koller, E.N. Johnson, Simulation and development environment for multiple heterogeneous uavs, in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Providence, RI, Aug 2004
- S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **30**, 846–894 (2011)
- F. Kendoul, A survey of advances in guidance, navigation and control of unmanned rotorcraft systems. *J. Field Robot.* **29**(2), 215–378 (2012)
- F. Kendoul, I. Fantoni, R. Lozano, Modeling and control of a small autonomous aircraft having two tilting rotors. *IEEE Trans. Robot.* **22**(6), 1297–1302 (2006)
- F. Kendoul, L. David, I. Fantoni, R. Lozano, Real-time nonlinear embedded control for an autonomous quad-rotor helicopter. *AIAA J. Guid. Control Dyn.* **30**(4), 1049–1061 (2007)
- F. Kendoul, Z. Yu, K. Nonami, Guidance and nonlinear control system for autonomous flight of mini-rotorcraft unmanned aerial vehicles. *J. Field Robot.* **27**(3), 311–334 (2010)
- Y.H. Kim, F. Lewis, *High-Level Feedback Control with Neural Networks*. Robotics and Intelligent Systems, vol. 21 (World Scientific, Singapore, 1998)
- H.A. Kingravi, G. Chowdhary, P.A. Vela, E.N. Johnson, Reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(7), 1130–1141 (2012)
- F.L. Lewis, Nonlinear network structures for feedback control (survey paper). *Asian J. Control* **1**(4), 205–228 (1999)
- Y. Lin, E.D. Sontag, Universal formula for stabilization with bounded controls. *Syst. Control Lett.* **16**, 393–397 (1991)
- A.M. Lipp, J.V.R. Prasad, Synthesis of a helicopter nonlinear flight controller using approximate model inversion. *Math. Comput. Model.* **18**, 89–100 (1993)
- B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft* (Kluwer, Boston/London, 2002)
- C. Munzinger, Development of a real-time flight simulator for an experimental model helicopter. Master's thesis, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA 30332, USA, 1998
- G.J. Pappas, Avoiding saturation by trajectory reparameterization, in *IEEE Conference on Decision and Control*, Kobe, 1996
- J. Park, I. Sandberg, Universal approximation using radial-basis-function networks. *Neural Comput.* **3**, 246–257 (1991)
- D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. *Nature* **323**(6088), 533 (1986)
- R.T. Rysdyk, A.J. Calise, Robust nonlinear adaptive flight control for consistent handling qualities. *IEEE Trans. Control Syst. Technol.* **13**(6), 896–910 (2005). Robust nonlinear ad
- S.N. Singh, W. Yim, W.R. Wells, Direct adaptive control of wing rock motion of slender delta wings. *J. Guid. Control Dyn.* **18**(1), 25–30 (1995)
- J.T. Spooner, M. Maggiore, R. Ordóñez, K.M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques* (Wiley, New York, 2002)
- B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation* (Wiley, New York, 2003)
- J.A. Suykens, J.P. Vandewalle, B.L.D. Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems* (Kluwer, Norwell, 1996)
- A. Teel, A nonlinear small gain theorem for the analysis of control systems with saturation. *IEEE Trans. Autom. Control* **41**(9), 1256–1270 (1996)
- A. Teel, Semi-global stabilization of linear systems with position and rate-limited actuators. *Syst. Control Lett.* **30**, 1–11 (1997)
- Unmanned Aircraft Systems Roadmap 2011–2036, Technical report, Office of the Secretary of Defense, 2011

Eugene Lavretsky

Contents

| | | |
|--------|---|-----|
| 30.1 | Introduction | 676 |
| 30.2 | Aircraft Flight Dynamics Equations of Motion | 677 |
| 30.3 | Simplified Flight Dynamics for Control Design | 677 |
| 30.3.1 | Longitudinal Dynamics | 678 |
| 30.3.2 | Lateral-Directional Dynamics | 679 |
| 30.3.3 | Model Generalizations for Adaptive Control Design | 680 |
| 30.4 | Model Following LQR PI Command Tracking | 681 |
| 30.4.1 | Design Example: Lateral-Directional LQR PI Control with Explicit Model Following | 685 |
| 30.5 | Model Reference Adaptive Control | 688 |
| 30.5.1 | MRAC Design for MIMO Dynamics | 690 |
| 30.5.2 | MRAC Design Modifications for Robustness | 694 |
| 30.5.3 | Observer-Based MRAC Design with Transient Guarantees | 697 |
| 30.6 | Conclusion | 709 |
| | References | 709 |

Abstract

This chapter contains a collection of reliable, efficient, robust, and adaptive control methods for aerial vehicles. It begins with a brief overview of flight dynamics models suitable for flight control design. The first control design method represents the well-understood and now-classical linear quadratic regulator (LQR) command tracker, with proportional-integral (PI) feedback connections. Such a system is the backbone of all other methods presented in this chapter. The main intent here is to demonstrate the design of predictable, formally justified, yet numerically efficient flight controllers, with an LQR PI baseline and with a direct model reference adaptive control (MRAC), as an augmentation

E. Lavretsky

Boeing Senior Technical Fellow, The Boeing Company, Huntington Beach, CA, USA

e-mail: eugene.lavretsky@boeing.com

to the baseline. Through extensive simulation, analysis, and actual flight testing, it has been found that (LQR PI + adaptive) – controllers provide robust stability and maintain tracking performance, when operated in the presence of “unknown unknowns” in the vehicle dynamics and in often “unfriendly” operational environment. Finally, a note is in order: All control methods described in this chapter were successfully flight tested and validated on a wide range of aerial vehicles.

30.1 Introduction

Robust control is an online policy capable of regulating systems (such as aerial vehicles) with bounded uncertainties in their dynamics. A robust controller would work satisfactorily for a set of plants, whether linear or nonlinear, while assuming the worst case conditions on uncertainties in the vehicle dynamics. Excluding ad hoc designs, all reliable control methods are model based. One often starts with a mathematical model of the vehicle. Such a model would be valid in a preset domain. The model may or may not be accurate in capturing significant effects in the vehicle dynamics. In order to overcome modeling deficiencies, one should seek a robust solution, designed based on the model, yet capable of controlling the real vehicle, and not just the model. It would also be highly desirable to have a controller whose performance “gracefully degrades” in the presence of uncertainties so that it would not abruptly break down when unprecedented events occur.

Embedding robustness properties into a control solution must be treated as one of the main criteria in any control design. For example, achieving closed-loop stability and tracking performance, while providing adequate stability margins, are the main goals, especially when dealing with linear system approximations of real processes. Methods and techniques to achieve this goal will be provided. Once a robust control solution is found, its robustness properties can be further extended to cover a wider class of uncertainties in the system dynamics. Such a problem could be addressed within the framework of adaptive control. It may allow the designer to cope with unbounded state-dependent nonlinear uncertainties that may exist in the vehicle dynamics.

What is the difference between robust and adaptive controllers? A robust controller is designed to operate under the worst case condition assumption. Such a controller may use excessive actions to accomplish the goal. In contrast, an adaptive controller would try to perform an online estimation of the process uncertainty, and then produce a control input to anticipate, overcome, or minimize the undesirable deviations from the prescribed closed-loop plant behavior. In addition to their adaptive properties, these controllers can be constructed to “learn” or equivalently to remember prior events. In this context, the notion of learning refers to remembering certain patterns, and then acting based on prior knowledge or “memory.” For instance, a tracking error integrator in a feedback loop is a learning controller. It accumulates and integrates regulation errors based on previous and current data.

It will be shown that an adaptive controller represents a nonlinear extension of a linear feedback integrator. In other words, adaptive loops form their output by integrating nonlinear functions of the system tracking errors.

A discussion of whether adaptive controllers outperform robust systems or vice versa is of no merit. One can argue that it is rather a seamless combination of both controllers that works best, in the sense of maintaining closed-loop stability, enforcing robustness to uncertainties, and delivering target performance, all and all while operating in the presence of unanticipated events.

30.2 Aircraft Flight Dynamics Equations of Motion

Six-degrees-of-freedom (6-DoF) rigid aircraft equations of motion can be derived based on Newton's second law of motion. These dynamics are often expressed in the aircraft-fixed body axes coordinate system (Etkin 1982; McRuer et al. 1990; Stevens and Lewis 1992). In general, the 6-DoF equations of motion represent a continuous dynamical multi-input–multi-output system in the standard state-space form,

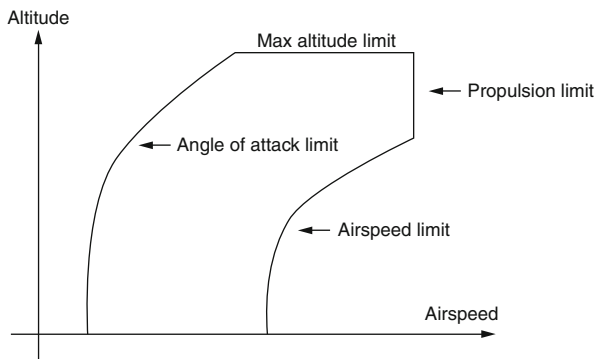
$$\dot{x} = f(x, u), \quad y = h(x, u) \quad (30.1)$$

with the state x , the control input u , and the measured/regulated output y . An attempt to use the fully coupled model (30.1) for control design would most likely result in an impractical control solution of unnecessary complexity and with a highly undesirable sensitivity due to model data. This phenomenon immediately presents a modeling-for-control challenge: How detailed does a control-oriented model need to be, so that the resulting control solution is simple, robust, effective, and works per design specifications, when applied to a real vehicle? The answer to this question of course depends on the application of interest. In the next section, simplified flight dynamics models for control design purposes will be constructed.

30.3 Simplified Flight Dynamics for Control Design

The 6-DoF motion of an aerial vehicle can be decomposed into a mean or a steady-state motion near an operating point (called “trim”) and perturbation dynamics around the trim conditions. Such a decomposition allows one to reduce the overall nonlinear fully coupled 6-DoF aircraft dynamics into a tractable form, suitable for control design and analysis. The notion of “trimming an aircraft” refers to finding a balance, or equilibrium, between aerodynamic, propulsive, and gravitational forces and moments that are constantly acting on the vehicle. In flight, an aircraft is trimmed by setting its primary controls to values that would result in the desired steady-state flight conditions. The trim function would be performed by a pilot or by an automatic flight control system.

Fig. 30.1 Aircraft operational flight envelope, as a function of altitude and airspeed



In mathematical terms, one is looking for a system equilibrium pair $(\vec{x}_{eq}, \vec{u}_{eq})$ in (30.1) such that the translational and angular accelerations are zeroed out:

$$0 = f(x_{eq}, u_{eq}) \quad (30.2)$$

These are steady-state flight conditions. Accelerated flight equilibrium is also possible.

An aircraft would have many distinct equilibrium points throughout the vehicle flight operational envelope (Fig. 30.1).

These trim points depend first hand on altitude and airspeed. Based on available trim flight conditions, the main idea behind constructing control-oriented models and then performing flight control design consists of several distinct steps. They are:

1. Cover the flight envelope with a dense set of trim points.
2. Write simplified linear models around each of the trim point.
3. Use these dynamics to design fixed-point flight controllers per point.
4. Interpolate (i.e., gain schedule based on flight conditions) to combine linear controllers.

The result is a gain-scheduled flight control system that would be valid for the entire operational envelope. In what follows, only Step 2 will be discussed, and linear simplified models (deviation dynamics from equilibrium) for a selected trim point will be defined.

When a conventional aircraft is trimmed wings level, at selected flight conditions, the vehicle dynamics naturally decouples into longitudinal and lateral-directional modes. Each of these modes is presented separately.

30.3.1 Longitudinal Dynamics

The aircraft longitudinal dynamics describe changes in forward, vertical, and pitching motion of the vehicle. These dynamics can be further decomposed into fast and slow components, or modes. The former is called the short-period and the

latter is the phugoid. Typically, there would be a time-scale separation between the two modes. The short-period describes fast coupling between the aircraft angle of attack and the pitch rate. The phugoid represents a much slower (when compared to the short-period) dynamic interchange between the vehicle altitude and airspeed, or, equivalently, between the aircraft potential and kinetic energy levels.

The short-period and the phugoid modes can be revealed after the aircraft model is linearized around a trim point (an equilibrium). For clarity of presentation, it is assumed that the thrust line is aligned with the vehicle x -axis. Then, the aircraft longitudinal equations of motion are

$$\begin{pmatrix} \dot{v}_T \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} X_V & X_\alpha & 0 & -g \cos \gamma_0 \\ \frac{Z_V}{V_0} & \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} & -\frac{g \sin \gamma_0}{V_0} \\ M_V & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_T \\ \alpha \\ q \\ \theta \end{pmatrix} + \begin{pmatrix} X_{\delta_{th}} \cos \alpha_0 & X_{\delta_e} \\ -X_{\delta_{th}} \sin \alpha_0 & \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_{th}} & M_{\delta_e} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix} \quad (30.3)$$

where V_0 is the trimmed airspeed and α_0 is trimmed angle of attack, $\gamma_0 = \theta_0 - \alpha_0$ is the trimmed flight path angle, θ_0 is the trimmed pitch angle, δ_{th} is the throttle position, and δ_e is the elevator position. The model states (v_T , α , q , θ) and the control inputs (δ_{th} , δ_e) are incremental due to their trimmed values. Also in (30.3), the matrix components represent constant (for fixed flight conditions) stability and control derivatives of the aircraft forces and moments, with respect to the longitudinal states and control inputs. When aircraft-specific values of these derivatives are substituted into model (30.3), most often the open-loop system eigenvalues will consist of a fast (short-period) and a slow (phugoid) pair of complex conjugate numbers. Such a decomposition explains the time-scale separation in the longitudinal dynamics of an aircraft.

The short-period mode is defined by the dynamics of α and q . Extracting those from model (30.3) yields

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} \\ M_\alpha & M_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_e} \end{pmatrix} \delta_e \quad (30.4)$$

These dynamics describe aircraft motion on a short interval of time, due to elevator input. In aerospace applications, the short-period system is utilized quite often to support the development of robust and adaptive control technologies.

30.3.2 Lateral-Directional Dynamics

Assuming constant thrust, airspeed, and angle of attack, lateral-directional dynamics of an aircraft can be derived by linearization of the 6-DoF system (30.2), around a selected trim point. The resulting dynamics are

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{\cos \gamma_0}{\cos \theta_0} & \frac{\sin \gamma_0}{\cos \theta_0} \\ \frac{g \cos \theta_0}{V_0} & \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ 0 & L_\beta & L_p & L_r \\ 0 & N_\beta & N_p & N_r \end{pmatrix} \begin{pmatrix} \varphi \\ \beta \\ p_s \\ r_s \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \frac{Y_{\delta_{\text{ail}}}}{V_0} & \frac{Y_{\delta_{\text{rud}}}}{V_0} \\ L_{\delta_{\text{ail}}} & L_{\delta_{\text{rud}}} \\ N_{\delta_{\text{ail}}} & N_{\delta_{\text{rud}}} \end{pmatrix} \begin{pmatrix} \delta_{\text{ail}} \\ \delta_{\text{rud}} \end{pmatrix} \quad (30.5)$$

When the airspeed is sufficiently high, the gravity term in (30.5) becomes negligible: $\frac{g \cos \theta_0}{V_0} \approx 0$. In this case, the bank dynamics can be eliminated:

$$\begin{pmatrix} \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix} = \begin{pmatrix} \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{pmatrix} \begin{pmatrix} \beta \\ p_s \\ r_s \end{pmatrix} + \begin{pmatrix} \frac{Y_{\delta_{\text{ail}}}}{V_0} & \frac{Y_{\delta_{\text{rud}}}}{V_0} \\ L_{\delta_{\text{ail}}} & L_{\delta_{\text{rud}}} \\ N_{\delta_{\text{ail}}} & N_{\delta_{\text{rud}}} \end{pmatrix} \begin{pmatrix} \delta_{\text{ail}} \\ \delta_{\text{rud}} \end{pmatrix} \quad (30.6)$$

The resulting third-order lateral-directional linear model would be suitable for a control design where the goal is to regulate the vehicle roll and yaw rates, as well as the angle of sideslip.

30.3.3 Model Generalizations for Adaptive Control Design

The aircraft short-period dynamics (30.4), as well as the lateral-directional models (30.5) and (30.6), belong to a class of linear time-invariant controllable systems in the form

$$\dot{x} = Ax + Bu \quad (30.7)$$

with the n -dimensional state x , the m -dimensional control u , the p -dimensional output:

$$y = Cx + Du \quad (30.8)$$

and with the matrices (A, B, C, D) of the corresponding dimensions.

In the next section, methods to construct and analyze robust linear LQR-optimal controllers are presented. The focus is on adaptive control techniques, with the goal of maintaining closed-loop stability and robustness in the presence of unexpected events. Specifically, inserting uncertainties into (30.7) and (30.8) results in the dynamical system

$$\dot{x} = Ax + B \Lambda (u + f(x)) \quad (30.9)$$

where the $(m \times m)$ -matrix Λ represents control actuation failures and the m -dimensional state-dependent vector function $f(x)$ denotes all other “unknown unknowns” in the system dynamics. The uncertain model (30.9) constitutes an attempt to embed extra realism into the “ideal” system (30.7). The uncertainties in (30.9) are called “matched,” in the sense that they enter the system dynamics through control channels. So, as long as Λ is invertible, the system controllability property is preserved. It so happens that the matched uncertainty assumption implies

existence of at least one control solution, capable of steering the system state along the desired trajectories.

Of interest are command tracking problems with nonmatched but bounded uncertainties, such as time-dependent noise and environmental disturbances, represented by an n -dimensional uniformly bounded piece-wise continuous vector function $\xi(t)$:

$$\dot{x} = Ax + B\Lambda(u + f(x)) + \xi(t) \quad (30.10)$$

Care must be taken in specifying conditions on $\xi(t)$ that preserve controllability of the system. So as long as the system remains controllable, the unwanted effects caused by bounded noise and disturbances can be mitigated through proper control synthesis. The forthcoming sections explore robust and adaptive methods to control uncertain systems, such as (30.9) and (30.10).

Readers may find the matched uncertainty assumption to be restrictive. Some may even argue that there are many dynamical systems, stemming from realistic applications, that do not satisfy the matching conditions. This is a true statement indeed. However, in aerospace applications, matched uncertainties are of primary concern. They represent unknown aerodynamic and thrust effects that may exist in the vehicle moments. These uncertainties are “matched” by available controls (such as elevator, aileron, rudder, etc.), in the sense that the moments explicitly depend on the control inputs, and the latter can be selected to counter (i.e., cancel or dominate) the matched uncertainties. In essence, moment regulation for an aircraft implies total control of the vehicle, and uncertain dynamical systems in the form of (30.10) can be utilized to achieve that goal. Finally, it is worth noting that the adaptive control methods presented in this chapter can be extended to handle systems with nonlinear-in-control matched uncertainties and with nonmatched uncertain dynamics, but these extensions are outside of the present scope.

30.4 Model Following LQR PI Command Tracking

For a linear time-invariant (LTI) system, such as (30.7), a model following command tracking control design can be defined to represent an LQR-optimal controller, with PI feedback connections. Overall, this design method generalizes the well-known servomechanism approach to a model following design.

Consider an LTI system in the form

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_p u \\ y_p &= C_p x_p + D_p u \end{aligned} \quad (30.11)$$

where $x_p \in R^{n_p}$ is the n_p -dimensional state vector, $u \in R^{m_p}$ is the m_p -dimensional vector of controls, $y_p \in R^{p_p}$ is the system p_p -dimensional vector of regulated outputs with $p_p \leq m_p$, and plant matrices (A_p, B_p, C_p, D_p) are of the corresponding dimensions. Moreover, it is assumed that matrix pair (A_p, B_p) is

stabilizable and the entire state vector x_p is available online as the system output measurement.

The control task of interest is command tracking, that is, one needs to find u such that the regulated output y_p tracks its target $y_{\text{ref}} \in \mathbb{R}^{p_p}$, while all other signals in the system remain bounded.

Desired dynamics of the target output y_{ref} is defined via the reference model:

$$\begin{aligned}\dot{x}_{\text{ref}} &= A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} r(t) \\ y_{\text{ref}} &= C_{\text{ref}} x_{\text{ref}} + D_{\text{ref}} r(t)\end{aligned}\quad (30.12)$$

where $x_{\text{ref}} \in \mathbb{R}^{n_{\text{ref}}}$ is the state of the reference dynamics, $r(t)$ represents a bounded external command, and matrices $(A_{\text{ref}}, B_{\text{ref}}, C_{\text{ref}}, D_{\text{ref}})$ are of the corresponding dimensions, with A_{ref} being Hurwitz. Furthermore, it is assumed that the system DC gain matrix is unity, that is

$$DC_{\text{gain}}^{\text{ref}} = -C_{\text{ref}} A_{\text{ref}}^{-1} B_{\text{ref}} + D_{\text{ref}} = I_{m_p \times m_p} \quad (30.13)$$

Note that the dimension of the reference model $n_{\text{ref}} \geq p_p$, and it does not have to be the same as the dimension of the plant n_p .

Extended open-loop dynamics is formed as a combination of the plant and the reference model:

$$\underbrace{\begin{pmatrix} \dot{x}_p \\ \dot{x}_{\text{ref}} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} A_p & 0_{n_p \times n_{\text{ref}}} \\ 0_{n_{\text{ref}} \times n_p} & A_{\text{ref}} \end{pmatrix}}_A \underbrace{\begin{pmatrix} x_p \\ x_{\text{ref}} \end{pmatrix}}_x + \underbrace{\begin{pmatrix} B_p \\ 0_{n_{\text{ref}} \times m_p} \end{pmatrix}}_B u + \underbrace{\begin{pmatrix} 0_{n_p \times m_p} \\ B_{\text{ref}} \end{pmatrix}}_{B_r} r(t) \quad (30.14)$$

Its regulated output can be written as

$$y = y_p - y_{\text{ref}} = C_p x_p + D_p u - C_{\text{ref}} x_{\text{ref}} - D_{\text{ref}} r(t) \quad (30.15)$$

or, equivalently in matrix form as

$$y = \underbrace{(C_p - C_{\text{ref}})}_C \underbrace{\begin{pmatrix} x_p \\ x_{\text{ref}} \end{pmatrix}}_x + \underbrace{D_p}_D u + \underbrace{(-D_{\text{ref}})}_{D_r} r(t) \quad (30.16)$$

Combining (30.14) and (30.16) yields extended open-loop dynamics in matrix form:

$$\begin{aligned}\dot{x} &= A x + B u + B_r r(t) \\ y &= C x + D u + D_r r(t)\end{aligned}\quad (30.17)$$

Note the explicit presence of the command $r(t)$ in the plant formulation (30.17). This is one way to embed a reference model tracking into the control problem

formulation. Once again, one must choose the reference model data such that the extended system (30.17) remains controllable.

In terms of (30.17), the control task consists of finding u such that the system output y asymptotically tends to the origin. In other words, the control design task is output stabilization in the presence of any known, bounded, and possibly time-varying external command signal $r(t)$.

In order to track a step-input command with zero errors (type-1 response), integral control will be employed. A convenient technique for designing a practical tracker is the command-generator tracker (CGT) method, where the tracking problem is converted into a regulator problem (Franklin et al. 1986). Toward that end, integrated tracking error vector e_{yI} is introduced into the system dynamics (30.17):

$$\begin{aligned}\dot{e}_{yI} &= y \\ \dot{x} &= Ax + Bu + B_r r \\ y &= Cx + Du + D_r r\end{aligned}\tag{30.18}$$

Rewriting the dynamics in matrix form gives

$$\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x} \end{pmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{pmatrix} 0_{p_p \times p_p} & C \\ 0_{n_p \times p_p} & A \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} e_{yI} \\ x \end{pmatrix}}_{\tilde{x}} + \underbrace{\begin{pmatrix} D_p \\ B \end{pmatrix}}_{\tilde{B}} u + \underbrace{\begin{pmatrix} D_r \\ B_r \end{pmatrix}}_{\tilde{B}_r} r(t)\tag{30.19}$$

with the regulated output defined as

$$y = \underbrace{(0_{p_p \times p_p} \ C)}_{\tilde{C}} \underbrace{\begin{pmatrix} e_{yI} \\ x \end{pmatrix}}_{\tilde{x}} + \underbrace{D}_{\tilde{D}} u + \underbrace{D_r}_{\tilde{D}_r} r(t)\tag{30.20}$$

Assuming that the matrix pair (\tilde{A}, \tilde{B}) is controllable, control design for the open-loop dynamics in (30.19) can now be performed using LQR servomechanism approach. Toward that end, assume constant external command $r(t) = r$, utilize matrices as defined in (30.19) and (30.20):

$$\tilde{A} = \begin{pmatrix} 0_{p_p \times p_p} & C \\ 0_{n_p \times p_p} & A \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} D \\ B \end{pmatrix}, \quad \tilde{D} = D, \quad \tilde{D}_r = D_r\tag{30.21}$$

and consider stabilization of

$$\dot{z} = \tilde{A}z + \tilde{B}v\tag{30.22}$$

where

$$z = \begin{pmatrix} \dot{e}_{yI} \\ \dot{x} \end{pmatrix}, \quad v = \dot{u}\tag{30.23}$$

It is easy to see that dynamics (30.22) are obtained by differentiating (30.19) while assuming constant external command $r(t) = r$. Next, the newly introduced control

input v in (30.22) is chosen to minimize the LQR performance index (cost):

$$J = \int_0^{\infty} (z^T Q z + v^T R v) dt \quad (30.24)$$

where $Q = Q^T \geq 0$ and $R = R^T > 0$ are the LQR weight matrices. Then, the algebraic Riccati equation

$$\tilde{A}^T P + P \tilde{A} + Q - P \tilde{B} R^{-1} \tilde{B}^T P = 0 \quad (30.25)$$

has a unique positive definite symmetric solution $P = P^T > 0$. Based on the latter, the corresponding LQR-optimal control strategy is given in feedback form:

$$\dot{u} = v = - \underbrace{R^{-1} \tilde{B}^T P}_K z = -K z = - \begin{pmatrix} K_I & K_P \end{pmatrix} \begin{pmatrix} \dot{e}_y \\ x \end{pmatrix} \quad (30.26)$$

Integrating both sides and ignoring constants of integration yields the LQR-optimal control solution in PI feedback form:

$$u = -K_I e_{yI} - K_P x = K_I \frac{(y_{\text{ref}} - y)}{s} - K_P x = \boxed{K_I \frac{(y_{\text{ref}} - y)}{s} - K_P^x x - K_P^{x_{\text{ref}}} x_{\text{ref}}} \quad (30.27)$$

where proportional gains K_P^x and $K_P^{x_{\text{ref}}}$ are defined below

$$K_P = \begin{pmatrix} K_P^x & K_P^{x_{\text{ref}}} \end{pmatrix} \quad (30.28)$$

Using LQR feedback (30.27) to control plant dynamics (30.11) will force the system-regulated output y_p to track its command target y_{ref} , which in turn tracks a step-input external command r with zero steady-state error. As a result, the system output y_p will converge to r exponentially fast (for a step input and in steady state). The model following command tracking problem is solved. Figure 30.2 shows the corresponding closed-loop system block-diagram.

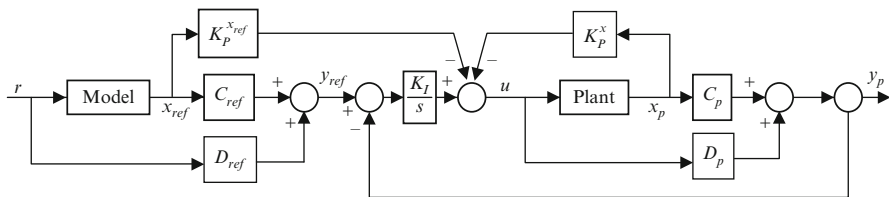
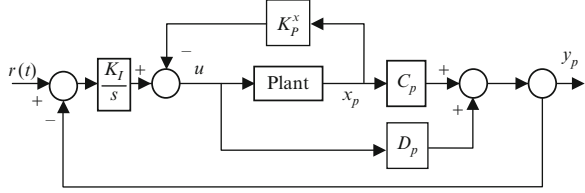


Fig. 30.2 Model following LQR PI control block-diagram

Fig. 30.3 Servomechanism
LQR PI control
block-diagram



The plant and the model shown in the figure are defined by (30.11) and (30.12), respectively. The reference model dynamics in (30.12) should not be confused with the assumed dynamics of the external command r . Most often, the assumed dynamics is defined as $\dot{r} = 0$, that is, the system is required to track step inputs with zero error. On the other hand, the reference model can be viewed as the command prefilter, that is the system will track the output of the reference model rather than the command itself. Thus, the main purpose of the reference model is to smooth out the command r before it goes into the system, and the model following state feedback LQR PI controller is designed to do precisely that.

If $(A_{\text{ref}}, B_{\text{ref}}, C_{\text{ref}})$ are set to zero and D_{ref} is the identity matrix, then according to (30.12), $y_{\text{ref}} = r(t)$ and the model following architecture from Fig. 30.2 reduces to the standard servomechanism control block-diagram (Fig. 30.3).

Note that in this case, the extended system remains controllable as long as

$$\text{rank} \begin{pmatrix} A_p & B_p \\ C_p & D_p \end{pmatrix} = n_p + m$$

which places restrictions on the regulated output selection.

30.4.1 Design Example: Lateral-Directional LQR PI Control with Explicit Model Following

In order to gain insights into the design, consider the aircraft lateral-directional dynamics (30.6), with the regulated output, whose components are the vehicle roll rate p_s and the angle of sideslip (AOS) β :

$$\underbrace{\begin{pmatrix} \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} \frac{Y_{\beta}}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ L_{\beta} & L_p & L_r \\ N_{\beta} & N_p & N_r \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \beta \\ p_s \\ r_s \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} \frac{Y_{\delta_a}}{V_0} & \frac{Y_{\delta_r}}{V_0} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{pmatrix}}_{B_p} \underbrace{\begin{pmatrix} \delta_{\text{ail}} \\ \delta_{\text{rud}} \end{pmatrix}}_u \quad (30.29)$$

$$y_p = \begin{pmatrix} p_s \\ \beta \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}}_{C_p} x_p + \underbrace{0_{2 \times 2}}_{D_p} u$$

where (p_s, r_s) are the aircraft roll and yaw angular rates in stability axes, $(\delta_{ail}, \delta_{rud})$ are the control inputs (aileron and rudder deflections), and V_0 is the aircraft true airspeed (trimmed). The rest of the parameters in the model represent the vehicle stability and control derivatives. In (30.29), $n_p = 3$, $m_p = p_p = 2$.

The control task is to construct aileron δ_{ail} and rudder δ_{rud} such that the aircraft roll rate p_s and AOS β track their step-input commands, p_{cmd} and β_{cmd} , respectively, while all other signals in the system remain bounded.

Desired roll rate and AOS dynamics are defined by the LTI reference model:

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{p}_{ref} \\ \dot{\beta}_{ref} \end{pmatrix}}_{\dot{x}_{ref}} &= \underbrace{\begin{pmatrix} -\frac{1}{\tau_p} & 0 \\ 0 & -\frac{1}{\tau_\beta} \end{pmatrix}}_{A_{ref}} \underbrace{\begin{pmatrix} p_{ref} \\ \beta_{ref} \end{pmatrix}}_{x_{ref}} + \underbrace{\begin{pmatrix} \frac{1}{\tau_p} & 0 \\ 0 & \frac{1}{\tau_\beta} \end{pmatrix}}_{B_{ref}} \underbrace{\begin{pmatrix} p_{cmd} \\ \beta_{cmd} \end{pmatrix}}_{r(t)} \\ y_{ref} &= \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{C_{ref}} x_{ref} + \underbrace{0_{2 \times 2}}_{D_{ref}} r(t) \end{aligned} \quad (30.30)$$

where τ_p and τ_β are small positive time constants that define the desired roll rate and the AOS dynamics, correspondingly. It is easy to check that the DC gain of the reference model is unity.

From (30.29) and (30.30) it follows that

$$n_p = 3, \quad p_p = m_p = 2, \quad n_{ref} = 2 \quad (30.31)$$

Using (30.14) gives extended open-loop dynamics:

$$\underbrace{\begin{pmatrix} \dot{x}_p \\ \dot{x}_{ref} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} A_p & 0_{3 \times 2} \\ 0_{2 \times 3} & A_{ref} \end{pmatrix}}_A \underbrace{\begin{pmatrix} x_p \\ x_{ref} \end{pmatrix}}_x + \underbrace{\begin{pmatrix} B_p \\ 0_{2 \times 2} \end{pmatrix}}_B \underbrace{\begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix}}_u + \underbrace{\begin{pmatrix} 0_{3 \times 2} \\ B_{ref} \end{pmatrix}}_{B_r} \underbrace{\begin{pmatrix} p_{cmd} \\ \beta_{cmd} \end{pmatrix}}_{r(t)} \quad (30.32)$$

where matrix pairs (A_p, B_p) and (A_{ref}, B_{ref}) are as in (30.29) and (30.30). According to (30.15), the system-regulated output is

$$y = y_p - y_{ref} = C_p x_p - C_{ref} x_{ref} = \underbrace{(C_p - C_{ref})}_C x + \underbrace{0_{2 \times 2}}_D u + \underbrace{0_{2 \times 2}}_{D_r} r(t) \quad (30.33)$$

Per (30.18), the extended dynamics (30.32) are augmented with the roll rate and AOS integrated tracking errors:

$$\begin{aligned} \dot{e}_{pI} &= p - p_{ref} \\ \dot{e}_{\beta I} &= \beta - \beta_{ref} \\ \dot{x} &= A x + B u + B_r r(t) \\ y &= C x \end{aligned} \quad (30.34)$$

The integrated tracking error vector e_{yI} has two components: integrated roll rate error e_{pI} and integrated AOS error $e_{\beta I}$. Similar to (30.19), one gets

$$\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x} \end{pmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{pmatrix} 0_{2 \times 2} & C \\ 0_{5 \times 2} & A \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} e_{yI} \\ x \end{pmatrix}}_{\tilde{x}} + \underbrace{\begin{pmatrix} 0_{2 \times 2} \\ B \end{pmatrix}}_{\tilde{B}} u + \underbrace{\begin{pmatrix} 0_{2 \times 2} \\ B_r \end{pmatrix}}_{\tilde{B}_r} r(t) \quad (30.35)$$

with the regulated output as in (30.20).

$$y = \begin{pmatrix} p - p_{\text{ref}} \\ \beta - \beta_{\text{ref}} \end{pmatrix} = \underbrace{\begin{pmatrix} 0_{2 \times 2} & C \end{pmatrix}}_{\tilde{C}} \tilde{x} + \underbrace{0_{m_p \times m_p}}_{\tilde{D}} u + \underbrace{0_{m_p \times m_p}}_{\tilde{D}_r} r(t) \quad (30.36)$$

Using matrices (\tilde{A}, \tilde{B}) from (30.35), choosing LQR weights (Q, R) appropriately, and solving the algebraic Riccati equation (30.25), yields the LQR PI matrix of optimal gains (see (30.26)):

$$K = R^{-1} \tilde{B}^T P \quad (30.37)$$

Partitioning the gains as in (30.28), the LQR-optimal control solution can be written in the form of (30.27):

$$u = K_I \frac{(y_{\text{ref}} - y)}{s} - K_P^x x - K_P^{x_{\text{ref}}} x_{\text{ref}} \quad (30.38)$$

where

$$K = \begin{pmatrix} K_I & K_P^x & K_P^{x_{\text{ref}}} \end{pmatrix} \quad (30.39)$$

In terms of the original notation, the model following LQR PI controller can be expressed as

$$\begin{cases} \delta_{\text{ail}} = K_{pI}^{\text{ail}} \frac{(p_{\text{ref}} - p)}{s} + K_{\beta I}^{\text{ail}} \frac{(\beta_{\text{ref}} - \beta)}{s} - K_{\beta}^{\text{ail}} \beta - K_p^{\text{ail}} p_s - K_r^{\text{ail}} r_s - K_{p_{\text{ref}}}^{\text{ail}} p_{\text{ref}} - K_{\beta_{\text{ref}}}^{\text{ail}} \beta_{\text{ref}} \\ \delta_{\text{rud}} = K_{pI}^{\text{rud}} \frac{(p_{\text{ref}} - p)}{s} + K_{\beta I}^{\text{rud}} \frac{(\beta_{\text{ref}} - \beta)}{s} - K_{\beta}^{\text{rud}} \beta - K_p^{\text{rud}} p_s - K_r^{\text{rud}} r_s - K_{p_{\text{ref}}}^{\text{rud}} p_{\text{ref}} - K_{\beta_{\text{ref}}}^{\text{rud}} \beta_{\text{ref}} \end{cases} \quad (30.40)$$

Comparing (30.39) and (30.40), yields the LQR PI optimal matrix gains:

$$K_I = \begin{pmatrix} K_{pI}^{\text{ail}} & K_{\beta I}^{\text{ail}} \\ K_{pI}^{\text{rud}} & K_{\beta I}^{\text{rud}} \end{pmatrix}, \quad K_P^x = \begin{pmatrix} K_{\beta}^{\text{ail}} & K_p^{\text{ail}} & K_r^{\text{ail}} \\ K_{\beta}^{\text{rud}} & K_p^{\text{rud}} & K_r^{\text{rud}} \end{pmatrix}, \quad K_P^{x_{\text{ref}}} = \begin{pmatrix} K_{p_{\text{ref}}}^{\text{ail}} & K_{\beta_{\text{ref}}}^{\text{ail}} \\ K_{p_{\text{ref}}}^{\text{rud}} & K_{\beta_{\text{ref}}}^{\text{rud}} \end{pmatrix} \quad (30.41)$$

The reader is encouraged to test this technique on a specific system of choice. Although the controller was designed to track step-input commands, its command following performance will be excellent for any bounded command, whose rate of change is within the closed-loop system bandwidth. In addition, one can and should

compute stability margins, at the plant input and output break points. These margins will be excessive, which is to be expected for any LQR-based design. In addition, the closed-loop system will be robust due to a large class of state-dependent nonlinear uncertainties (matched), confined within a sector, that may exist at the plant input. All these features make the LQR PI feedback design highly efficient in controlling aerial platforms. The next section introduces adaptive augmentation algorithms to extend robustness of an LQR PI controller, with respect to a wide class of matched uncertainties and beyond.

30.5 Model Reference Adaptive Control

The concept of model reference adaptive control (MRAC) was originally proposed in 1958 by Whitaker et al., at MIT (Whitaker et al. 1958), and later extended in (Butchart and Shackcloth 1965; Parks 1966). Current state of the art in adaptive control is well documented and can be found in textbooks (Narendra and Annaswamy 2005; Ioannou and Fidan 2006). The original intent of adaptive control was to specify the desired command-to-output performance of a servo-tracking system using a reference model that would define the ideal response of the system due to external commands. A generic block-diagram of the MRAC system is shown in Fig. 30.4.

As seen from the diagram, the controller parameter adjustments (the adaptive law) are made based on the tracking error (the difference between the system actual response and its target specified by the reference model output), an output feedback from the process, and the external command. For the sake of clarity and in order to motivate further discussions, consider MRAC design equations for a scalar system shown below:

$$\begin{array}{l}
 \text{Process : } \dot{x} = ax + bu \\
 \text{Ref. Model : } \dot{x}_{\text{ref}} = a_{\text{ref}}x_{\text{ref}} + b_{\text{ref}}r \\
 \text{Controller : } u = \hat{k}_x x + \hat{k}_r r \\
 \text{Adaptive Law : } \begin{cases} \dot{\hat{k}}_x = -\gamma_x x (x - x_{\text{ref}}) \\ \dot{\hat{k}}_r = -\gamma_r r (x - x_{\text{ref}}) \end{cases}
 \end{array} \tag{30.42}$$

where a and b are unknown constant parameters in the process dynamics with the known $sgnb > 0$. The control input u is selected such that the system state x follows the reference model state x_{ref} , driven by any bounded external command $r = r(t)$. Also in (30.42), the reference model data $a_{\text{ref}} < 0$ and b_{ref} are chosen to yield the desired speed of response and a DC gain (unity in most applications) from the reference model output $y_{\text{ref}} = x_{\text{ref}}$ to the system-regulated output $y = x$.

In this case, closed-loop system stability and global asymptotic tracking are achieved via a specific choice of the adaptive law in (30.42), with the adaptive

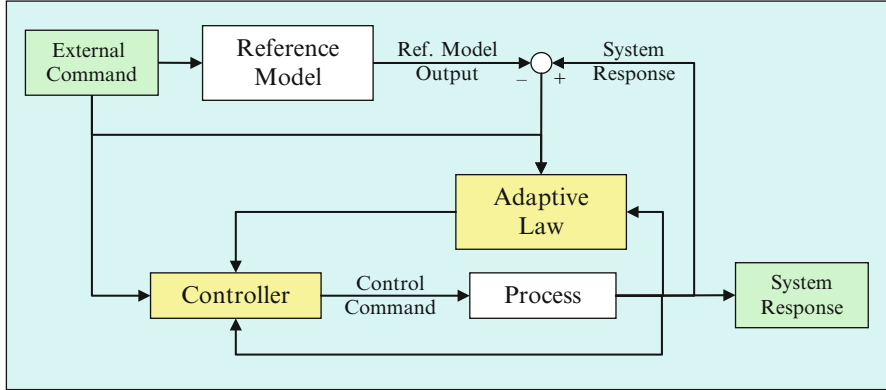


Fig. 30.4 MRAC block-diagram

gains (\hat{k}_x, \hat{k}_r) , whose dynamics are influenced by two positive constant rates of adaptation (γ_x, γ_r) . As seen from (30.42), the state tracking error

$$e = x - x_{\text{ref}} \quad (30.43)$$

drives the adaptive laws. Existence of a servo-control solution for this particular scalar dynamics is provided by the matching conditions:

$$\begin{aligned} a_{\text{ref}} &= a + b k_x \\ b_{\text{ref}} &= b k_r \end{aligned} \quad (30.44)$$

where k_x and k_r denote the ideal unknown constant parameters (gains of the ideal controller). For scalar dynamics, such as the process in (30.42), it is clear that the matching relations (30.44) always have a solution. Let,

$$\Delta k_x = \hat{k}_x - k_x, \quad \Delta k_r = \hat{k}_r - k_r \quad (30.45)$$

represent the parameter estimation errors. Substituting the matching conditions (30.44) into (30.42), one can derive the tracking error dynamics:

$$\dot{e} = a_{\text{ref}} e + b (\Delta k_x x + \Delta k_r r) \quad (30.46)$$

which indeed define transients in the corresponding closed-loop system. The tracking error dynamics and the transient dynamics are equivalent notions. If and when e becomes small (asymptotically), the system output tracks the reference model with diminishing errors. On the other hand, the transient dynamics define what happens between the start of a maneuver and the time when the error gets small. Both questions will be addressed next.

Going back to (30.46), one can employ Lyapunov arguments (Khalil 1996) to prove global asymptotic stability of the tracking error dynamics. In fact, using a radially unbounded quadratic Lyapunov function candidate in the form

$$V(e, \Delta k_x, \Delta k_r) = e^2 + \left(\frac{\Delta k_x^2}{\gamma_x} + \frac{\Delta k_r^2}{\gamma_r} \right) * b \quad (30.47)$$

it is not difficult to show that with the adaptive law (30.42), the time derivative of V , evaluated along the trajectories of the error dynamics (30.46), becomes nonpositive. This argument constitutes the inverse Lyapunov-based design. It provides (a) the adaptive law and (b) the required proof of closed-loop global asymptotic stability. As a result, one can formally show that for any initial condition, any bounded time-varying external command, and any positive rates of adaptation, the tracking error dynamics (30.46) are globally asymptotically stable:

$$\lim_{t \rightarrow \infty} |e(t)| = \lim_{t \rightarrow \infty} |x(t) - x_{\text{ref}}(t)| = 0 \quad (30.48)$$

and all signals in the corresponding closed-loop dynamics remain uniformly bounded, forward in time.

30.5.1 MRAC Design for MIMO Dynamics

The scalar MRAC design methodology generalizes to multi-input-multi-output (MIMO) systems with matched parametric uncertainties in the form

$$\dot{x} = Ax + B\Lambda(u + f(x)) \quad (30.49)$$

where $x \in R^n$ is the system state, $u \in R^m$ is the control input, and $B \in R^{n \times m}$ is the known control matrix, while $A \in R^{n \times n}$ and $\Lambda \in R^{m \times m}$ are unknown constant matrices. In addition, it is assumed that Λ is diagonal, its elements λ_i are strictly positive, and the pair $(A, B\Lambda)$ is controllable. The uncertainty in Λ is introduced to model control effectiveness failures or modeling errors. In (30.49), the unknown possibly nonlinear vector function $f(x) : R^n \rightarrow R^m$ represents the system-matched uncertainty. It is assumed that each individual component $f_i(x)$ of $f(x)$ can be written as a linear combination of N known locally Lipschitz-continuous basis functions $\phi_i(x)$, with unknown constant coefficients. So,

$$f(x) = \Theta^T \Phi(x) \quad (30.50)$$

where $\Theta \in R^{N \times m}$ is a constant matrix of the unknown coefficients and

$$\Phi(x) = (\phi_1(x) \dots \phi_N(x))^T \in R^N$$

is the known regressor vector. Of interest is the design of a MIMO state feedback adaptive control law such that the system state x globally uniformly asymptotically tracks the state $x_{\text{ref}} \in R^n$ of the reference model

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} r(t) \quad (30.51)$$

where $A_{\text{ref}} \in R^{n \times n}$ is Hurwitz, $B_{\text{ref}} \in R^{n \times m}$, and $r(t) \in R^m$ is the external bounded command vector.

It is also required that during tracking, all signals in the closed-loop system remain uniformly bounded. Thus, given any bounded command $r(t)$, the control input u needs to be chosen such that the state tracking error

$$e(t) = x(t) - x_{\text{ref}}(t) \quad (30.52)$$

globally uniformly asymptotically tends to zero, that is,

$$\lim_{t \rightarrow \infty} \|x(t) - x_{\text{ref}}(t)\| = 0 \quad (30.53)$$

If matrices A and Λ were known, one could have calculated and applied the ideal fixed-gain control law:

$$u = K_x^T x + K_r^T r - \Theta^T \Phi(x) \quad (30.54)$$

and obtain the closed-loop system:

$$\dot{x} = (A + B \Lambda K_x^T) x + B \Lambda K_r^T r \quad (30.55)$$

Comparing (30.55) with the desired reference dynamics (30.51), it follows that for the existence of a controller in the form of (30.54), the ideal unknown control gains, K_x and K_r , must satisfy the matching conditions:

$$\begin{aligned} A + B \Lambda K_x^T &= A_{\text{ref}} \\ B \Lambda K_r^T &= B_{\text{ref}} \end{aligned} \quad (30.56)$$

Assuming that these matching conditions hold, it is easy to see that using (30.54) yields the closed-loop system which is exactly the same as the reference model. Consequently, for any bounded reference input signal $r(t)$, the fixed-gain controller (30.54) provides global uniform asymptotic tracking performance. Note that given a set of matrices $(A, B, \Lambda, A_{\text{ref}}, B_{\text{ref}})$, in general there is no guarantee that ideal gains K_x, K_r exist to enforce the matching conditions (30.56). In other words, the control law (30.54) may not be able to meet the design objective. However often in practice, the structure of A is known, and the reference model matrices $A_{\text{ref}}, B_{\text{ref}}$

are chosen so that the system (30.56) has at least one ideal solution pair (K_x, K_r) . Assuming that K_x, K_r in (30.56) do exist, consider the following control law:

$$u = \hat{K}_x^T x + \hat{K}_r^T r - \hat{\Theta}^T \Phi(x) \quad (30.57)$$

where $\hat{K}_x \in R^{n \times m}$, $\hat{K}_r \in R^{m \times m}$, $\hat{\Theta} \in R^{N \times n}$ are the estimates of the ideal unknown matrices K_x, K_r, Θ , respectively. These estimated parameters will be generated online through the inverse Lyapunov analysis. Substituting (30.57) into (30.49), the closed-loop system dynamics can be written as

$$\dot{x} = \left(A + B \Lambda \hat{K}_x^T \right) x + B \Lambda \left(\hat{K}_r^T r - \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \right) \quad (30.58)$$

Subtracting (30.51) from (30.58), it is possible to compute the closed-loop dynamics of the n -dimensional tracking error vector $e(t) = x(t) - x_{\text{ref}}(t)$:

$$\dot{e} = \left(A + B \Lambda \hat{K}_x^T \right) x + B \Lambda \left(\hat{K}_r^T r - \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \right) - A_{\text{ref}} x_{\text{ref}} - B_{\text{ref}} r \quad (30.59)$$

With the matching conditions (30.56) in place, one further gets

$$\begin{aligned} \dot{e} &= \left(A_{\text{ref}} + B \Lambda \left(\hat{K}_x - K_x \right) \right) x - A_{\text{ref}} x_{\text{ref}} + B \Lambda \left(\hat{K}_r - K_r \right) r - B \Lambda \\ &\quad \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \\ &= A_{\text{ref}} e + B \Lambda \left[\left(\hat{K}_x - K_x \right)^T x + \left(\hat{K}_r - K_r \right)^T r - \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \right] \end{aligned} \quad (30.60)$$

Let $\Delta K_x = \hat{K}_x - K_x$, $\Delta K_r = \hat{K}_r - K_r$, and $\Delta \Theta = \hat{\Theta} - \Theta$ represent the parameter estimation errors. In terms of the latter, the tracking error dynamics become

$$\dot{e} = A_{\text{ref}} e + B \Lambda \left[\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x) \right] \quad (30.61)$$

Introduce rates of adaptation: $\Gamma_x = \Gamma_x^T > 0$, $\Gamma_r = \Gamma_r^T > 0$, $\Gamma_\Theta = \Gamma_\Theta^T > 0$. Going back to analyzing stability of the tracking error dynamics (30.61), consider a globally radially unbounded quadratic Lyapunov function candidate in the form

$$\begin{aligned} V(e, \Delta K_x, \Delta K_r, \Delta \Theta) &= e^T P e + \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \Delta K_x + \Delta K_r^T \Gamma_r^{-1} \Delta K_r \right. \right. \\ &\quad \left. \left. + \Delta \Theta^T \Gamma_\Theta^{-1} \Delta \Theta \right] \Lambda \right) \end{aligned} \quad (30.62)$$

where $P = P^T > 0$ satisfies the algebraic Lyapunov equation

$$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q \quad (30.63)$$

for some $Q = Q^T > 0$. Then the time derivative of V , evaluated along the trajectories of (30.61), can be calculated:

$$\begin{aligned}
 \dot{V} &= \dot{e}^T P e + e^T P \dot{e} + 2 \operatorname{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \right] \Lambda \right) \\
 &= (A_{\text{ref}} e + B \Lambda (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x)))^T P e \\
 &\quad + e^T P (A_{\text{ref}} e + B \Lambda (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x))) \\
 &\quad + 2 \operatorname{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \right] \Lambda \right) \\
 &= e^T (A_{\text{ref}} P + P A_{\text{ref}}) e + 2 e^T P B \Lambda (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x)) \\
 &\quad + 2 \operatorname{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \right] \Lambda \right) \quad (30.64)
 \end{aligned}$$

Using (30.63) further yields

$$\begin{aligned}
 \dot{V} &= -e^T Q e + \left[2 e^T P B \Lambda \Delta K_x^T x + 2 \operatorname{tr} \left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x \Lambda \right) \right] \\
 &\quad + \left[2 e^T P B \Lambda \Delta K_r^T r + 2 \operatorname{tr} \left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r \Lambda \right) \right] \\
 &\quad + \left[-2 e^T P B \Lambda \Delta \Theta^T \Phi(x) + 2 \operatorname{tr} \left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \Lambda \right) \right] \quad (30.65)
 \end{aligned}$$

Via the well-known trace identity

$$\begin{aligned}
 \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta K_x^T x}_b &= \operatorname{tr} \left(\underbrace{\Delta K_x^T x}_b \underbrace{e^T P B \Lambda}_{a^T} \right) \\
 \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta K_r^T r}_b &= \operatorname{tr} \left(\underbrace{\Delta K_r^T r}_b \underbrace{e^T P B \Lambda}_{a^T} \right) \\
 \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta \Theta^T \Phi(x)}_b &= \operatorname{tr} \left(\underbrace{\Delta \Theta^T \Phi(x)}_b \underbrace{e^T P B \Lambda}_{a^T} \right) \quad (30.66)
 \end{aligned}$$

Substituting (30.66) into (30.65) results in

$$\begin{aligned}
 \dot{V} &= -e^T Q e + 2 \operatorname{tr} \left(\Delta K_x^T \left[\Gamma_x^{-1} \dot{\hat{K}}_x + x e^T P B \right] \Lambda \right) \\
 &\quad + 2 \operatorname{tr} \left(\Delta K_r^T \left[\Gamma_r^{-1} \dot{\hat{K}}_r + r e^T P B \right] \Lambda \right) + 2 \operatorname{tr} \left(\Delta \Theta^T \left[\Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi(x) e^T P B \right] \Lambda \right) \quad (30.67)
 \end{aligned}$$

If the adaptive laws are selected as

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x x e^T P B \\ \dot{\hat{K}}_r &= -\Gamma_r r(t) e^T P B \\ \dot{\hat{\Theta}} &= \Gamma_\Theta \Phi(x) e^T P B\end{aligned}\quad (30.68)$$

then the time derivative of V in (30.67) becomes globally negative semidefinite:

$$\dot{V} = -e^T Q e \leq 0 \quad (30.69)$$

Therefore, the closed-loop error dynamics are uniformly stable. Hence, the tracking error $e(t)$ as well as the parameter estimation errors $\Delta K_x(t)$, $\Delta K_r(t)$, and $\Delta \Theta(t)$ are uniformly bounded and so are the parameter estimates $\hat{K}_x(t)$, $\hat{K}_r(t)$, and $\hat{\Theta}(t)$. Since $r(t)$ is bounded and A_{ref} is Hurwitz, then $x_{\text{ref}}(t)$ and $\dot{x}_{\text{ref}}(t)$ are bounded. Consequently, the system state $x(t)$ is uniformly bounded, and the control input $u(t)$ in (30.57) is bounded as well. The latter implies that $\dot{x}(t)$ is bounded, and thus $\dot{e}(t)$ is bounded. Furthermore, the second time derivative of $V(t)$

$$\ddot{V} = -e^T Q \dot{e} = -2e^T Q \dot{e} \quad (30.70)$$

is bounded, and so $\dot{V}(t)$ is uniformly continuous. Since in addition, $V(t)$ is lower bounded and $\dot{V}(t) \leq 0$, then using Barbalat's lemma (Khalil 1996) gives $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$, which implies (via (30.69))

$$\left[\lim_{t \rightarrow \infty} e^T Q e = 0 \right] \Leftrightarrow \left[\lim_{t \rightarrow \infty} \|e\| = 0 \right] \Leftrightarrow \left[\lim_{t \rightarrow \infty} \|x(t) - x_{\text{ref}}(t)\| = 0 \right] \quad (30.71)$$

It has been formally proven that the state tracking error $e(t)$ tends to the origin globally, uniformly, and asymptotically. The MIMO command tracking problem is solved, and a summary of the derived MRAC design equations is given in Table 30.1.

30.5.2 MRAC Design Modifications for Robustness

Shown in Table 30.1, the design is valid for MIMO dynamics (30.9), with a positive definite diagonal matrix Λ (control uncertainty) and a matched unknown-in-parameters function $f(x) = \Theta^T \Phi(x)$, where $\Theta \in R^{N \times m}$ represents a constant matrix of unknown parameters, $\Phi(x) \in R^N$ is the known regressor vector, and N is an integer. If components of the regressor are chosen from a certain class of approximation-capable maps (such as splines, trigonometric functions, polynomials, Gaussians, radial basis functions, sigmoidal/ridge functions), then the MIMO adaptive laws will provide semiglobal stability and command tracking for a wide class of matched uncertain functions $f(x)$ and bounded noise $\xi(t)$.

Table 30.1 MIMO MRAC laws

| | |
|-----------------------------|---|
| Open-loop plant | $\dot{x} = A x + B \Lambda (u + \Theta^T \Phi(x))$ |
| Reference model | $\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} r$ |
| Model matching conditions | $A + B \Lambda K_x^T = A_{\text{ref}}, \quad B \Lambda K_r^T = B_{\text{ref}}$ |
| Tracking error | $e = x - x_{\text{ref}}$ |
| Control input | $u = \hat{K}_x^T x + \hat{K}_r^T r - \hat{\Theta}^T \Phi(x)$ |
| Algebraic Lyapunov equation | $P A_{\text{ref}} + A_{\text{ref}}^T P = -Q$ |
| MIMO MRAC laws | $\dot{\hat{K}}_x = -\Gamma_x x e^T P B$ $\dot{\hat{K}}_r = -\Gamma_r r(t) e^T P B$ $\dot{\hat{\Theta}} = \Gamma_{\Theta} \Phi(x) e^T P B$ |

Table 30.2 Robustness modifications in MRAC design

| | |
|---------------------|---|
| Dead-zone | $\dot{\hat{\Theta}} = \Gamma_{\Theta} \Phi(x) \mu(\ e\) e^T P B$ |
| σ -mod | $\dot{\hat{\Theta}} = \Gamma_{\Theta} \left(\Phi(x) e^T P B - \sigma \hat{\Theta} \right)$ |
| e mod | $\dot{\hat{\Theta}} = \Gamma_{\Theta} \left(\Phi(x) e^T P B - \sigma \ \Phi(x) e^T P B\ \hat{\Theta} \right)$ |
| Projection operator | $\dot{\hat{\Theta}} = \text{Proj} \left(\hat{\Theta}, \Gamma_{\Theta} \Phi(x) e^T P B \right)$ |

These are nonparametric uncertainties. In order to mitigate the latter, the adaptive laws must be modified to become robust (Ioannou and Fidan 2006; Narendra and Annaswamy 2005). Table 30.2 shows four robustness modifications for adaptive control of MIMO uncertain systems, with their dynamics extended to include integrated tracking errors.

Overall, an MRAC controller enforces global uniform asymptotic tracking performance of the preselected reference model dynamics, driven by a bounded time-varying command while keeping the rest of the signals in the corresponding closed-loop system uniformly bounded. Such a controller adapts to matched uncertainties, and it remains robust to nonparametric nonmatched time-varying process noise $\xi(t)$. The key feature that enables robustness of MRAC to process noise is the dead-zone modification (Fig. 30.5):

$$\mu(\|e\|) = \max \left(0, \min \left(1, \frac{\|e\| - \delta e_0}{(1 - \delta) e_0} \right) \right) \quad (30.72)$$

where $0 < \delta < 1$ is a constant.

The dead zone freezes the MRAC laws if and when the magnitude (2-norm) of the tracking error vector

$$e = x - x_{\text{ref}}$$

becomes smaller than a preset tolerance e_0 . Any realistic adaptive system must have a continuous dead-zone modification, such as (30.72), to avoid potential discontinuities in feedback connections. The “must-have” dead-zone modification will prevent adaptive parameters from drifting away due to persisting noise or other nonparametric uncertainties in the system dynamics.

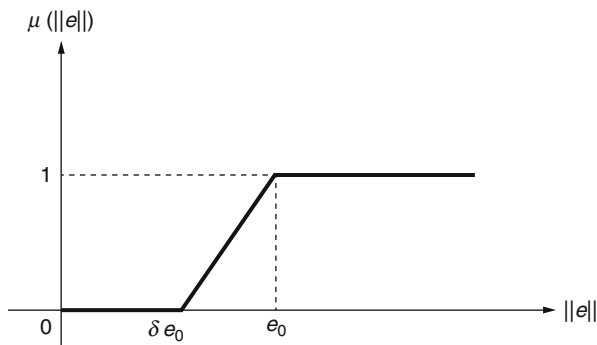


Fig. 30.5 The dead-zone function

Table 30.3 The projection operator

| | |
|--|--|
| Max parameter bounds | $\ \theta\ \leq \theta^{\max}$ |
| Convex function | $f(\hat{\theta}) = \frac{(1+\varepsilon)\ \theta\ ^2 - (\theta^{\max})^2}{\varepsilon(\theta^{\max})^2}$ |
| Two convex sets | $\Omega_0 = \{\theta : f(\theta) \leq 0\} = \{\theta : \ \theta\ \leq \frac{\theta^{\max}}{\sqrt{1+\varepsilon}}\}$ $\Omega_1 = \{\theta : f(\theta) \leq 1\} = \{\theta : \ \theta\ \leq \theta^{\max}\}$ |
| Projection operator | $\text{Proj}(\theta, y) = \begin{cases} y - \frac{\Gamma \nabla f(\theta) (\nabla f(\theta))^T}{(\nabla f(\theta))^T \Gamma \nabla f(\theta)} y f(\theta), & \text{if } [f(\theta) > 0 \wedge (y^T \nabla f(\theta)) > 0] \\ y, & \text{if not} \end{cases}$ |
| Convex inequality for proof of stability | $(\theta - \theta^*)^T (\Gamma^{-1} \text{Proj}(\theta, \Gamma y) - y) \leq 0,$ $\forall \theta^* \in \Omega_0, \theta \in \Omega_1, y \in R^n$ |
| Uniform boundedness of parameters | $\dot{\theta} = \text{Proj}(\theta, \Gamma y)$ $[\theta(0) \in \Omega_0] \Rightarrow [\theta(t) \in \Omega_1, \forall t \geq 0]$ |

Also of key importance is the projection operator modification (Ioannou and Fidan 2006), shown in Table 30.3.

As defined, this modification acts on two column vectors $(\theta, \Gamma y)$. For matrices, the projection operator is applied column-wise. By design, projection-based MRAC laws will force the tracking error to become small while keeping the adaptive parameters within their prespecified bounds.

Without robustness modifications, the adaptive law dynamics are defined by integrating a nonlinear function, represented by the regressor vector $\Phi(x)$, multiplied by a linear combination of the state tracking errors $(e^T P B)$. This product is further multiplied by a constant matrix Γ_Θ (the integral gain), and finally it is integrated to yield the adaptive parameters $\hat{\Theta}(t)$ (see Fig. 30.6).

As seen from the block-diagram, there is a chain of nonlinear integrators in a feedback loop, whose output constitute the adaptive parameters. In all practical applications, feedback integrators must be “managed” in the sense that their

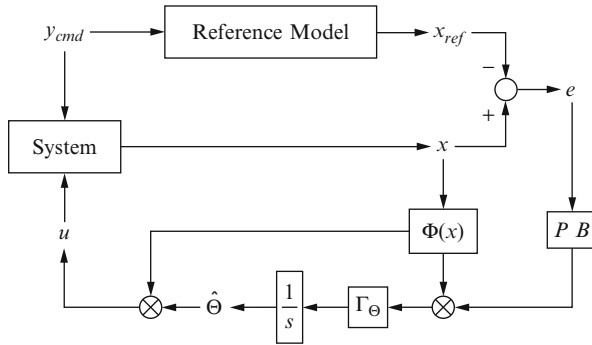


Fig. 30.6 MRAC system viewed as a nonlinear integral feedback controller

output signals (i.e., the adaptive parameters) need to be constrained. This prevents integrators against “winding up” due to nonlinear saturation functions in the control channels, where the system achievable control limits are defined and enforced. Control techniques that prevent the integrator windup problems are called the “anti-windup” methods, and the projection operator is one of them. So in practice, an MRAC architecture would consist of the smoothed dead-zone modification coupled with the projection operator. These are the two must-have modifications for enabling MRAC systems to efficiently operate in unknown environment.

30.5.3 Observer-Based MRAC Design with Transient Guarantees

Even though an adaptive controller enables command tracking asymptotically in time (as $t \rightarrow \infty$), it provides no uniformly guaranteed bounds on how large the transients might become prior to acquiring the command. In order to yield fast tracking and thus shorten transients, one needs to increase the rates of adaptation and, thus, to speed up MRAC laws. However, experience shows that if these rates grow large, unwanted transient oscillations will appear during the initial few seconds (the transient time) of operation. The balance between achieving fast tracking and avoiding undesired transients constitutes the MRAC design trade-off phenomenon. In essence, the rates of adaptation must be chosen large enough for fast tracking but not too large so that unwanted transients are precluded.

To understand the intricacies in MRAC design, reconsider the scalar design (30.42). What complicates the MRAC tuning process is the direct dependence of the transient dynamics (30.46) on (a) the external command and (b) the initial conditions for the system and the adaptive controller. These dependencies may too lead to undesirable transients. Consider the error dynamics (30.46). Using Lyapunov arguments, one can prove that the time-varying signal

$$\varphi(t) = b(\Delta k_x(t)x(t) + \Delta k_r(t)r(t)) \quad (30.73)$$

is uniformly bounded in time and that the tracking error $e(t)$ globally asymptotically tends to zero, as shown in (30.48). Still, the time constant of the transient dynamics (30.46) $\tau_e = \frac{1}{|a_{\text{ref}}|}$ is exactly the same as in the reference model (30.42). Although having the same time constant in both systems is theoretically correct, any control practitioner would want to have the transient dynamics (30.46) evolve faster than the desired reference model. In other words, the transients must die out quickly, relative to the dynamics of the reference model trajectories. This design requirement is identical to the one that takes place during the construction of asymptotic state observers, originally developed by Luenberger in his PhD thesis at Stanford (1963). Per Luenberger, the reference model in (30.42) represents an open-loop observer. So, just like in choosing the closed-loop observer dynamics, one can add an error feedback term to the reference model and arrive at the observer-like reference model:

$$\dot{x}_{\text{ref}} = a_{\text{ref}} x_{\text{ref}} + b_{\text{ref}} r + \boxed{k_e(x - x_{\text{ref}})} \quad (30.74)$$

Error Feedback Term

where $k_e > 0$ is the reference model feedback gain. The newly introduced error feedback term in (30.74) is equivalent to the output innovation feedback in a state observer. It is easy to see that in this case, the corresponding error dynamics become faster than the open-loop reference model from (30.42):

$$\dot{e} = (a_{\text{ref}} - k_e) e + b (\Delta k_x x + \Delta k_r r) \quad (30.75)$$

Once again, Lyapunov-based arguments can be easily repeated to prove (a) global asymptotic stability of the modified error dynamics (30.74) and (b) uniform boundedness of all signals in the related closed-loop system. Readers familiar with the MRAC stability proof concept should recognize that using the same Lyapunov function candidate (30.47), one needs to compute its time derivative along the trajectories of (30.75), substitute the adaptive law from (30.42), and then show that the resulting time derivative is globally nonpositive. This will prove uniform boundedness of the tracking error e and of the parameter estimation errors (30.45). Furthermore, since in the observer-like reference model (30.74), $a_{\text{ref}} < 0$ and the error feedback term is bounded, then the model state x_{ref} is bounded as well. The rest of the proof follows standard (in MRAC) stability arguments, finally arriving at (30.48).

Revised block-diagram with the observer-like reference model (30.74) is shown in Fig. 30.7.

As the reference model error feedback gain k_e is increased, the system transient dynamics become less oscillatory. In order to gain further insights into the transient behavior, choose $k_0 > 0$, a small positive parameter ε , and redefine the reference model feedback gain:

$$k_e = \frac{k_0}{\varepsilon} \quad (30.76)$$

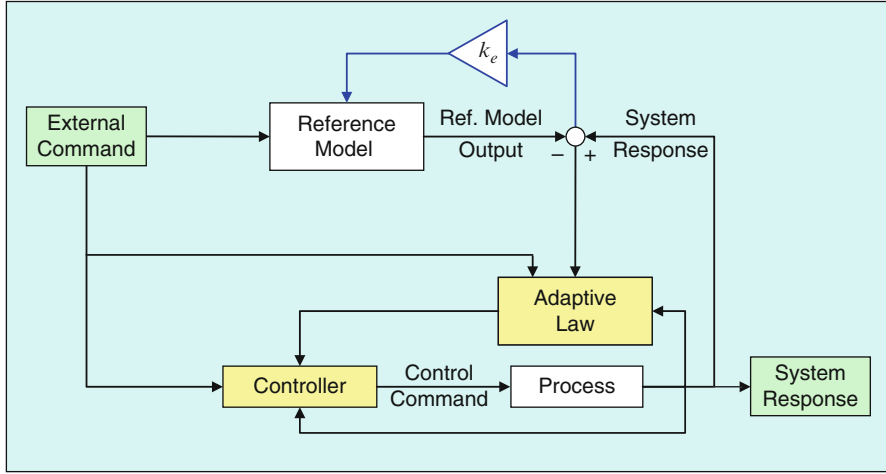


Fig. 30.7 MRAC block-diagram with observer-like reference model

Then the modified error dynamics (30.75) become

$$\varepsilon \dot{e} = (\varepsilon a_{\text{ref}} - k_0) e + \varepsilon \underbrace{[b (\Delta k_x x + \Delta k_r r)]}_{\varphi(t)} \quad (30.77)$$

Since all signals in the closed-loop system are uniformly bounded, it is not difficult to show that there exists a strictly positive finite constant $0 < \varphi_{\max} < \infty$ such that for any $\varepsilon > 0$, the upper bound $|\varphi(t)| \leq \varphi_{\max}$ holds uniformly in time and ε . Furthermore, starting from an initial condition $e(0) = e_0$, the solution of (30.77) can be written explicitly:

$$e(t) = e^{(a_{\text{ref}} - \frac{k_0}{\varepsilon})t} e(0) + \int_0^t e^{(a_{\text{ref}} - \frac{k_0}{\varepsilon})(t-\tau)} \varphi(\tau) d\tau \quad (30.78)$$

and one can compute an upper bound for this signal:

$$|e(t)| \leq e^{-k_0 \frac{t}{\varepsilon}} |e_0| + \frac{\varphi_{\max}}{k_0} \varepsilon \quad (30.79)$$

This relation is valid for any fixed $\varepsilon > 0$ uniformly in time. So, the system state $x(t)$ converges within $\left(\pm \frac{\varphi_{\max}}{k_0} \varepsilon\right)$ of the reference model state $x_{\text{ref}}(t)$ exponentially fast and at the rate no slower than $e^{-k_0 \frac{t}{\varepsilon}}$. This term gives an upper-bound quantification for the decay rate of the MRAC transient dynamics due to initial conditions mismatch $x(0) \neq x_{\text{ref}}(0)$. Otherwise, the system transients would remain within

ε -dependent bounds $\left(\pm \frac{\varphi_{\max}}{k_0} \varepsilon\right)$. Consequently, the system transients are reduced by decreasing ε , which according to (30.76) corresponds to increasing the reference model feedback gain k_e . Being able to influence and shape the MRAC transient dynamics constitutes the essential benefit of the Luenberger-like reference model modification (30.74)–(30.76). Relation (30.79) can also be written as

$$x(t) = x_{\text{ref}}(t) + C e^{-k_e t} + o(1) \quad (30.80)$$

where $C > 0$ is a constant independent of k_e and $o(1)$ is the “small-O” signal (a function of time that decays to zero asymptotically, as $t \rightarrow \infty$). The second term in (30.80) defines the transient dynamics due to initial conditions. Consequently, with a large enough feedback gain k_e , MRAC transient dynamics can be quantified and forced to decay as fast as needed. Note that since k_e is inversely proportional to ε , then the obvious trade-off in the modified MRAC design would be to avoid high gain effects in the reference model.

As for standard MRAC, it is also possible to generalize the observer-based MRAC to a broad class of nonlinear MIMO uncertain dynamical systems in the form

$$\underbrace{\begin{pmatrix} \dot{e}_y \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{m \times m} & C_p \\ 0_{n_p \times m} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_y \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 0_{m \times m} \\ B_p \end{pmatrix}}_B \Lambda \left(u + \overbrace{\Theta_d^T \Phi_d(x_p)}^{d(x_p)} \right) + \underbrace{\begin{pmatrix} -I_{m \times m} \\ 0_{n_p \times m} \end{pmatrix}}_{B_{\text{ref}}} y_{\text{cmd}} \quad (30.81)$$

$$y = \underbrace{\begin{pmatrix} 0_{m \times m} & C_p \end{pmatrix}}_C x$$

These dynamics incorporate an n_p -dimensional open-loop system with m control inputs u - and m - regulated outputs y . This is the original plant, whose state is $x_p \in R^{n_p}$. The plant is augmented by the m -dimensional integrated output tracking error dynamics $\dot{e}_y = C_p x_p - y_{\text{cmd}}$, where $C_p \in R^{m \times n_p}$ is a known constant matrix. The order of the complete system (30.81) is $n = n_p + m$. In addition, $x \in R^n$ is the system state vector, $u \in R^m$ is the control input, $y \in R^p$ is the regulated output, $y_{\text{cmd}} \in R^m$ is the commanded signal for y to follow, $d(x_p) = \Theta_d^T \Phi_d(x_p) \in R^m$ is a nonlinear state-dependent matched parametric uncertainty, $\Theta_d \in R^{N \times m}$ is the matrix of unknown constant “true” parameters, and $\Phi_d(x_p) \in R^N$ is the known N -dimensional regressor vector, whose components are locally Lipschitz continuous in x , that is, there exists a finite positive known constant $0 < L_{\Phi_d} < \infty$ such that for any $(x_1, x_2) \in R^{n_p}$ from a bounded neighborhood of the origin, the following inequality holds

$$\|\Phi_d(x_1) - \Phi_d(x_2)\| \leq L_{\Phi_d} \|x_1 - x_2\| \quad (30.82)$$

Also in (30.81), $A \in R^{n \times n}$, $B \in R^{n \times m}$, $B_{\text{ref}} \in R^{n \times m}$, and $C \in R^{m \times n}$ are constant known matrices, while $\Lambda \in R^{m \times m}$ is a constant diagonal unknown matrix with strictly positive diagonal elements.

Consideration of the process dynamics (30.81) is largely motivated by aerospace applications, where x_p models the 6-DoF motion of an airborne platform and $d(x_p)$ represents uncertainties in the vehicle aerodynamic moments. By definition, the moment uncertainties appear together with the system control inputs, thus enforcing the matching conditions needed to justify mere existence of a control solution. Moreover, control actuator uncertainties, control effectiveness reduction, and other control failures are modeled by an unknown constant matrix Λ . Finally, inclusion of the integrated output tracking error $\dot{e}_{yI} = C_p x_p - y_{\text{cmd}}$ into the open-loop system leads to the extended system formulation (30.81). This inclusion is optional, yet it allows the designer to explicitly account for baseline controllers with integral feedback, and it also allows to avoid feedforward terms in a control solution. Other dynamics, such as structural notch filters, sensors, and actuators, can also be added in the formulation of the extended open-loop system.

In order to control a dynamical system such as (30.81), one needs the nominal system (no uncertainties) to be controllable. It is well known that controllability of (A_p, B_p) , coupled with the rank condition,

$$\text{rank} \begin{pmatrix} A_p & B_p \\ C_p & 0_{p \times m} \end{pmatrix} = n_p + m = n \quad (30.83)$$

ensures controllability of the extended pair (A, B) . Disregarding the system uncertainties, it is straightforward to form the ideal reference model dynamics:

$$\dot{x}_{\text{ref ideal}} = A_{\text{ref}} x_{\text{ref ideal}} + B_{\text{ref}} y_{\text{cmd}} \quad (30.84)$$

where

$$A_{\text{ref}} = A - B \underbrace{(R_{\text{ref}}^{-1} B^T P_{\text{ref}})}_{K_{\text{lqr}}^T} \quad (30.85)$$

is Hurwitz, K_{lqr} is the baseline LQR feedback gain, P_{ref} is the unique symmetric positive definite solution of the ARE,

$$P_{\text{ref}} A + A^T P_{\text{ref}} - P_{\text{ref}} B R_{\text{ref}}^{-1} B^T P_{\text{ref}} + Q_{\text{ref}} = 0 \quad (30.86)$$

and $(Q_{\text{ref}}, R_{\text{ref}})$ are some appropriately chosen symmetric positive definite matrices. Using the LQR design is the preferred way to formulate reference model dynamics and to embed basic performance of an LQR PI controller into system specifications. Due to inclusion of the integrated tracking error in (30.81), the DC gain of the reference model (30.84) is unity. Consequently, if $\Lambda = I_{m \times m}$ and $d(x) = 0_{m \times 1}$, then the baseline LQR PI linear state feedback control $u_{\text{lqr}} = -K_{\text{lqr}}^T x$ enforces global exponential stability of the reference model (30.84), and it makes the regulated output $y(t)$ track any bounded command $y_{\text{cmd}}(t)$, with bounded errors. For a step-input command, the LQR PI controller provides global exponential tracking with zero steady-state errors. Also, it is easy to see that such a choice of the reference model enforces the model matching conditions, whereby given a

Hurwitz matrix A_{ref} and an unknown constant positive definite diagonal matrix Λ , there exists a constant possibly unknown gain matrix K_x such that

$$A_{\text{ref}} = A - B \Lambda K_x^T \quad (30.87)$$

It is important to understand that in this case, existence of K_x is guaranteed for any controllable pair (A, B) and for any nonsingular matrix Λ . In particular, relations (30.85) and (30.87) imply

$$K_x = K_{\text{lqr}} \Lambda^{-1} \quad (30.88)$$

Using (30.87), it is convenient to rewrite the system dynamics (30.81) in the form

$$\begin{aligned} \dot{x} = A_{\text{ref}} x + B \Lambda \left(u + \underbrace{\left[K_x^T x + \Theta_d^T \Phi_d(x_p) \right]}_{\Theta^T} \right) + B_{\text{ref}} y_{\text{cmd}} \quad (30.89) \\ \underbrace{\left(K_x^T \Theta_d^T \right)}_{\Theta^T} \underbrace{\begin{pmatrix} x \\ \Phi_d(x_p) \end{pmatrix}}_{\Phi(x)} \end{aligned}$$

and then get

$$\dot{x} = A_{\text{ref}} x + B \Lambda \left(u + \Theta^T \Phi(x) \right) + B_{\text{ref}} y_{\text{cmd}} \quad (30.90)$$

The control goal of interest is bounded tracking of y_{cmd} in the presence of the system parametric uncertainties $\{\Lambda, \Theta\}$. Specifically, one needs to find a control input u such that the regulated output $y = C x \in R^m$ tracks any bounded time-varying command $y_{\text{cmd}}(t) \in R^m$ with bounded errors, while the rest of the signals in the corresponding closed-loop system remain bounded. In addition, it is desirable to have smooth and quantifiable transient characteristics in the closed-loop dynamics. Using Lyapunov-based arguments (Khalil 1996), coupled with asymptotic analysis (Kevorkian and Cole 1996), one can derive MRAC systems with quantifiable transient performance.

Similar to (30.74) and for the system dynamics (30.90), consider a Luenberger-like reference model in the form

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + \underbrace{\left[L_v(x - x_{\text{ref}}) \right]}_{\text{Error Feedback Term}} + B_{\text{ref}} y_{\text{cmd}} \quad (30.91)$$

where $\hat{x} \in R^n$ is the reference model state and $L_v \in R^{n \times n}$ is the error feedback gain, parameterized by a positive scalar $v > 0$ (to be defined). The system control input u is selected as

$$u = -\hat{\Theta}^T \Phi(x) \quad (30.92)$$

Substituting (30.92) into the system dynamics (30.90) gives

$$\dot{x} = A_{\text{ref}} x - B \Lambda \underbrace{\left(\hat{\Theta} - \Theta^T \right)}_{\Delta \Theta} \Phi(x) + B_{\text{ref}} y_{\text{cmd}} \quad (30.93)$$

where $\Delta \Theta \in R^{N \times m}$ denotes the matrix of parameter estimation errors.

In what follows, the pair $(L_v, \hat{\Theta})$ will be selected such that the system state x globally asymptotically tracks x_{ref} – the state of the observer-like reference model (30.91), and so $y \xrightarrow{t \rightarrow \infty} y_{\text{ref}}$. Also, one can show that x_{ref} tracks $x_{\text{ref ideal}}$, which in turn implies that $y_{\text{ref}} \xrightarrow{t \rightarrow \infty} y_{\text{ref ideal}}$. Furthermore, since the output of the ideal reference model (30.84) follows its command $y_{\text{ref ideal}} \rightarrow y_{\text{cmd}}$, with bounded errors, and $y \xrightarrow{t \rightarrow \infty} y_{\text{ref}} \xrightarrow{t \rightarrow \infty} y_{\text{ref ideal}}$, then the system-regulated output y will also track y_{cmd} with bounded errors. This argument constitutes the proposed design strategy.

Begin by choosing adaptive laws for $\hat{\Theta}$ so that x globally asymptotically tracks x_{ref} , in the presence of the system uncertainties. Let,

$$e = x - x_{\text{ref}} \quad (30.94)$$

denote the state tracking error. Subtracting (30.91) from (30.93), gives the system transient dynamics:

$$\dot{e} = (A_{\text{ref}} - L_v) e - B \wedge \Delta \Theta^T \Phi(x) \quad (30.95)$$

Choose the error feedback gain L_v as

$$L_v = P_v R_v^{-1} \quad (30.96)$$

where $P_v = P_v^T > 0$ is the unique solution of the following ARE,

$$P_v A_{\text{ref}}^T + A_{\text{ref}} P_v - P_v R_v^{-1} P_v + Q_v = 0 \quad (30.97)$$

with the ARE weight matrices (Q_v, R_v) selected as

$$Q_v = Q_0 + \left(\frac{\nu + 1}{\nu} \right) I_{n \times n}, \quad R_v = \frac{\nu}{\nu + 1} I_{n \times n} \quad (30.98)$$

using a constant parameter $\nu > 0$. This constant will eventually become the design “tuning knob,” where small values of ν yield better MRAC transients. However, the corresponding feedback gain L_v will increase at the rate of $\frac{1}{\nu}$. In fact, as ν tends to zero, the error feedback gain tends to infinity:

$$L_v = \left(1 + \frac{1}{\nu} \right) P_v = O\left(\frac{1}{\nu}\right) \quad (30.99)$$

while the solution P_v of the ARE (30.97) tends to a constant positive definite symmetric matrix P_0 . It is easy to verify that the ARE (30.97) possesses the unique symmetric positive definite solution P_v . Furthermore, because of (30.97), the observer closed-loop matrix

$$A_v = A_{\text{ref}} - L_v = A_{\text{ref}} - P_v R_v^{-1} = A_{\text{ref}} - P_v \left(1 + \frac{1}{v}\right) \quad (30.100)$$

satisfies

$$P_v \underbrace{\left(A_{\text{ref}} - \underbrace{P_v R_v^{-1}}_{L_v} \right)^T}_{A_v} + \underbrace{\left(A_{\text{ref}} - \underbrace{P_v R_v^{-1}}_{L_v} \right)}_{A_v} P_v + P_v R_v^{-1} P_v + Q_v = 0 \quad (30.101)$$

or equivalently

$$P_v A_v^T + A_v P_v = -P_v R_v^{-1} P_v - Q_v < 0 \quad (30.102)$$

and therefore, A_v is Hurwitz for any $v > 0$.

Since P_v is the unique symmetric positive definite solution of the ARE (30.97), then the matrix inverse $\tilde{P}_v = P_v^{-1}$ exists for any $v \geq 0$ and the following relation takes place:

$$A_v^T \tilde{P}_v + \tilde{P}_v A_v = -R_v^{-1} - \tilde{P}_v Q_v \tilde{P}_v < 0 \quad (30.103)$$

The design task is to choose adaptive laws for $\hat{\Theta}$ so that the tracking error e globally asymptotically tends to the origin. Toward that end, consider the following Lyapunov function candidate:

$$V(e, \Delta\Theta) = e^T \tilde{P}_v e + \text{trace}(\Lambda \Delta\Theta^T \Gamma_{\Theta}^{-1} \Delta\Theta) \quad (30.104)$$

where $\Gamma_{\Theta} = \Gamma_{\Theta}^T > 0$ is the adaptation rate. The time derivative of V , along the trajectories of the error dynamics (30.95), can be computed as

$$\begin{aligned} \dot{V}(e, \Delta\Theta) &= e^T \tilde{P}_v \dot{e} + \dot{e}^T \tilde{P}_v e + 2 \text{trace}(\Lambda \Delta\Theta^T \Gamma_{\Theta}^{-1} \dot{\hat{\Theta}}) \\ &= e^T \tilde{P}_v (A_v e - B \Lambda \Delta\Theta^T \Phi(x)) + (A_v e - B \Lambda \Delta\Theta^T \Phi(x))^T \tilde{P}_v e \\ &\quad + 2 \text{trace}(\Lambda \Delta\Theta^T \Gamma_{\Theta}^{-1} \dot{\hat{\Theta}}) \\ &= e^T (\tilde{P}_v A_v + A_v^T \tilde{P}_v) e - 2e^T \tilde{P}_v B \Lambda \Delta\Theta^T \Phi(x) \\ &\quad + 2 \text{trace}(\Lambda \Delta\Theta^T \Gamma_{\Theta}^{-1} \dot{\hat{\Theta}}) \end{aligned} \quad (30.105)$$

Because of (30.102) and using the properties of the matrix trace operator,

$$\dot{V}(e, \Delta\Theta) = -e^T (R_v^{-1} + \tilde{P}_v Q_v \tilde{P}_v) e + 2 \text{trace}(\Lambda \Delta\Theta^T (\Gamma_{\Theta}^{-1} \dot{\hat{\Theta}} - \Phi(x) e^T \tilde{P}_v B)) \quad (30.106)$$

If the adaptive laws are chosen as

$$\dot{\hat{\Theta}} = \Gamma_{\Theta} \Phi(x) e^T \tilde{P}_v B \quad (30.107)$$

then

$$\dot{V}(e, \Delta\Theta) = -e^T (R_v^{-1} + \tilde{P}_v Q_v \tilde{P}_v) e \leq 0 \quad (30.108)$$

and, hence, $V(e, \Delta\Theta)$ is the Lyapunov function for the error dynamics (30.95). For this reason, the tracking error signal e as well as the parameter error matrix $\Delta\Theta$ are uniformly bounded in time, that is, $(e, \Delta\Theta) \in L_\infty$. Since A_{ref} in (30.91) is Hurwitz by design and $(e, y_{\text{cmd}}) \in L_\infty$, then $(x_{\text{ref}}, \dot{x}_{\text{ref}}) \in L_\infty$ and consequently $x \in L_\infty$. Since the unknown parameters Θ are constant and $\Delta\Theta \in L_\infty$ then $\hat{\Theta} \in L_\infty$. The regressor vector $\Phi(x_p)$ is Lipschitz continuous and $(x, \hat{\Theta}) \in L_\infty$. Therefore, from definition (30.92) it follows that $u \in L_\infty$ and consequently $\dot{x} \in L_\infty$. Also, since $\dot{x}_{\text{ref}} \in L_\infty$, then $\dot{e} \in L_\infty$. Using (30.108) yields

$$\ddot{V}(e, \Delta\Theta) = -2e^T (R_v^{-1} + \tilde{P}_v Q_v \tilde{P}_v) \dot{e} \in L_\infty \quad (30.109)$$

The function V from (30.104) is lower bounded and has a nonincreasing time derivative as in (30.108). Thus, V tends to a limit, as $t \rightarrow \infty$. Also the function's second time derivative is uniformly bounded. Therefore, \dot{V} is a uniformly continuous function of time. Using Barbalat's lemma (Khalil 1996) implies that $\dot{V}(t)$ tends to zero, as $t \rightarrow \infty$. Finally, and due to (30.108),

$$\lim_{t \rightarrow \infty} \|e(t)\| = 0 \quad (30.110)$$

which proves global asymptotic stability of the tracking error, attained by the adaptive controller (30.92), the adaptive laws (30.107), and the observer-like reference model (30.91).

In order to show that x_{ref} asymptotically tracks $x_{\text{ref ideal}}$, it is sufficient to subtract (30.84) from (30.91) and write the dynamics of the reference model error $e_{\text{ref}} = x_{\text{ref}} - x_{\text{ref ideal}}$:

$$\dot{e}_{\text{ref}} = A_{\text{ref}} e_{\text{ref}} + L_v \underbrace{e(t)}_{o(1)} \quad (30.111)$$

Then,

$$e_{\text{ref}}(t) = \exp(A_{\text{ref}} t) e_{\text{ref}}(0) + \int_0^t \exp(A_{\text{ref}}(t - \tau)) L_v \underbrace{e(\tau)}_{o(1)} d\tau = o(1) \xrightarrow{t \rightarrow \infty} 0 \quad (30.112)$$

So $x \xrightarrow{t \rightarrow \infty} x_{\text{ref}} \xrightarrow{t \rightarrow \infty} x_{\text{ref ideal}}$, and hence,

$$(y = C x) \xrightarrow{t \rightarrow \infty} (y_{\text{ref}} = C x_{\text{ref}}) \xrightarrow{t \rightarrow \infty} (y_{\text{ref ideal}} = C x_{\text{ref ideal}}) \rightarrow y_{\text{cmd}}(t) \quad (30.113)$$

In other words, the system-regulated output y asymptotically tracks its ideal reference command $y_{\text{ref ideal}}$, and y also tracks its original command y_{cmd} with bounded errors.

Table 30.4 Observer-based MRAC design summary

| | |
|------------------------------------|---|
| Open-loop plant | $\dot{x} = A_{\text{ref}} x + B \Lambda (u + \Theta^T \Phi(x)) + B_{\text{ref}} y_{\text{cmd}}$ |
| Observer-like reference model | $\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + L_v (x - x_{\text{ref}}) + B_{\text{ref}} y_{\text{cmd}}$ |
| State tracking error | $e = x - x_{\text{ref}}$ |
| Riccati equation for adaptive laws | $P_v A_{\text{ref}}^T + A_{\text{ref}} P_v - P_v R_v^{-1} P_v + Q_v = 0$ |
| ARE weight matrices | $Q_v = Q_0 + \left(\frac{v+1}{v}\right) I_{n \times n}, \quad R_v = \frac{v}{v+1} I_{n \times n}$ |
| Observer gain | $L_v = P_v R_v^{-1}$ |
| Total control input | $u = -\hat{\Theta}^T \Phi(x)$ |
| MRAC laws | $\dot{\hat{\Theta}} = \Gamma_{\Theta} \Phi(x) e^T P_v^{-1} B$ |

The design summary is given in Table 30.4.

In order to analyze the transient dynamics (30.95), substitute (30.96) into (30.95) and write the transient error dynamics as

$$\dot{e} = \underbrace{(A_{\text{ref}} - P_v R_v^{-1})}_{\text{Hurwitz Matrix}} e - \underbrace{B \Lambda \Delta \Theta(t)^T \Phi(x(t))}_{\varphi(t) = \text{Uniformly Bounded Function of Time}} \quad (30.114)$$

Using (30.98) gives

$$\dot{e} = \left(A_{\text{ref}} - \left(1 + \frac{1}{v} \right) P_v \right) e - \varphi(t) \quad (30.115)$$

In (Lavretsky 2011), it is shown that the asymptotic relation

$$P_v = P_0 + O(v), \quad \text{as } v \rightarrow 0 \quad (30.116)$$

holds with a constant positive definite symmetric matrix P_0 . Then,

$$\dot{e} = \left(A_{\text{ref}} - \left(1 + \frac{1}{v} \right) (P_0 + O(v)) \right) e - \varphi(t) \quad (30.117)$$

or equivalently

$$v \dot{e} = (v A_{\text{ref}} - (v+1) (P_0 + O(v))) e - v \varphi(t) \quad (30.118)$$

Rewrite (30.118) as

$$\begin{aligned} v \dot{e} &= (v A_{\text{ref}} - (v+1) (P_0 + O(v))) e - v \varphi(t) \\ &= \left(-P_0 + \underbrace{(v A_{\text{ref}} - v (P_0 + O(v)) - O(v))}_{O(v)} \right) e + v \varphi(t) \\ &= (-P_0 + O(v)) e + v \varphi(t) \end{aligned} \quad (30.119)$$

or, equivalently,

$$\dot{e} = \frac{1}{\nu} (-P_0 + O(\nu)) e + \varphi(t) \quad (30.120)$$

It is not difficult to show (by direct integration), that solutions of (30.120) satisfy the following asymptotics

$$e(t) = O\left(e^{-\gamma \frac{t}{\nu}}\right) + O(\nu), \quad (\nu \rightarrow 0) \quad (30.121)$$

uniformly in time, with a positive constant γ and for all sufficiently small $\nu > 0$. So, the transient dynamics exponentially decay to a neighborhood of the origin, no slower than $O\left(e^{-\gamma \frac{t}{\nu}}\right)$. Moreover, the “diameter” of the convergence set can be made arbitrarily small, by choosing ν to be sufficiently small. This argument formally proves and quantifies transient dynamics improvements in MIMO MRAC systems with observer-like reference models.

There is also an alternative way to analyze the transient dynamics in (30.118). This system can be viewed as singularly perturbed, where ν plays the role of a small parameter. To understand the intricacies of the system behavior, one can employ the singular perturbation arguments (Khalil 1996; Kevorkian and Cole 1996). Setting $\nu = 0$ gives the isolated root $e = 0$ for the corresponding reduced system, which describes asymptotic behavior as $t \rightarrow \infty$, that is, for a sufficiently small $\nu > 0$, the error trajectories converge to a small neighborhood of the manifold $e \equiv 0$ and will evolve near this manifold thereafter. Next, the boundary-layer system is formed to quantify and characterize the transient dynamics. These dynamics are derived by “stretching” the time

$$\tau = \frac{t}{\nu} \quad (30.122)$$

rewriting (30.118) in the “fast” time scale τ , and then setting $\nu = 0$. The resulting boundary-layer dynamics

$$\frac{d e}{d \tau} = -P_0 e \quad (30.123)$$

are globally exponentially stable, since P_0 is symmetric and positive definite (Kevorkian and Cole 1996). In this case, one can claim (Khalil 1996) that for a sufficiently small $\nu > 0$, while starting from an initial time $t_0 \geq 0$, the singular perturbation system (30.118) has a unique solution $e(t, \nu)$, defined on an infinite interval $[t_0, \infty)$, and the asymptotic relation

$$e(t, \nu) = \bar{e}\left(\frac{t}{\nu}\right) + O(\nu) \quad (30.124)$$

holds uniformly on $[t_0, \infty)$, where $\bar{e}\left(\frac{t}{\nu}\right)$ is the solution of the boundary-layer system (30.123) and $t_0 > 0$ is the initial time instant. Since,

$$\bar{e}\left(\frac{t}{\nu}\right) = \exp(-P_0(t - t_0)) \bar{e}(0) \quad (30.125)$$

then substituting (30.125) into (30.124) results in

$$e(t, v) = \exp\left(-P_0 \left(\frac{t-t_0}{v}\right)\right) (x(t_0) - x_{\text{ref}}(t_0)) + O(v) \quad (30.126)$$

This asymptotic relation is conservative. In fact, it has been already proven that the tracking error $e(t, v)$ asymptotically converges to the origin, starting from any initial condition. Consequently,

$$\varphi(t) = B \Lambda \underbrace{\left[\Delta \Theta(t)^T \Phi(x(t)) \right]}_{o(1)} = o(1), \quad (t \rightarrow \infty) \quad (30.127)$$

and so, (30.126) can be rewritten as

$$x(t, v) = \underbrace{\exp\left(-P_0 \left(\frac{t-t_0}{v}\right)\right) (x(t_0) - x_{\text{ref}}(t_0))}_{\text{Transient Dynamics}} + \underbrace{x_{\text{ref}}(t) + O(v)o(1)}_{\text{Global Asymptotic Stability}} \quad (30.128)$$

where P_0 is a constant symmetric positive definite matrix, $o(1)$ is a function of time with $\lim_{t \rightarrow \infty} o(1) = 0$, and $O(v)$ decays to zero no slower than v . Design details and stability proofs can be found in (Lavretsky 2011).

The asymptotic expansion (30.128) quantifies the MRAC transient dynamics. Indeed, for a sufficiently small $v > 0$, the transients, described by the first term in (30.128), decay exponentially fast, while the second term defines asymptotic behavior of the tracking error, as $t \rightarrow \infty$. This constitutes the main benefit of using the error feedback in an observer-based reference model. Essentially, with a sufficiently small parameter $v > 0$, one ensures quantifiable transient characteristics, and the latter are given by the first term in (30.128). The observer-based MRAC design method represents a numerically efficient technique of reducing unwanted transient oscillations in state feedback/feedforward MRAC systems.

The plant dynamics (30.81) and the corresponding control problem formulations can be modified to include nonparametric uncertainties, such as matched uncertainty approximation errors and bounded possibly nonmatched process noise. In that case, one can use known robustification techniques (i.e., σ -modification, e -modification, and the projection operator) to prove bounded tracking performance and then establish transient characteristics. Also, the state feedback MRAC design, with an observer-like reference model, can be extended to adaptive output feedback controllers (Lavretsky 2012).

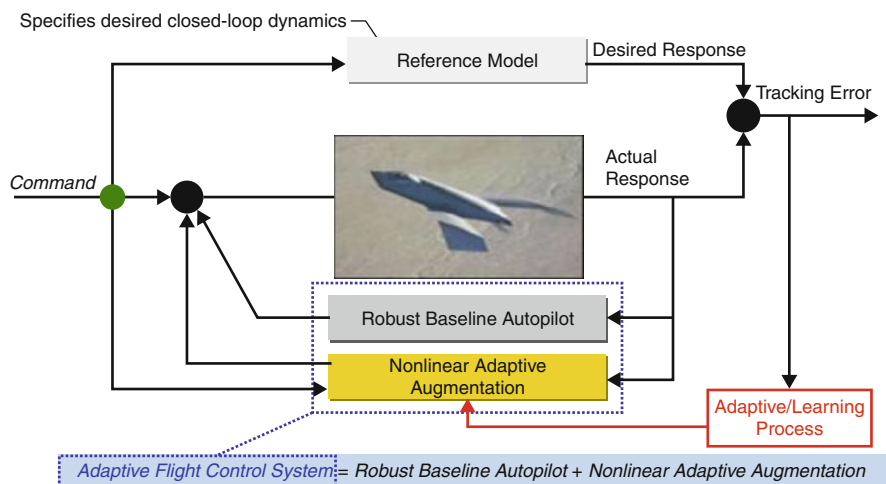


Fig. 30.8 (Robust + adaptive) flight control system

30.6 Conclusion

Robust and adaptive methods can be seamlessly combined to construct resilient controllers, applicable to a wide range of systems, including aerial platforms. A notional block-diagram is shown in Fig. 30.8.

This system is designed to track and execute external commands, provided by a pilot, a guidance logic, or an autonomous mission planner. The architecture embeds a robust baseline controller (LQR PI feedback). The reference model represents the baseline closed-loop dynamics that would be achieved under the baseline controller and without uncertainties. The adaptive control acts as an augmentation to the baseline. Its purpose is to recover the desired baseline performance while operating in the presence of “unknown unknowns” in the system dynamics and operational environment. As depicted in the figure, this control architecture was designed and flown on various vehicles at the Boeing Company. Some are in production today, and yet others were designed to test and verify extreme capabilities and resilience of aerial platforms equipped with robust and adaptive flight controllers.

References

- R.L. Butchart, B. Shackcloth, Synthesis of model reference adaptive control systems by Lyapunov's second method, in *Proceedings of 1965 IFAC Symposium on Adaptive Control*, Teddington, UK, 1965
- B. Etkin, *Dynamics of flight. Stability and control*, 2nd edn. (Wiley, New York, 1982)

- G.F. Franklin, J.D. Powell, A. Emami-Naeni, *Feedback Control of Dynamic Systems* (Addison-Wesley, Reading, 1986)
- P. Ioannou, P. Fidan, *Adaptive Control Tutorial*. Advances in Design and Control (SIAM, Philadelphia, 2006)
- J. Kevorkian, J.D. Cole, *Multiple Scale and Singular Perturbation Methods*. Applied Mathematical Sciences, vol 114 (Springer, New York, 1996)
- H. Khalil, *Nonlinear Systems*, 3rd edn. (Prentice Hall, Upper Saddle River, 1996)
- E. Lavretsky, Reference dynamics modification in adaptive controllers for improved transient performance, in *Proceedings of AIAA Guidance, Navigation and Control Conference*, Portland, OR, 2011
- E. Lavretsky, Adaptive output feedback design using asymptotic properties of LQG/LTR Controllers. *IEEE Trans. Autom. Control* **57**(6), 1587–1591 (2012)
- D. McRuer, I. Ashkenas, D. Graham, *Aircraft Dynamics and Automatic Control* (Princeton University Press, Princeton, 1990)
- K.S. Narendra, A.M. Annaswamy, *Stable Adaptive Control* (Dover, New York, 2005)
- P.D. Parks, Lyapunov redesign of model reference adaptive systems. *IEEE Trans. Autom. Control* **11**, 362–367 (1966)
- J.-J.E. Slotine, W. Li, *Applied Nonlinear Control* (Prentice Hall, Englewood Cliffs, 1995)
- B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation* (Wiley, New York, 1992)
- H.P. Whitaker, J. Yamron, A. Kezer, Design of Model-Reference Control Systems for Aircraft, Rep. R-164, Instrumentation Laboratory, MIT, Cambridge, MA (1958)

Section VII

UAV Communication Issues

Eric W. Frew

Kimon P. Valavanis and George J. Vachtsevanos

The absence of a pilot onboard an unmanned aircraft necessitates the development and application of robust, effective, and secure communication technologies that will enable unmanned aircraft to unmanned aircraft, unmanned aircraft to ground or mother ship communications, and unmanned aircraft to air traffic controller communications. Unmanned aircraft control and communication issues are tightly coupled and they are treated as such, since the “*Achilles heel*” of UAVs is controlling them when they are out of range.

► **Problem of UAV Communications** by Heppe summarizes the current consensus view (as of the end of 2012) regarding critical data flows for UAS, control link performance requirements, potential frequency bands that can satisfy the relevant technical as well as regulatory constraints, challenges that must be overcome to ensure reliable operation, and possible data link design principles that could lead to a safe and workable implementation. Both line-of-sight and beyond-line-of-sight data links are addressed. It is shown that the challenges to be faced and overcome are significant, but a safe and workable implementation appears to be achievable through data link diversity and other error correction and error recovery techniques.

► **Cognitive Networking for UAV Swarms** by Brown, McHenry, and Jaroonvanichkul examines the enabling role of cognitive radio technologies for UAS to access more spectrum needed for flight operations. After requirements are set, different architecture choices available to the UAS cognitive radio designer

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

are presented, including the communication architecture, the spectrum awareness techniques for assessing what spectrum is available, and the spectrum access techniques for deciding which available spectrum to use. Information in this chapter is relevant for the development of future UAS rules and standards.

► **Layered Approach to Networked Command and Control of Complex UAS** by Elston, Stachura, Dixon, Agrow, and Frew discusses different networking hardware, protocols, and sensors that when combined, they create a diverse and complex UAS through a layered design approach with modular supporting software. It is shown that critical software components, such as service discovery, simplify the inclusion of a diverse set of subsystems and sensors. Maintaining the modularity of these software components ensures that the system can be expanded while requiring minimal software changes. A detailed description of a system is presented, which enabled flight operations of a multi-vehicle unmanned aircraft system for performing targeted, in situ sampling of supercell thunderstorms during the 2010 VORTEX2 field campaign.

► **Cognitive Radio Architectures for Unmanned Aircraft Systems** by Wietfeld and Daniel discusses challenges and solution approaches with respect to self-organized and robust multi-UAV systems. The focus is on designing novel service-oriented system architecture to achieve a flexible deployment of UAS. The design aspects of mobility control algorithms addressing the competing requirements of communication reliability and spatial distribution of UAV swarms are demonstrated using aerial sensor networks and ad hoc aerial relay networks. Solution approaches rely on agent-based control of UAV swarms operating autonomously in dynamically changing environments. Different communication-aware algorithms for microscopic as well as macroscopic mobility control are presented, such as Cluster Breathing, Smart Cube, Potential Fields, and Role-Based Connectivity Management. The chapter also discusses how the network and application task-specific performance requirements are met with the proposed mobility control algorithms even in the cases of temporarily unavailable communication links. The self-healing capabilities therefore allow for reliable networking and control of aerial robot swarms in diverse use cases, such as emergency response, environmental monitoring, and ad hoc network provisioning.

Collectively, this section addresses current and future trends and challenges in single and networked UAV communication issues, also highlighting UAV integration issues and communication and networking requirements as dictated by air traffic organizations.

Stephen B. Heppe

Contents

32.1 Introduction and Scope of Communications Challenge..... 716

32.2 Detailed CNPC Performance Requirements..... 722

32.3 Frequency Bands for CNPC Communications 729

32.4 Technical Challenges for CNPC Communications 732

32.5 A Strawman Architecture for CNPC Communications 742

32.6 Conclusion 747

References 748

Abstract

Unmanned aircraft have been phenomenally successful in military operations, but have yet to achieve widespread civilian use. This is chiefly due to a concern that unmanned aircraft would pose a danger to other aircraft in the air (including manned aircraft) as well as humans and property on the ground. In order to address this concern and ensure an adequate level of safety, the communication link between the unmanned aircraft and the ground-based pilot – the control link – must carry certain safety-critical data and must be extremely robust and reliable. This chapter summarizes the current consensus view (2012) regarding critical data flows for unmanned aircraft systems, control link performance requirements, potential frequency bands that can satisfy the relevant technical as well as regulatory constraints, challenges that must be overcome to ensure reliable operation, and possible data link design principles that could lead to a safe and workable implementation. Both line-of-sight and beyond-line-of-sight (i.e., satellite based) data links are addressed. It will be seen that the

S.B. Heppe
Telenergy, Inc., Hood River, OR, USA
e-mail: steveheppe@aol.com

challenges are significant, but a safe and workable implementation appears to be achievable through data link diversity and other error correction and error recovery techniques.

32.1 Introduction and Scope of Communications Challenge

In principle, an unmanned aircraft (UA) or unmanned air vehicle (UAV) can operate in several modes:

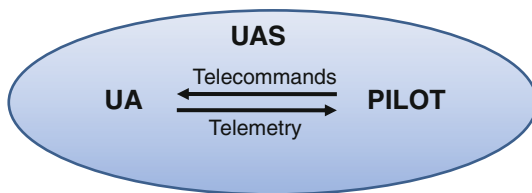
- (a) *Man-in-the-loop*. The pilot on the ground has direct real-time control of all aircraft systems including engine and aerodynamic control surfaces. This is the typical mode of operation for radio-controlled (RC) aircraft operated by the hobbyist.
- (b) *Man-on-the-loop (sometimes called “semi-autonomous”)*. Onboard automation, such as an autopilot, ensures stable and controlled flight in accordance with flight plans and other directives (including real-time flight commands) received from a pilot on the ground.
- (c) *Autonomous*. The aircraft operates without direct real-time human control and responds automatically to changes in its operating environment or aircraft state (although a human may optionally monitor aircraft operations in real time).

Of course, a single aircraft might be capable of all three modes of operation. For example, the nominal operating mode for most of a flight might be “man-on-the-loop,” but with the capability for “man-in-the-loop” for launch and recovery, and “autonomous return to base” in the event of control link failure.

While an aircraft capable of fully autonomous operation from launch to recovery might be a desirable *theoretical* goal, the practical reality is that the aviation industry – and society at large – does not yet trust the state of the art in automation to allow fully autonomous operation as a normal matter. The operating assumption is that, in the general fault-free case, a certified human pilot will have responsibility for the UA while it is operating (RTCA 2007, p. 11). This phrasing allows for at least two modes of operation: (a) man-in-the-loop (direct real-time control of the aircraft flight trajectory by the pilot); and (b) man-on-the-loop (semi-autonomous with the UA autopilot executing a flight plan previously commanded by the pilot). Of course, with the pilot not physically on the aircraft, a communications link must be provided. In addition, an autonomous mode will be needed for the (extremely rare) event of a control link failure. It should also be noted that policy guidance for very small UA (micro UA) might be less stringent but is yet to be established.

This chapter will generally refer to a UA, which is synonymous with a UAV, as part of an “unmanned aircraft system” (UAS) as shown in Fig. 32.1. The UAS consists of the UA and the pilot. These two elements are tied together by control link interactions that provide telemetry and status on the “return link” (UA to pilot) and telecommands on the “forward link” (pilot to UA). For a line-of-sight (LOS) system, the return link and forward link are synonymous with “downlink” and “uplink,” respectively. In a beyond-line-of-sight (BLOS) system, both the return link and the forward link comprise an uplink and downlink to/from a satellite. For example, the

Fig. 32.1 Internal data flows for an unmanned aircraft system (UAS)



forward link comprises a satellite uplink from the pilot control station (or some other ground-based satellite gateway terminal to which the control station is connected) to the satellite, and a downlink from the satellite to the UA.

The UA and its pilot must solve all the problems of a more traditional (manned) aircraft. Specifically, the UAS as a whole must be able to “aviate, navigate, and communicate” – three verbs that are drilled into every pilot. The first, *aviate*, refers to managing and controlling the aircraft, ensuring stable flight and avoiding obstacles both in the air and on the ground. The second, *navigate*, refers to knowing one’s current position and flight plan. The third, *communicate*, refers to voice and data communication with air traffic control, other aircraft in the airspace (which may or may not be actively coordinating with the UA), and associated systems on the ground. In a traditional manned aircraft, only the last required a “technical” means of communication – typically a radio, although light signals and even hand signals have been used in the past. The first two (*aviate* and *navigate*) involved “communication” only in the sense of a pilot observing his or her instruments, making adjustments to those instruments, and manually controlling the aircraft itself (either directly, or indirectly via the autopilot). In the case of a UA, all three activities now involve technical means of communication since the intimate connection between the pilot and his or her UA must now be extended over longer distances – from hundreds of meters to thousands of kilometers. These new communications links must be highly reliable and exhibit very high levels of integrity in order to provide the same level of positive control that is achieved by a pilot actually flying in a plane. In order to quantify the levels of performance required, the aviation community relies on a paradigm called “required communications performance” (RCP) which consists of four performance metrics (ICAO 2008):

1. *Availability*. The probability that an operational communication transaction can be initiated when needed.
2. *Continuity*. The probability that an operational communication transaction can be completed within the communication transaction time.
3. *Integrity*. The probability that communications transactions are completed within the communication transaction time with undetected error.
4. *Latency and communication transaction time*. Latency is the maximum time to deliver a one-way message (or a two-way message plus response) for a given communications system. Communications transaction time is the time for completion of the operational communication transaction (including human response time) after which the initiator should revert to an alternative procedure.

“Availability” is simply the long-term average probability that a communications link exists and is able to support a needed communication transaction, at an arbitrary instant of time when a human (or automated system) attempts to send a message. If there is a single radio link connecting the pilot and UA, “availability” would be the probability that both the ground radio and airborne radio are in working order; that the radio link is not corrupted by rain, foliage, jamming, or airframe blockage; and that the radio channel is not over-subscribed by other users. Note that random bit errors on an otherwise fault-free channel are excluded from this definition. “Continuity” is a conditional probability that a communication transaction can be completed (within the necessary time limit), given that the communications facility was “available” at the start of the transaction. Continuity is typically affected by short-term “soft failures” such as multipath fading and airframe blockage. If one ignores the case where a communications link is “failed” at the start of a transaction but “recovers” and successfully delivers a message within the required timeframe, then the probability of actually delivering an arbitrary message is $\Pr\{\text{delivery}\} = (\text{availability}) (\text{continuity})$. Values from 0.9 to 0.999 are typical. Particular requirements for UA communications are identified below. The general term “reliability” can be used as a shorthand term encompassing both availability and continuity.

Integrity is the probability of an undetected error in a message (sometimes defined as the probability of experiencing an undetected error in any message over a defined span of time, such as an hour). Typical values range from 10^{-5} to 10^{-10} or even lower. This definition actually quantifies a “loss of integrity,” but it is easier to cite a single exponent than a large number of nines (e.g., either 5 nines or 10 nines, or even more).

Transaction expiration time is the time allowed to complete a transaction before an unsafe condition would result. Typical values have traditionally fallen in the range of seconds to hundreds of seconds. However, for linking a pilot to a UA for real-time perception of the airborne environment and real-time control, more stringent values are needed (fractions of a second to seconds). For one-way transactions such as delivery of telemetry, transaction expiration time is equivalent to the end-to-end latency of the data link. For two-way transactions, such as the delivery of a particular telemetry message indicating a potential collision threat, plus an uplink command executing an avoidance maneuver, transaction expiration time includes the two-way latency of the data link plus the pilot response time.

Before detailing the numerical requirements for communications performance, and engineering methods to achieve that performance, it is useful to review the types of communication that must be supported. Figure 32.2 illustrates electromagnetic (radio and radar) interactions between the UAS (the UA and its pilot) and external entities. Some of these have an obvious impact on UAS communications. For example, in the navigation domain, many UA will have GPS and VOR receivers (as examples). These must be controlled and managed by the pilot via the telecommand forward link, and their output data must be delivered to the pilot by the telemetry return link. These data flows are generally low bandwidth (i.e., only limited amounts of data per second) and repetitive (thus, occasional message losses are not critical). Similarly, ATC ground surveillance involves various airborne subsystems which

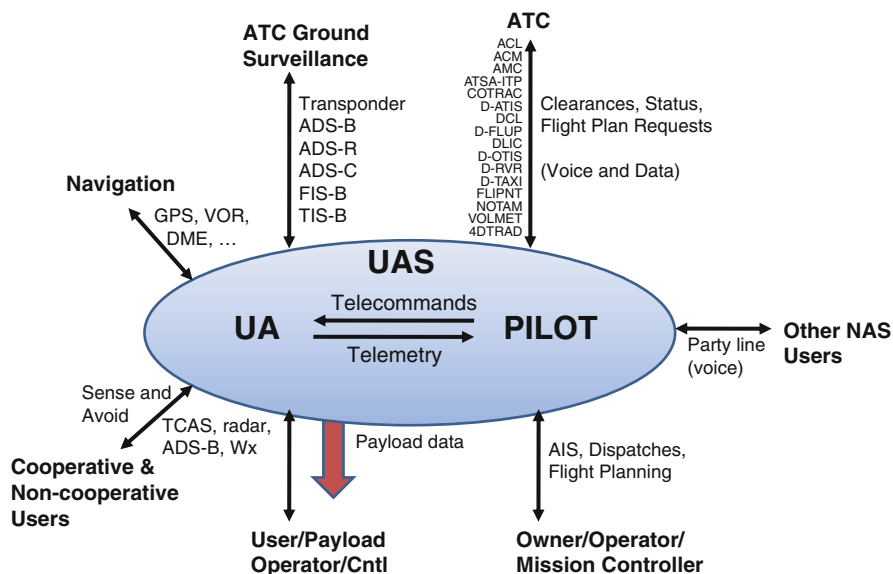


Fig. 32.2 UAS external interactions

must be controlled by the pilot (e.g., the radar transponder and automatic dependent surveillance – broadcast transmitter), and in some cases their output must be returned to the pilot as well (e.g., traffic information service – broadcast (TIS-B) receivers). It should be noted that some data can flow directly to or from the pilot without passing through the UA.

The sense and avoid data streams (for cooperative and non-cooperative targets) are some of the most critical since aircraft move quickly and pilots generally have only a few seconds to make key decisions when things go wrong. These data streams place stringent constraints on the availability, continuity, latency, and integrity of the telecommand and telemetry data links of medium and large UA. Typical airborne equipment in this domain includes traffic alert and collision avoidance (TCAS), the ADS-B receiver, traffic radar, and weather radar. The data from these sensors must flow down to the pilot as telemetry data with very high reliability and integrity, and very low latency. Pilot telecommand responses to the aircraft, necessary to control the aircraft trajectory, avoid obstacles, and ensure safety of flight, must be returned to the aircraft with similar levels of reliability, integrity, and latency.

The “air traffic control” (ATC) data flows comprise the traditional voice and data interactions between a pilot and the air traffic control system. The end-points of these interactions are the pilot on one side and the air traffic controller (or other ATC system element) on the other side. Typically for a manned aircraft, a radio on the aircraft would communicate with a radio at the desired ATC facility (such as an airport tower or en route center) or alternatively with a remote radio facility that was networked through ground infrastructure to ATC. This is shown graphically as the radio communications links labeled “A” and “B” in Fig. 32.3. These links

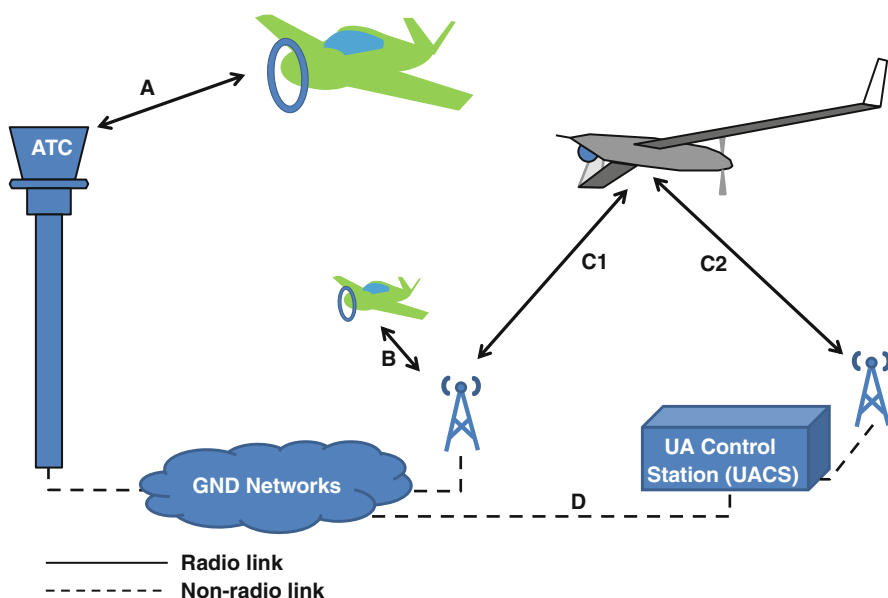


Fig. 32.3 Data link options for ATC communications

are usually shared on a party-line basis by many aircraft in a single ATC sector. For a UA, one could insist that ATC communications continue to be routed to (through) the aircraft as indicated by the links labeled “C1” and “C2” in the figure. This has the advantage that ATC deals with a UA just as it deals with a manned aircraft; the link C1 is equivalent to the link A or B from the standpoint of ATC. This would also maintain the party-line nature of the ATC/UA link since manned aircraft in the vicinity would be able to hear both the ATC and the relayed UA pilot’s transmissions. However, the voice (or data) on the aircraft must be routed to/from the pilot which requires a second radio link C2 in a different frequency band (i.e., to prevent cosite interference caused by simultaneous transmission and reception). C2 could be the same radio link used for UA telecommand and telemetry or a different link. Either way, the tandem combination of C1 and C2 will have lower end-to-end link availability, continuity, integrity, and worse latency, than either C1 or C2 alone. Furthermore, since there is no way for the UA to determine autonomously when a voice signal is coming from ATC and intended for its pilot, versus being transmitted by (or to) another pilot, the link C2 must actually carry *all* the traffic on link C1 – even though only a small fraction of this traffic is intended for the UA and its pilot. Hence, the bandwidth penalty is significant – each UA would effectively consume the equivalent spectrum of an entire ATC radio channel (albeit in a different band), instead of only a small fraction of a channel as is typical today.

An alternative is to recognize that the UA pilot is actually on the ground and potentially could be accessed without any reliance on radio at all. This is indicated by landline “D” in the figure. Since landlines and other forms of terrestrial communication can be made more robust compared to radio links, this approach has advantages in terms of end-to-end availability, continuity, integrity, and latency (in addition to the savings in terms of bandwidth). However, this so-called “wired ATC network” approach requires an enhancement to the switching and networking capability of the ATC system.

A third possibility is to rely on a radio link between ATC (or a remote radio tower) and the pilot in the UA control station (i.e., from one radio tower to another in the figure), thereby avoiding any change to the ATC switching system but nevertheless bypassing the UA itself, thereby avoiding the need for a relay. This approach is particularly convenient when the UA control station (the UACS) is in the same geographic area as, and has line-of-sight visibility to, the ATC facility or its remote radio tower.

In reality, all three approaches should be accommodated in a mature architecture since some missions may not allow the pilot control station to easily access terrestrial infrastructure (e.g., disaster relief, remote/oceanic operations, etc.). However, the “wired ATC network” approach should be emphasized due to its benefits in terms of availability, continuity, integrity, latency, and spectrum utilization efficiency.

Returning to Fig. 32.2, the party-line (voice) communication to other NAS users refers to the current situational awareness afforded by each pilot listening to the voice interaction between ATC and other aircraft in the vicinity. This may become less important in the future as ATC shifts to data link instead of voice, but party-line voice communications can be maintained (to the extent used) by suitable voice-activated switching capability between landlines and radios.

All of the external data flows addressed so far, in relation to Fig. 32.2, involve communications related to the safety of flight. These external data flows, along with the internal telecommand and telemetry data flows, are jointly identified as “control and non-payload communications” (CNPC). To the extent that radios are used for these flows (recall that ATC voice and data would ideally rely on landlines), the frequency channels used must be protected from a regulatory standpoint and reside in “aeronautical safety spectrum.” This is addressed further below.

The owner, operator, and mission controller data flows, in Fig. 32.2, connect the pilot to the aircraft owner/operator or mission commander and do not normally involve radio communication in protected aeronautical spectrum.

Finally, payload data (and control and status of the payload) must be addressed in the context of a complete system. However, this is a special category of communication since it does not involve safety of flight. Certain payloads, for example, video cameras, can involve very large data rates and bandwidths but must be handled outside of “aeronautical safety spectrum” if they are used as a mission payload. However, if used for ensuring safety of flight (e.g., for separation assurance or takeoff and landing support), they would fall within the CNPC “safety of flight” domain.

32.2 Detailed CNPC Performance Requirements

The international community has been working for several years to estimate the future bandwidth and performance requirements of UA CNPC. This work has proceeded within the RTCA (a not-for-profit corporation that functions as a Federal Advisory Committee for the U.S. Federal Aviation Administration), EUROCAE (a European body also dedicated to aviation standardization), the International Civil Aviation Organization (ICAO), and the International Telecommunications Union (ITU). As of this writing (2012), a consensus is beginning to emerge. This consensus can be understood in terms of the bandwidth needs of an individual UA, the aggregate bandwidth needs of all UA in the airspace (requiring additional assumptions on the number, type, and deployment of UAS), and the RCP performance parameters needed to ensure safety of flight.

Bandwidth Requirements for a Single Large UA. The key CNPC data flows illustrated previously in Fig. 32.2, which must be supported in aviation-protected spectrum, can be grouped into three broad categories: (1) command and control, (2) relay of ATC voice and data, and (3) sense and avoid. These involve forward link as well as return link communications which vary as a function of phase of flight and type of flight control employed (manual control, with the pilot directly commanding the control surfaces in real time, versus “automatic” (man-on-the-loop) with the pilot directing an onboard autopilot in terms of, for example, speed, climb rate, turn rate, heading, or a detailed flight plan). The estimated data throughput requirements (bits per second) are tabulated in Fig. 32.4 for a large,

| PHASE OF FLIGHT AND OPERATING MODE (MANUAL OR AUTOMATIC) | COMMAND AND CONTROL | | | | ATC RELAY | | | SENSE AND AVOID | | |
|--|---------------------|------------------|---------|------|-------------|------------|----|------------------|------------------------------|---------|
| | CONTROL | | NAVAIDS | | VOICE RELAY | DATA RELAY | | TARGET TRACKS | AIRBORNE WEATHER RADAR | VIDEO |
| | UPLINK (UL) | DOWNLINK (DL) | UL | DL | UL and DL | UL | DL | DL | DL | DL |
| | | | | | | | | | | |
| MAXIMUM FOR ANY PHASE OF FLIGHT [1] | 4600 | 7600 | 670 | 1100 | 4800 | 50 | 60 | 9100 | 28 000 | 270 000 |
| AVERAGE FOR MANUAL OPERATION | 1700 | 3200 | 670 | 870 | 4800 | 24 | 31 | 9100 | 8700 | 270 000 |
| AVERAGE FOR AUTOMATIC OPERATION | 440 | 650 | 140 | 190 | 4800 | 24 | 31 | 9100 | 8700 | 270 000 |
| OVERALL AVERAGE | 690 | 1200 | 250 | 330 | 4800 | 24 | 31 | 9100 | 8700 | 270 000 |
| References: RTCA SC-203 CC008 (May 2009) (relying in part on ITU-R, 2009, Table 16) | | | | | | | | | | |
| NOTES: | | | | | | | | | | |
| 1. Flight phases include departure, en route, and arrival. Pre-flight and post-flight phases are excluded. | | | | | | | | | | |
| 2. Data throughput values include overhead and are rounded to two significant figures. | | | | | | | | | | |
| 3. Target tracks assumes 60 targets in local environment of the UA. | | | | | | | | | | |
| 4. Video is only needed intermittently, for some aircraft. | | | | | | | | | | |
| 5. Overall average assumes 80% of aircraft are operating automatically in each flight phase; 20% operating manually. | | | | | | | | | | |

Fig. 32.4 Estimated peak and average non-payload throughput requirements (bps) for a typical large UA over a 4 h flight

well-equipped UA (smaller UA might not support all listed data flows). Note: The “uplink” and “downlink” terminology of Fig. 32.4 can be converted to “forward link” and “return link,” respectively, in order to generalize for both LOS and BLOS communication architectures. The bottom row indicates average requirements for a large UA assuming 80 % of such aircraft are operated “automatically” (man-on-the-loop). All values include link overhead for network control, error correction, and authentication.

Examining Fig. 32.4, the peak uplink (forward link) data throughput requirement is roughly 10 kbps (summing the uplink contributions in the top row), and the average data throughput requirement is roughly 5.8 kbps (summing the uplink contributions in the bottom row). A significant element in both of these values is ATC voice relay (4.8 kbps) which, as noted earlier, might be handled via landline without any RF spectrum impact at all – for at least some users. As a consequence, the future CNPC data link must be able to accommodate an uplink (forward link) data throughput of at least 10 kbps per UA (this might be required for some UA for some periods of time), but careful design might allow a large number of UA to share the spectrum resource with each UA consuming on average only 5.8 kbps of system capacity if the UA is used to relay ATC voice communication or only 1 kbps of system capacity if a “wired ATC network” is employed to connect ATC directly with the pilot, bypassing the UA.

Similarly, for the downlink (return link), the peak requirement is roughly 320 kbps (again driven by the terminal area approach phase), and the average is about 290 kbps. These are large numbers, but several caveats are important:

- (a) The 270 kbps for “sense-and-avoid video” (intended to enhance situational awareness) is assumed to represent an intermittent requirement. It is not clear, at this time, if video is intermittently *required* for some or all classes of UA or only *desirable*. Some UA might not have this subsystem, and the associated data burden, at all.
- (b) Weather radar is also a significant data load (the two values in the table reflect compressed versus uncompressed data depending on phase of flight) but might only be installed on a small fraction of the medium and larger UA. It seems likely that a pilot in a ground-based UA control station could access real-time weather products via landline or satellite broadcast, without requiring a dedicated data stream from his or her own UA.
- (c) Target tracks (sense-and-avoid downlink indications of nearby traffic, including potential “intruders” that represent a risk of collision) are arguably one of the most critical data flows in the entire system. The data volume of this component is estimated at roughly 9.1 kbps. It is imperative that the pilot have a complete and accurate understanding of nearby traffic. However, even here, there is a potential for capacity savings through, for example, “reporting by exception” as suggested at the end of this chapter.
- (d) As with the uplink assessment, a portion of the downlink is based on relay of ATC voice (4.8 kbps) which, as noted earlier, might be handled via landline without any RF spectrum impact at all – for at least some users.

As a result of these considerations, the future CNPC data link(s) must be able to accommodate up to 320 kbps per UA on the return link; however, some UA, some of the time, may require as little as 1.5–2 kbps if they do not use video for sense and avoid, do not have a weather radar installed and operational, rely on target reporting by exception, and rely on a wired ATC network for ATC voice to/from the pilot.

Because of the wide range of data throughput requirements for a single UA, it is imperative that a multi-access sharing system be implemented which minimizes wasted spectrum and allows for efficient allocation of radio channel resources to individual UA.

Bandwidth Requirements for a Regional Population of UA. The international community is currently attempting to allocate RF spectrum for UA CNPC data links. This requires an assessment of aggregate bandwidth requirements considering the needs of individual UA; the extent to which small, medium, and large UA participate in the data flows indicated above; the population distribution of small, medium, and large UA; the number of UA that might be operational at any one time; and the spatial distribution of these UA both geographically and in altitude (this affects the number of UAS that are mutually visible to one another as well as their associated ground stations). Furthermore, the total bandwidth required for a regional population of UA depends on whether the aircraft are supported by (a) line-of-sight (LOS) radios between the UA and the pilot as illustrated in Fig. 32.3 or (b) beyond-line-of-sight (BLOS) radios using satellite relay.

As a first step, the participating members of ITU-R Working Party 5B estimated the peak areal density of UA in the 2030 timeframe. Two methods were used. The first method estimated areal density for all aircraft and then assumed that 10 % of these would be UA. A second method attempted a bottom-up analysis by projecting the number and types of UA missions (military and civilian, governmental, and commercial) that could be implemented by 2030, assuming regulatory guidelines were in place. These two methods resulted in expected areal densities of between 8.6 and 10.4 UA per 10,000 km² (ITU-R 2009, Tables 5 and 7). This is equivalent to 0.00086–0.00104 UA/km². These are average densities which do not account for variations associated with human population density or particular mission requirements (military or otherwise).

The total expected population of UA was also subdivided into three broad aircraft classes: small, medium, and large. These three classes were assigned certain operational characteristics, such as typical operating altitude and avionics equipment and typical operating regime. Working from the more conservative (larger) areal density noted above, the consensus view is that, for a uniform geographic distribution, small UA operating below 300-m altitude can be expected to represent an areal density of ~ 0.0008 UA/km², medium UA operating between 300 and 5,500 m can be expected to represent an areal density of ~ 0.0002 UA/km², and large UA operating above 5,500 m can be expected to represent an areal density of ~ 0.00004 UA/km². The altitude range affects channel sharing and frequency reuse across geographic domains, and avionic equipment affects flight operations. For example, small UA were assumed to operate only in daylight at low altitude, within visual range of the pilot (and the UACS), and have only the most basic

flight controls and sensors and therefore have only limited need for data transfer in aviation-protected spectrum.

ITU-R WP5B also considered the impact of nonuniform geographic distribution of UA – for example, due to disaster relief such as floods and hurricanes and normal geographic asymmetries due to airport operations. While these factors can lead to a localized increase in areal density by as much as a factor of 10, the group determined that such nonuniformities could be accommodated by relying on spectrum resources in neighboring areas. Thus, total spectrum requirements could be based on the uniform distribution.

Based on the analyses performed, ITU-R WP5B concluded (ITU-R 2009) that the total aviation-protected spectrum requirements for UA CNPC data links are:

- 34 MHz for a terrestrial LOS system supporting all UA
- 56 MHz for a regional-beam satellite BLOS system supporting medium and larger UA only (small UA are assumed to be too small to support the antenna needed for a satellite BLOS system)

The satellite-based system requires more spectrum, despite supporting only a subset of the UA, because it is comparatively less able to rely on frequency reuse. Within a terrestrial LOS system, bandwidth requirements would be split between uplink and downlink with UACS to UA data flows consuming 4.6 MHz and UA to UACS data flows consuming 29.4 MHz. Within a satellite-based BLOS system comprising multiple satellites and a degree of frequency reuse, the forward links from the UACS to the satellite, and from the satellite to the UA, each consume 4.1 MHz and the return links from the UA to the satellite, and from the satellite back down to the UACS, each consume 24.05 MHz.

RCP Thresholds for CNPC Data Flows. In addition to the individual and aggregate bandwidth requirements, which affect user channelization and total system capacity, it is important to understand the required communications performance in terms of availability, continuity, integrity, and transaction expiration time (including latency). Figure 32.5 provides guidance on the RCP levels recommended for pilot-to-controller voice and data communications, based on ICAO (2008). This provides

| RCP Type | Transaction Time (sec) | Continuity (per flight hr) | Availability (per flight hr) | Integrity* (per flight hr) | Usage |
|----------|------------------------|----------------------------|------------------------------|----------------------------|--|
| RCP 10 | 10 | 0.999 | 0.99998 | 10 ⁻⁵ | Controller voice intervention supporting separation assurance in a 5 nmi radius environment |
| RCP 60 | 60 | 0.999 | 0.9999 | 10 ⁻⁵ | Controller routine communication in a 5 nmi radius environment – data |
| RCP 120 | 120 | 0.999 | 0.9999 | 10 ⁻⁵ | Controller voice intervention supporting separation assurance in a 15 nmi radius environment |
| RCP 240 | 240 | 0.999 | 0.999 | 10 ⁻⁵ | Controller voice intervention supporting separation assurance in a 30/50 nmi radius environment |
| RCP 400 | 400 | 0.999 | 0.999 | 10 ⁻⁵ | Controller voice intervention supporting separation assurance outside a 30/50 nmi radius environment |

*This is actually the probability of a loss of integrity during an arbitrary flight hour.

Fig. 32.5 ICAO guidance on RCP for controller/pilot voice and data communication

information related to some, but not all, of the data flows associated with UA CNPC. The reader may note that continuity and availability levels are relatively high (typically 0.999–0.9999 and even 0.99998 for voice availability in an RCP 10 environment associated with a 10-s transaction expiration time) but that the transaction times are on the order of tens to hundreds of seconds. This implicitly allows for some “soft failures” where a link may be interrupted or interfered with for a period of time but then recovers.

RTCA Special Committee 203 (RTCA/SC-203) has been working to identify additional requirements, beyond those identified by ICAO, specifically necessary to support UA CNPC data links. Final consensus has not been achieved, but some preliminary results are available in working papers within the committee and can be reported here. Readers are cautioned that further refinement may occur.

With regard to controller/pilot voice and data communications for UA, RTCA/SC-203 Working Group 2 has tentatively determined that the technically achievable round-trip latency for UA is likely to be up to a second longer than the technically achievable round-trip latency for manned aircraft. For ATC *voice communications*, the current capability for manned aircraft is about 770 ms (95 %) according to RTCA (2003). An additional second of delay for UA will still easily satisfy the ICAO guidance summarized above in Fig. 32.5. However, the situation for ATC *data communications* is less clear. In 1995, the FAA produced a report entitled Air Ground Data Link VHF ACARS Preliminary Test Report (Rehmann and Mestre 1995). The report concluded that the average round-trip message delay associated with ACARS (an existing air/ground data link system used by airlines and ATC) fell within the range of 10–20 s, with 5 out of approximately 2,300 messages lost. While the authors noted that Aeronautical Radio Inc. (ARINC), the operator of the ACARS, did not endorse the tests performed, these results would fall short of the most stringent ICAO guidance for data communications, even without an additional second of delay. Luckily, it is possible to improve the observed performance of ACARS with an enhanced (or a new) data link system. The working group has recommended further analysis to validate the ability to meet controller/pilot data link requirements for future UA operations.

While controller/pilot data link communications represent an obvious issue for future UA operations, the real driving consideration for RCP, in the context of UAS, appears to be the interaction between a pilot and his or her UA in a time-critical safety-related scenario – specifically, a loss of separation between two aircraft flying in opposite directions on a collision course (see Fig. 32.6).

In the case of an unmanned aircraft flying beyond visual range of the UACS and without the benefit of an autonomous collision avoidance system, the UA must sense the intruder, alert the pilot through a reliable return link, and respond to pilot forward link commands in order to avoid the collision. The detection of an intruding aircraft, and determination of a collision threat, could be based on TCAS, ADS-B, TIS-B, radar, or even visual detection with a suitable processor and software. The critical data flows are the telemetry containing the target track for the intruding aircraft and the telecommand containing the pilot's command for the avoidance maneuver.

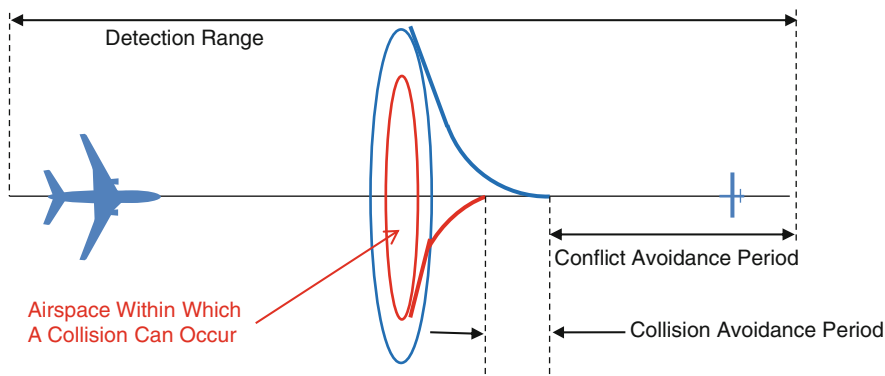


Fig. 32.6 Collision avoidance scenario (RTCA 2011)

The entire timeline for this collision avoidance scenario spans only a few tens of seconds. This places high demands on communications link availability, continuity, and latency. RTCA/SC-203 has addressed this scenario with a two-pronged analytic approach:

- (a) First, identify RCP performance levels that appear adequate to achieve the same levels of safety as demonstrated by manned aircraft today.
 - (b) Second, develop a strawman communications architecture, with sufficient engineering detail, to verify that these performance levels can be achieved in practice.
- As a preliminary matter, the working group recognized that the short timeline for the collision avoidance scenario places a stringent upper limit on communication latency. Simulations indicate that round-trip delays beyond a few seconds (exclusive of pilot reaction time) can adversely affect the outcome. Furthermore, the working group has recognized that the round-trip communication transaction time cannot be less than roughly 500 ms for a LOS communications system, or 1,100 ms for a BLOS communications system, due to the inherent message assembly and transmission times and propagation delays (RTCA 2009a). Hence, there is a strong incentive to deliver each message reliably on the first attempt. While message repetition (multiple transmissions separated in time) is not precluded as a viable communications technique, it should not be the primary means employed to achieve the necessary levels of link reliability. A precise latency requirement has not been identified, but the working group is proceeding under the assumption that the round-trip latency requirement for 95 % of the transactions must be on the order of 2 s or less.

Turning to availability and continuity, two methods were used to identify total probability of message delivery, which could then be used to derive RCP performance levels for availability and continuity (RTCA 2011). Method #1 relied on historical midair accident rates for general aviation (Part 91) and transport category aircraft (Part 121). It then applied assumptions relating to the likelihood of experiencing a potential midair collision scenario and the likelihood that a pilot could avoid a collision by observation and piloting. The UA CNPC data link must

have sufficient performance to not significantly degrade the pilot's ability to avoid the collision. Method #2 relied on the probability of catastrophic failures noted in FAA/AC 23.1309 for Class I and Class III aircraft. Class I (single reciprocating engine, gross weight <6,000 pounds) was used as a surrogate for smaller UA (Part 91 equivalent operations in Method #1). Class III (single or multiengine, reciprocating or turbine, gross weight >6,000 pounds) was used as a surrogate for large UA (Part 121 equivalent operation in Method #1). These two methods resulted in comparable performance values for total probability of message delivery: 99.8 % for smaller and medium UA equivalent to general aviation aircraft operating under Part 91 and 99.999 % for larger UA equivalent to transport category aircraft (Part 121 operations). These are stringent values for a radio communication link – not easily achieved without careful engineering. Furthermore, the reader's attention is drawn to the fact that larger UA have substantially more challenging requirements than smaller UA.

The RCP performance parameters of availability and continuity may be traded off against one another to some degree but must be set so as to jointly satisfy the required overall delivery probability for a particular class of aircraft (either 99.8 or 99.999 %), keeping in mind that

$$\text{Pr}\{\text{delivery}\} = (\text{availability}) (\text{continuity}).$$

With regard to message integrity, ICAO has already identified a target level for ATC/pilot voice and data communications as noted above in Fig. 32.5. This level is no more than one undetected message error per 10^5 flight hours. This can be converted to a per-message probability of undetected error (an undetected message error rate, or UMER) by considering the maximum number of messages that can be delivered over a span of time. For example, if the eventual UA CNPC data link supports 25 messages per second, there would be the potential for 90,000 messages per hour and required UMER = $1.1 \times 10^{-10} \sim 10^{-10}$. This level of integrity may also apply to relatively noncritical pilot/UA control interactions such as routine telemetry and some classes of telecommand which do not have an immediate impact on flight operations. However, this conjecture requires validation.

There is also the question of integrity for pilot/UA interactions that are associated with extremely time-critical safety messages (such as track reports for potential intruder aircraft) and uplink commands that affect aircraft trajectory (such as avoidance maneuvers) or changes to flight plans stored in the UA. These have not been addressed by ICAO, but the RTCA/SC-203 has associated these messages with a potential catastrophic loss in the event of an undetected message error and tentatively assigned a required integrity level of no more than one undetected message error per 10^9 flight hours for larger UA. This is based on the historical midair collision rate of 2.5×10^{-9} per flight hour for Part 121 operations, the threshold level of catastrophic failure for Class III aircraft of 10^{-8} per flight hour from FAA AC 23/25.1309, and the likelihood of a catastrophic hazard (10^{-9}) as listed in FAA (2008). As with the integrity level for ATC voice and data communications, this can be converted to a UMER by considering the number of

message opportunities per hour. If the eventual UA CNPC data link supports 25 messages per second, the derived UMER = 10^{-14} .

It should be noted that the analytic framework and RCP performance values and bounds described above are tentative and have not yet been validated by any regulatory body. The ITU based its analysis of bandwidth needs on three classes of UA (small, medium, and large), and these bandwidth requirements are currently being used to identify suitable RF spectrum resources. In slight contrast, the RTCA has tended to analyze RCP performance requirements relative to just two classes – a class of “smaller” UA which is somewhat related to the small UA of the ITU but is more directly associated with performance requirements derived from those of general aviation (Part 91 operations) and a class of “larger” UA which is somewhat related to the medium and large UA of the ITU but is more directly associated with performance requirements derived from those of transport category aircraft (Part 121 operations). The ultimate classification of UA and UAS operations, for regulatory purposes, will be agreed within ICAO based on consensus among the world’s civil aviation authorities; this work has yet to be completed. The performance requirements noted above should be viewed as *likely representative* of the range of requirements that may ultimately be applied to various classes of UA, recognizing that the specific classifications, breakpoints between classes, and associated requirements are subject to refinement and validation.

32.3 Frequency Bands for CNPC Communications

As noted in ITU-R (2010), current UAS use a wide range of frequency bands for control of the UA in segregated airspace. Systems operate on frequencies ranging from VHF (72 MHz) up to Ku-band (15 GHz) for both line-of-sight (LOS) and beyond-line-of-sight (BLOS). The factors driving the choice of frequency are related to limiting the size, weight, and power of the airborne data link equipment – particularly antennas and power amplifiers – as well as satisfying the data rates required. Many LOS systems operate in frequency bands allocated to the mobile service, or the aeronautical mobile service, but are not accorded extra protection as a safety service such as bands allocated to the aeronautical mobile (route) service (AM(R)S) or the aeronautical mobile satellite (route) service (AMS(R)S). Furthermore, many LOS and BLOS systems share the control link and the payload return link on a common carrier. This is desirable from the standpoint of minimizing radio size, weight, and power but mixes different data types in a way that would not be allowed in safety spectrum.

In the future, considering that UA will operate in civil nonsegregated airspace, and that the UA CNPC data link will help ensure the safety of life and property, the RF spectrum employed for CNPC will need to be designated as a safety service. Figure 32.7 lists some existing aeronautical bands, between 100 and 5,150 MHz, which were examined by RTCA/SC-203 for UAS. Other aeronautical or aeronautical-related bands exist but were excluded from this list because either (a)

| Frequency Band | Allocated Use (typical systems) |
|-------------------|---|
| 108 - 112 MHz | ARNS (ILS LOC) |
| 112 - 117.975 MHz | ARNS (VOR, GBAS) |
| 117.975 - 137 MHz | AM(R)S (VHF voice and data) |
| 960 - 1215 MHz | ARNS (DME, TACAN, Radar Transponders) |
| 1525 - 1559 MHz | AMS(R)S |
| 1610 - 1660 MHz | AMS(R)S |
| 2900 - 3100 MHz | Radiolocation, Radionavigation |
| 5000 - 5150 MHz | ARNS (MLS), Radionavigation Satellite, AMS(R)S, FSS (but limited to MSS feeder links) |

Fig. 32.7 Potential aeronautical or aeronautical related bands for UA CNPC data links

they are already allocated to systems which would not be suitable for sharing with UAS or (b) they are at frequencies inappropriate for UAS use. After considering numerous factors including existing congestion in each band, potential for sharing, available bandwidth, link range, capacity, propagation characteristics, and air and ground cosite compatibility (among others), the list was narrowed to two candidates: portions of the 960–1,024-MHz ARNS band for a LOS system; and the 5,030–5,091-MHz band for either a LOS or BLOS system (or both) (RTCA 2009b). While the 960–1,024-MHz subband currently lacks an AM(R)S allocation as would be required for a LOS system, this was viewed as achievable from a regulatory standpoint. As for the 5,030–5,091 MHz band, the only incumbent user is the microwave landing system (MLS) which is deployed in very limited numbers. Technical analyses indicated that both LOS and BLOS (satellite) CNPC systems could effectively share the band with MLS.

While other candidate bands could potentially be used on a case-by-case basis, they would not be suitable for meeting any significant portion of the projected bandwidth requirements of the UAS community (34 MHz of terrestrial spectrum and 54 MHz of satellite spectrum).

In a similar vein, the ITU-R has identified portions of the 960–1,164 and 5,030–5,091-MHz bands as good candidates to support control links for UAS without causing harmful interference to incumbent services and systems (ITU-R 2010). As noted by the ITU, the 960–1,164-MHz band is highly favorable for UAS control links. Rain losses are negligible and free-space path losses are low enough to permit reliable long-range LOS communication between relatively low-power radios using omnidirectional and medium-gain antennas. While much of the band is heavily used, substantial subbands (960–976 and 1,151–1,156 MHz) are not used by airborne NAVAID transmitters and contain no fixed ground-based assignments in some countries. In such countries, 10.4 MHz of spectrum could be made available, which would furnish small UA with badly needed access to protected spectrum and provide

UA of all types with band diversity that is essential for reliable UA CNPC. While the identified spectrum (10.4 MHz) is less than the 34 MHz identified as needed for terrestrial/LOS links, it “would suffice to meet all UAS CNPC requirements except for backup links, video, and downlinking of airborne weather-radar data in some countries.”

RTCA/SC-203 has also noted (RTCA 2009b) that Inmarsat and Iridium (satellite systems operating in the 1.5-/1.6-GHz bands) could be considered as existing BLOS systems able to support UA CNPC links. Inmarsat operates in a band with a clear AMS(R)S allocation, has a total bandwidth of 14 MHz available on a worldwide basis, and is suitable for BLOS UA CNPC. Iridium also operates in an aeronautical band and has been used for BLOS UA CNPC by at least one UAS. But its first-generation system has rather limited per-channel bandwidth which limits its suitability for the full range of UAS (however, this may be overcome in its second-generation system).

RTCA/SC-203 has also considered the potential for a new AMS(R)S allocation at either Ku-band or Ka-band that could support BLOS UA CNPC with a wide-band geostationary satellite system. At Ku-band, the 12/14 GHz commercial fixed satellite service (FSS) band and the 13.25–13.4 and 15.4–15.7-GHz bands are all attractive. The latter two bands are allocated to the aeronautical radio navigation service (ARNS) on a primary basis, although the lower band is limited to Doppler navigation systems and both bands have other coprimary allocations. Nevertheless, the existing ARNS allocation indicates the potential to add another aeronautical safety service such as AMS(R)S. Similarly, the 19.7–20.2 and 29.5–30.0-GHz bands (Ka-band) appear potentially suitable for UAS BLOS CNPC links, with the upper 50 MHz of each of these bands appearing to be particularly attractive given other existing users, allocations, and interference considerations. SC-203 noted that “The close proximity between these bands and those used by Ka Band equipped military UAS makes it possible for the military to use the same equipment in both segregated and non-segregated airspace by just tuning to different frequencies without the need to change antennas or modems. This is a very attractive feature of the proposed K/Ka band segments” (RTCA 2009b).

As is clear from the preceding discussion, it is challenging but not impossible to assemble the necessary spectrum resources for UAS CNPC data links. Of course, a single UAS could operate within the currently allocated bands using less than a MHz of spectrum, and the first several UAS deployed would not perceive any bandwidth constraints. However, uncontrolled or uncoordinated proliferation of UAS with differing CNPC data link formats, protocols, and waveforms could lead to relatively inefficient use of the spectrum and the potential for unsafe conditions brought about by interference. Therefore, there is a strong incentive to define a common data link standard that could efficiently support the entire community. The work of defining such a standard is likely to take place initially in various advisory bodies such as RTCA and EUROCAE, with an eventual standard being finalized by consensus within the International Civil Aviation Organization (ICAO).

32.4 Technical Challenges for CNPC Communications

As noted previously, the required communications performance (RCP) thresholds for UA CNPC data links are stringent. The information flows for sense-and-avoid and pilot control, in particular, must be extremely reliable, have low latency, and have very high integrity, in order to ensure safety of flight. While numerical requirements have not yet been formally adopted, the emerging consensus at RTCA indicates that availability and continuity must satisfy message delivery probabilities on the order of 99.8–99.999 %, latency must be on the order of 1–2 s, and undetected message error rates (integrity) for at least some information flows (for some UA) must be on the order of 10^{-10} – 10^{-14} per message. In order to meet these requirements, the system designer must address hardware failures that could permanently disable a link. Hardware failures primarily affect link availability and continuity (not latency or message integrity) and can be mitigated through careful design and system redundancy using well-understood techniques. These will not be further discussed. In addition, the system designer must address “soft failures” such as airframe shadowing, multipath fading, rain attenuation, and random bit errors introduced on the channel, which lead to short-term outages and potential undetected errors. These short-term outages are actually the dominant concern for the most stringent RCP thresholds identified. For example, airframe shadowing or excess rain attenuation, which persists for several seconds, could prevent a pilot from recognizing a safety threat and taking appropriate and timely action. The key impairments, of interest to the UA system designer, are addressed below in the context of the “link budget” which provides a quantitative framework for understanding link performance.

The Link Budget as a Framework for Understanding Soft Failure Issues. All radio communication links operate by transmitting energy in a particular physical format (e.g., a “waveform”) from the transmitter to the receiver. Generally, for digital data links, one or more particular parameters of a sinusoidal radio wave are adjusted among a finite number of possibilities from one transmitted symbol to another. For example, the amplitude, frequency, or phase of the sinusoid might be adjusted. These alternatives generally go by the names of amplitude shift keying (ASK), frequency shift keying (FSK) and phase shift keying (PSK). The number of different symbols that can be sent is usually a power of 2; for example, a system with two possible phase states is called binary PSK or BPSK. A system with eight possible frequencies is called 8ary FSK or 8-FSK. Literally scores of different waveforms have been invented over the years. However, all of them involve the transmission of energy from a transmitter to a receiver. The desired signal is generally received at a very low signal level (picowatts, or even less) and in the presence of natural and man-made RF noise from the environment, as well as RF noise generated internal to the receiver itself. This RF noise tends to corrupt the desired signal and prevent accurate decision-making as to the symbol or symbols actually sent by the transmitter. Hence, a primary task of the communications engineer is to ensure that the amount of received energy in the desired signal, during a given symbol

interval or set of symbol intervals, is sufficient to allow accurate reception of the vast majority of transmitted messages.

The analytic framework used to predict reliable reception is called the “link budget.” A link budget is similar to a household financial budget, where one starts with monthly take-home pay or revenue, subtracts known fixed expenses as well as an allowance for variable expenses, and checks to see if everything can be covered – hopefully with a little extra “margin” that can be saved for a rainy day (i.e., when variable expenses turn out to be larger than expected). In the case of an RF link budget, one starts with the power transmitted by the transmitter, accounts for known fixed losses as well as an allowance for variable losses, and checks to see if there is sufficient power at the receiver (or, in other words, sufficient energy in each received symbol or set of symbols) to allow reliable reception given the expected RF noise – hopefully with a little extra “margin” that can be used if losses are greater than expected or the transmitter is somewhat weaker than expected.

| Parameter | Units | Uplink | Downlink |
|--|-------|--------|----------|
| Transmit power | dBm | 40 | 40 |
| Transmit antenna gain ¹ | dBi | 8 | 0 |
| Cable and filtering loss | dB | 1 | 1 |
| Transmit EIRP | dBm | 47 | 39 |
| FSPL (1 GHz, 75 nmi) | dB | 135 | 135 |
| Additional losses ² | dB | 23 | 23 |
| Receive antenna gain ¹ | dBi | 0 | 8 |
| Cable and filtering loss | dB | 1 | 1 |
| Received signal power | dBm | -112 | -112 |
| Receiver noise power | dBm | -126 | -126 |
| Available Carrier-to-noise ratio (C/N) | dB | 14 | 14 |
| Theoretical C/N | dB | 6 | 6 |
| Implementation losses | dB | 2 | 2 |
| Safety margin | dB | 6 | 6 |
| Required C/N | dB | 14 | 14 |
| Excess margin ³ | dB | 0 | 0 |

¹Variability of UA antenna gain due to airframe blockage or reflections is taken into account in the airframe loss part of the “additional losses” row.

²Additional losses comprise atmospheric losses, airframe losses and multipath losses. These are convolved together and the quoted value provides for 99.8% link availability (0 dB margin, 99.8% of the time).

³With double or triple ground diversity, the excess margin could increase by 9 to 14 dB.

Fig. 32.8 Simplified link budget for line-of-sight UA CNPC data link at L-band (Based on RTCA (2011) and Wilson (2011))

A simplified sample link budget, showing the major categories of fixed and variable losses that must be considered for a LOS UA CNPC link, is illustrated in Fig. 32.8. This link budget is used here only as an example – many of the values are particular to the specific design and assumed operating scenario; other assumptions as to operating band, waveform, and operating scenario, would lead to different numbers. Also, a BLOS link budget must consider four links instead of just two (the UA uplink and downlink to the satellite and the ground station uplink and downlink to the satellite). Finally, the reader should note that the various gains and losses are multiplicative, but the link budget converts everything to a scaled logarithm called a “decibel” (dB). This allows the gains and losses to be added and subtracted as in a financial budget.

Looking to the figure, the first line is the actual transmitter power in dB (milliwatts) or dBm. The numerical value of 40 dBm is equivalent to a 10 Watt transmitter. This link budget assumes that both the ground-based radio (for the uplink) and the airborne radio (for the downlink) have the same transmitter power. The second line is the antenna gain. The ground antenna is assumed to have a moderately directive antenna (8 dB relative to isotropic, or 8 dBi) which “focuses” the RF energy into a beam with roughly 8 dB more power than if the energy were radiated equally in all directions (8 dB is roughly a factor of 6:1). The airborne antenna is assumed to be nominally isotropic, so it has no directive gain relative to an isotropic antenna. However, the airborne antenna is mounted on the aircraft and can easily be shadowed by the airframe. As a result, the actual gain experienced from one moment to the next can vary significantly from this value. Generally it will be worse. This is accounted for in the “additional losses” line below. The third line accounts for transmitter cable and filtering losses. Finally, the fourth line accumulates the first three into a “figure of merit” for the transmitter – its “effective isotropic radiated power” (EIRP). This is used for the subsequent calculations.

The free-space path loss (FSPL) row accounts for the weakening of the RF signal as it propagates from the transmitter to the receiver. For a radio link at roughly 1-GHz (L-band), and a 75-nmi path, the loss is 135 dB. This is a nominal value assuming no big obstructions and no refractive losses. Further data and calculation methods are provided in ITU-R (2007). For UA operating at long range and low altitude (expected to be a common scenario), the design engineer should specifically account for diffraction losses which can add on the order of 15–20 dB of loss for an aircraft close to the radio horizon (ITU-R 2007; CCIR 1991).

The “Additional losses” row accounts for several time-varying effects which are critical to an understanding of the UA CNPC data link design. These include airframe shadowing, antenna variability, multipath fading, and atmospheric and rain attenuation (especially relevant at 5 GHz and higher). Since these different effects are physically independent, a statistical analysis is needed to determine the “allowance” that must be provided in order to guarantee a given level of performance. For the sample link budget provided here and based on other data and design assumptions as discussed below (including at least two antennas on the aircraft but a single ground antenna), a convolution of the probability distributions indicated that the combined losses would be no worse than 23 dB, 99.8 % of the

time. Thus, a CNPC data link attempting to achieve an availability of 99.8 %, with these assumptions, should allow for 23 dB of variable losses.

The next three lines account for the receive antenna gain, cable and filtering losses at the receiver, and final received power level (dBm) at the input to the first-stage amplifier, assuming the variable losses noted above were at the threshold value. The aircraft antenna gain (0 dBi) is again a nominal value that does not account for the expected variation already handled in the “additional losses” row. The assessed received signal power is -112 dBm (6×10^{-15} W).

The “receiver noise power” line accounts for ambient and receiver-internal RF noise across the channel bandwidth of the receiver. This is usually based on measured data for the environment and measured or predicted data for the receiver equipment chain including the antenna, cables, filters, mixers, and first-stage amplifier (at least), referenced to the same reference point as the received carrier power (in this case, the input to the first-stage amplifier). The sample link budget indicates an equivalent noise power of -126 dBm.

The difference between the received signal power, or carrier power (C), and the receiver noise power (N) is the “carrier-to-noise power ratio” (C/N). This value (14 dB for the sample link budget) must be large enough to allow for reliable operation, allowing the messages to be received without error (at least most of the time). Note that C/N refers to the integrated power (for the desired signal and the noise) over the receiver channel bandwidth. Another important metric is ratio of the energy in a single transmitted symbol or bit (abbreviated E_s or E_b) and the noise power density per Hz of bandwidth (abbreviated N_0), called E_s/N_0 or E_b/N_0 , respectively. All of these metrics represent a “signal-to-noise ratio” (SNR), but because of the definitional differences, any reference to “SNR” should be precisely defined as one of these particular metrics for a particular application or calculation.

The “theoretical C/N ” is the C/N for a particular waveform and data link protocol (with its synchronization features, authentication, and forward error correction) needed to achieve the desired message error rate, before accounting for implementation losses (hardware imperfections). For the waveform and forward-error-correction coding scheme assumed here, this value is 6 dB. To this theoretical C/N value, it is necessary to account for the receiver implementation losses (such as improper timing, inaccurate carrier tracking, amplifier nonlinearities, filter distortion) and also allow for a “safety margin” accounting for all additional effects and link impairments beyond the ability of the engineering team to predict. In this case, the combination of the theoretical C/N , implementation losses, and safety margin ($6 + 2 + 6 = 14$ dB) exactly matches the available C/N assuming variable link losses at the 99.8 % availability level. Hence, the link closes 99.8 % of the time as required, taking into account all the known variables (the engineering team could say they succeeded in the link design), but there is no “excess margin” beyond the safety margin.

Considering the link budget as a whole, it is clear that the “additional losses” line is a key driver. The primary physical factors contained in this line are now addressed.

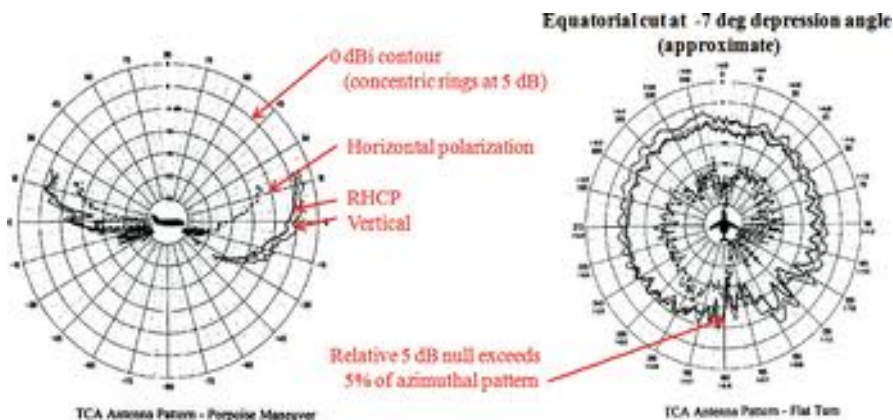


Fig. 32.9 Typical top-mounted antenna performance on a mid-sized aircraft (RTCA 2011) (Ref: Colby et al. 1996)

Airframe Shadowing and Installed Antenna Performance. It is convenient – but not accurate – to assume that an omnidirectional antenna mounted on an aircraft will actually deliver uniform gain in all directions (or even uniform gain over one hemisphere, such as an upper hemisphere for a top-mounted antenna or a lower hemisphere for a bottom-mounted antenna). Figure 32.9 illustrates typical performance for a mid-sized manned aircraft with a top-mounted, nominally omnidirectional L-band antenna (either horizontal, vertical, or right-hand circular polarized) operating at 1,575.42 MHz, relatively close to the potential new AM(R)S allocation around 1 GHz currently being considered for UAS CNPC data links.

The left-hand panel of Fig. 32.9 indicates significant gain variation within 15° of the local horizontal plane and very poor performance (as much as 25 dB below nominal) below the aircraft. For an aircraft that is a significant lateral distance from the serving ground station, any bank “away from” the ground station would expose the bottom of the aircraft to the ground station and create a significant drop in antenna performance. The right-hand panel (measured at a 7° depression angle roughly associated with long-range communications to a ground station, for an aircraft in straight-and-level flight) indicates that typical gain is roughly -5 dB but typically varies by several dB for different azimuth angles relative to the aircraft centerline. Furthermore, more than 5 % of the azimuthal pattern exhibits an additional 5 dB loss, and there are narrow nulls (e.g., exactly off the tail) where antenna gain is as much as 10 dB worse than nominal (if “nominal” is taken to be -5 dB relative to an ideal isotropic antenna). Clearly, these variations must be accounted for if the system is intended to achieve 99.8 % or greater link availability.

The loss in performance (as much as 25 dB in this case) can be partially mitigated by installing, and relying on, two or more CNPC data link antennas strategically placed on the UA. For example, top, and bottom-mounted antennas, or antennas located at each wingtip, can be used to ensure that the worst-case losses are never

experienced simultaneously by both (or all) antennas. Multiple antennas on the ground can also be used – particularly if the UA CNPC architecture relies on a shared nationwide infrastructure with multiple radio cells (sectors) each with its own antenna. For three widely separated ground antennas surrounding a UA, spanning more than 180° of arc as perceived by the UA, at least one ground antenna will always have a nominally unobstructed view to at least one antenna on the aircraft. These techniques are both called “antenna diversity” or “space diversity” (because one node or “station” has two or more antennas at different locations in space). The improvement due to antenna diversity is illustrated in Fig. 32.10 for a particular UAV with a carbon composite body and a 3-m wingspan. The upper panel is for a single L-band antenna installed in a winglet (i.e., at the end of a wing, in an RF-transparent housing). The five traces are for different elevation or depression angles relative to the local horizontal, but the key point of this plot is the worst-case performance for a single antenna, experienced at any angle, at roughly -28 dBi. This would imply a required link margin of at least 28 dB, just for this impairment alone, relative to a link designed for a nominal 0-dBi antenna.

The lower panel indicates that diversity antennas can reduce the required margin to about 13 dB.

If multiple antennas are available on the ground as well, arranged so as to avoid a condition where all of them are on a “line of symmetry” relative to the airborne antennas, or all of them can view only one of the airborne antennas (this requires at least three widely separated ground antennas surrounding the aircraft and spanning more than 180° of arc as seen from the UA), the margin requirement can likely be reduced to roughly 10 dB or less (RTCA 2011). This is an 18-dB improvement over the no-diversity case.

Multipath Fading. Multipath refers to portions of the transmitted signal which are reflected off of objects or terrain in the environment and which arrive at the receiving antenna nearly “out-of-phase” relative to the direct signal. This can significantly reduce the received signal strength. Close to complete cancellation (fading) can occur for short periods of time. Figure 32.11 illustrates typical fading on an L-band air/ground data link for an aircraft flying at 500 m, communicating with a ground station antenna 25 m above ground level.

The large-scale lobing evident in Fig. 32.11, with signal strength varying by 10–20 dB, is due to so-called ground-bounce multipath and is relatively predictable for a particular environment and flight profile (as indicated by the red trace which is a mathematical model) (ICAO 1999). The fine-scale fading is due to other effects such as reflections off of individual terrain features, buildings, small bodies of water, and the airframe itself. As with the antenna effects illustrated earlier, these multipath fades must be accounted for if the system is intended to achieve 99.8 % link availability or greater. ITU-R (2007) provides a general method to estimate the probability of fade depth for multipath fading, but does not provide any framework to assess correlation time (the time that the fade depth is relatively constant). This turns out to be a significant factor for CNPC link availability and continuity. Also, while there is a fair amount of experimental data for air/ground data links at VHF, there is relatively little experimental data for air/ground data links (on flight

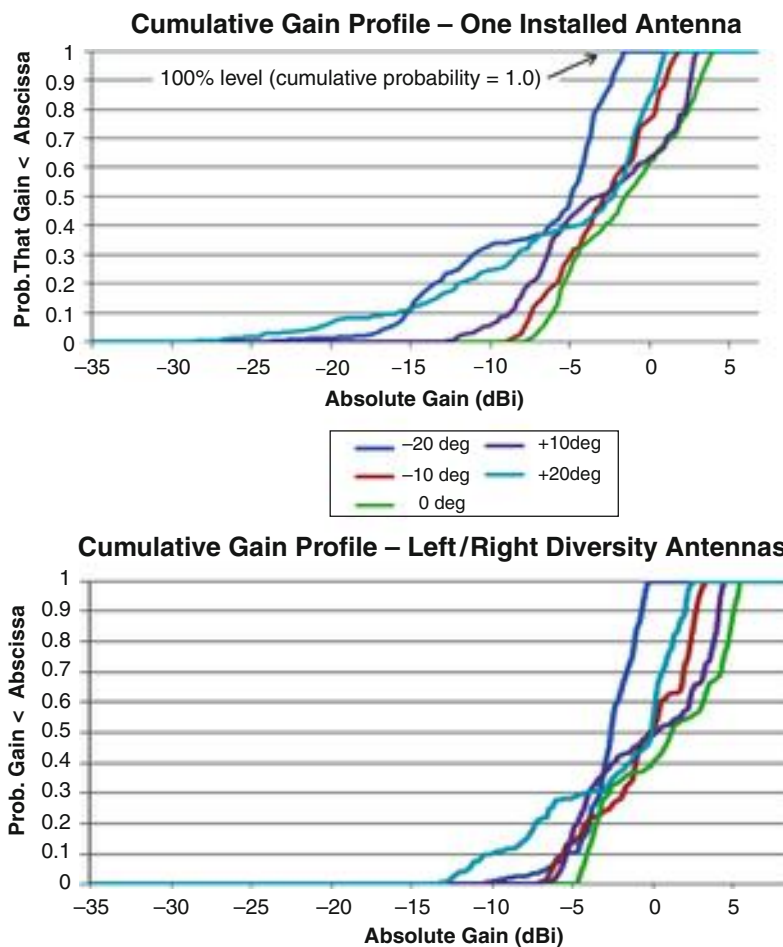


Fig. 32.10 Antenna performance with and without diversity

profiles that would be representative of typical UA operations) at L-band and above. While some estimates of expected UA CNPC link performance can be made by extrapolating from lower frequencies and compensating for known differences at the higher frequencies, further work is needed in this area.

Lacking precise information on the correlation time of multipath fading for typical UA operating scenarios at the expected operating frequencies, one can nevertheless make a first estimate of required fade margin by calculating the cumulative fade distribution for a given environment, using approved techniques as described, for example, in ITU-R (2007). Typical values at L-band are 18 dB of fading (at the 99.8 % availability level) for an aircraft at 75-nmi range and 10,000-ft altitude, and a ground antenna at 100 ft, and 22 dB of fading (for 99.8 % availability) if the aircraft is at 5,000-ft altitude. At shorter range, for example, 30 nmi, multipath

fading may be on the order of 10 dB less severe – but this is still a significant impairment.

As with airframe and antenna variation, multipath effects can be partially mitigated by multiple data link antennas on the aircraft or on the ground (or both). Widely separated ground antennas can partially mitigate ground-bounce multipath (the large-scale variation in Fig. 32.11), and antenna diversity on the ground as well as the aircraft can partially mitigate the short-term fluctuations simply because the reflected signals are unlikely to arrive at all the available antennas with the same relationship of amplitudes and phase shifts. For example, with two data link paths exhibiting uncorrelated statistics, the pair will deliver 99.8 % link availability if each individual link is engineered to provide roughly 95 % link availability. At long range and low altitude, this can reduce necessary fade margins by roughly 10 dB (e.g., to the range of 7–10 dB). Additional diversity paths provide additional gain, although improvements quickly become marginal.

Furthermore, if multiple messages are sent in quick succession (i.e., within the correlation time of the fading, so that several messages in succession might be corrupted), a change in channel operating frequency from one message to the next can frequently provide for independent fading statistics. This can be beneficial in systems that rely on selective retransmission (or even multiple transmissions for all messages) to enhance overall link availability. Preliminary evidence and analysis indicate that a slow frequency-hopping system operating within the contiguous bandwidth expected for a LOS system (i.e., on the order of 10 MHz or greater) will provide the necessary statistical independence from one message to the next.

Atmospheric and Rain Attenuation. All radio signals are attenuated by atmospheric gases, water vapor, and liquid rain. At L-band and below, this attenuation is generally very small and can be ignored or compensated by a relatively modest margin allowance. However, at 5 GHz and above and especially at Ku- and Ka-band

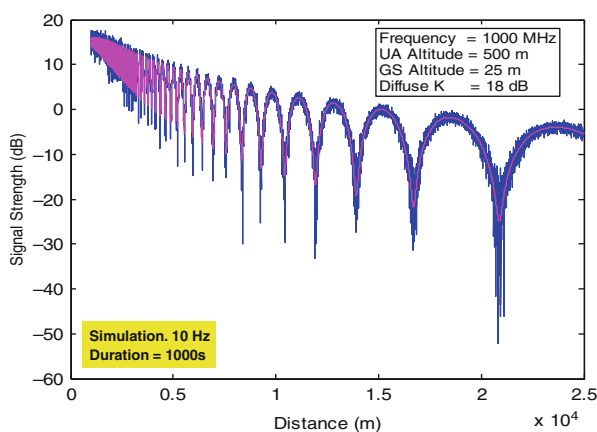


Fig. 32.11 Typical multipath fading (See also ICAO 1999 for a comparison of measured and modeled fading on typical air-ground VHF data links)

frequencies currently under consideration for at least some BLOS links, both clear sky and rain attenuation can be significant. Of course, it does not rain all the time (the worldwide average is only about 5 % at a random location), and some aircraft might actually fly above the rain layer for most of their mission. But for aircraft flying in the rain (e.g., during takeoff and landing and for low-altitude operations) and using a data link frequency of 5 GHz or higher for safety communications, link outages caused by rain must be explicitly considered. There are several widely used models for attenuation caused by atmospheric gases, water vapor, and rain. ITU-R (2007) offers a good starting point; this ITU recommendation draws on several other ITU recommendations and provides a good road map for analysis. Generally speaking, atmospheric gases contribute a fairly predictable and constant attenuation for a given operational scenario, whereas the rain models provide a statistical understanding of the percentage of time that a particular attenuation level is exceeded. This directly affects the assessment of link availability. Surprisingly, it turns out that continuity (the problem of a link failing *during* a message transmission period) can be substantially ignored at the availability levels (and probability of message delivery) relevant to UA CNPC. This is because rain fades tend to persist for tens of seconds or longer, with fade levels changing significantly only over seconds (not milliseconds) as a consequence of the physics of rain. Specifically, the volume of space, through which the RF signal travels, can be thought of as an extended tube or ellipsoid which is partially filled and emptied by the rain. An intense rain burst will tend to “fill” the radio path with water, whereas a lull in the rain will tend to allow the radio path to empty out. Because raindrops fall at a characteristic rate and because the radio path has a significant length and girth, it takes a finite amount of time (typically measured in seconds) to significantly change the amount of liquid water contained in the tube or ellipsoid.

Antenna diversity (space diversity) is generally not as effective at combating rain attenuation on BLOS links since the rain fade on two data links supported by a single aircraft tends to be correlated, even if multiple satellites are used for these two data links (i.e., two antennas on the UA pointing to different satellites).

Convolution of Time-Varying Effects. As noted earlier, once the characteristics of antenna performance, airframe shadowing, multipath fading, and rain attenuation are understood for a particular LOS or BLOS CNPC architecture and operational scenario, the four effects should be treated as statistically independent and convolved to derive an aggregate distribution of fade probability, to avoid an overly conservative link design. For the case of a typical L-band CNPC data link operating an LOS path over 75 nmi, intended to achieve an overall availability of 99.8 % with dual-link diversity, these separate effects would have a combined margin requirement in excess of 30 dB. In contrast, a proper statistical analysis indicates that the actual requirement is only 23 dB. The reader is cautioned that these numbers are illustrative, and particular UA operating scenarios should be separately validated to ensure that required link performance can be achieved.

Lost-Link Considerations. Despite all the engineers’ best efforts, there is always a possibility that the CNPC data link (including all its diversity paths) will be

permanently lost. This could affect the telemetry link or the telecommand link or both. Potential causes are: hardware failures on the UA, hardware failures at the UACS, hardware failures in the ground infrastructure, widespread jamming or interference, and pilot error (e.g., commanding the UA to fly beyond communications coverage). If ATC voice and data communications were being routed through the UA at the time of failure (instead of being routed directly between ATC and the UA control station through a wired ground network), the ATC data flows would be affected as well (note: this is another reason to favor the wired ground network, and only rely on ATC data flows relayed through the UA when no other alternative exists).

When a lost-link event occurs, procedural methods must be in place to ensure safety of flight for the affected UA, safety of flight for other aircraft in the airspace, and safety for people and property on the ground. There is as yet no worldwide consensus on how to handle a UA with a lost CNPC link. However, several reasonable procedures can be identified now based on currently available technology and operational procedures:

- First, “heartbeat messages” on uplink and downlink can be used to track link performance and quickly identify a lost-link event when it occurs (time out thresholds are currently TBD). If found to be operationally beneficial, summary statistics could even be echoed back to the other node (e.g., the UA could transmit down to the UACS, as part of its telemetry stream, the statistics of the heartbeat messages it observed on the uplink).
- If the UA detects a failed CNPC data link, it could autonomously set a unique radar transponder code which would serve to alert ATC, as well as other aircraft in the airspace, that it was suffering a lost link. This would be similar to the existing procedure for radio failure on a manned aircraft. A lost forward link is directly observable by the UA. A lost return link, which leaves the forward link intact, is also possible although less likely. This can be detected if the UACS is echoing back return link heartbeat statistics or if the UACS has the ability to send a dedicated command indicating a lost return link. Hence, it is always possible for the UA transponder to be properly set. Suitable transponder codes remain to be defined but are currently being considered. Special messages could also be broadcast via future (or upgraded) ADS-B systems.
- If controller-pilot communications are supported with a wired ground network, these data flows will likely remain intact even if the UA CNPC RF data link fails. Therefore, the pilot can inform ATC that the UA has suffered a lost link. Similarly, if the UA has reset the transponder code, this will be apparent to ATC, and the controller could then contact the pilot.
- The UA should have a default flight plan, stored in its onboard computer, which will be autonomously executed in the event of a lost CNPC data link. This flight plan might be updated from time to time during a particular flight but should in any case comprise standardized procedures (following guidelines preapproved by aviation regulatory authorities). For example, this might involve a short period following its current flight plan while broadcasting its condition via its

radar transponder and ADS-B transmitter (this allows ATC to clear the airspace immediately around the UA), followed by a climb to a preapproved altitude, followed by execution of a new flight plan such as “circle for X minutes waiting for link to be reestablished” or “return to base” or “fly to alternate recovery site” or “fly to previously identified remote area and self-destruct.” If the UA is equipped with ADS-B, the flight plan being executed could potentially be broadcasted via this medium so that ATC, and surrounding aircraft, would be aware of the UA’s intent. It should be noted that some UA operational today already implement such a lost-link flight plan strategy as part of their standard operational doctrine (absent the broadcast via ADS-B).

32.5 A Strawman Architecture for CNPC Communications

Potential architectures for UA CNPC communications have evolved within RTCA, the ITU-R, and other groups. The strawman architecture described in this section (the “strawman”) is one such solution based on the work in RTCA and is presented here simply to illustrate the types of solutions that could be envisioned for a future system. Other options are certainly possible and will need to be explored before a final conclusion can be developed into a consensus standard for the UAS industry to adopt.

With regard to frequency bands, the strawman incorporates LOS links at L-band and C-band and BLOS links at C-band and either Ku-band or Ka-band (or both). Precise band edges are TBD, but current international efforts indicate that it will be possible to assemble at least 34 MHz of LOS spectrum and 56 MHz of BLOS spectrum by relying on this approach. Note that C-band is shared between LOS and BLOS; the strawman assumes that the BLOS system will operate with uplink and downlink channels at the edges of the allocated C-band spectrum (likely the 5,031–5,090-MHz band) with a central gap between the two. This central gap is necessary to avoid self-interference by BLOS users who will be transmitting and receiving simultaneously. The gap is then filled with the C-band portion of the LOS subsystem. The exact split between the LOS and BLOS spectrum at C-band is TBD and could potentially vary from one geographic region to another. Alternative bands may also be used by individual UAS – for example, VHF for LOS connectivity and L-band (using either Inmarsat or Iridium) for BLOS connectivity. But the strawman concept does not focus on these alternatives.

The LOS subsystem is assumed to rely on airborne antennas which are nominally omnidirectional. However, as noted above, each airborne antenna is assumed to suffer significant shadowing and nulling due to its interaction with the airframe. Thus, at least two antennas and radios are assumed for even the smallest UA.

The transmitters on the aircraft, and the transmitters and antennas at the ground stations, must be sized to provide sufficient transmit power and antenna gain to close the link against the expected fixed and variable impairments as discussed above. Engineering trade-offs among these elements (i.e., the airborne and ground transmitters and the ground antennas) are expected within the framework of the

strawman design and the associated link budget analyses. Phased-array or “smart antenna” technology, with signal detection and characterization and multichannel beam-forming capability, is expected to be used in the ground stations in order to track and provide sufficient ground antenna gain to multiple UA at arbitrary locations in a given coverage sector. An additional advantage of this technology is that multipath fading on the downlink could potentially be eliminated or reduced to very small levels.

Diversity techniques are heavily used in order to meet anticipated (but not yet validated) RCP requirements. The UA population is assumed to be sub-divided into two groups based primarily on size but possibly including other factors such as speed and operating regime. The first group is comprised of “smaller” UA which must satisfy a message delivery probability of 99.8 % within 1–2 s in order to satisfy presumed safety requirements. The second group is comprised of “larger” UA which must satisfy a message delivery probability of 99.999 % within 1–2 s. The first group is expected to have relatively small CNPC data throughput requirements, is not expected to operate BLOS because of their small size, and may rely almost exclusively on L-band LOS links. At least two antennas and radios are assumed to exist on each of these smaller UA in order to provide support for the necessary level of diversity to achieve 99.8 % message delivery probability with feasible transmit power and omnidirectional airborne antennas (but even more antennas and radios may be installed, if required). The second group is expected to have larger CNPC data throughput requirements as well as more stringent RCP requirements will use either LOS or BLOS (or both) and will rely on a dual-band system offering at least four-way diversity. For example, a larger UA could rely on L-band and C-band for LOS links, with at least two antennas and radios in each band (a total of at least four antennas and radios), or some combination of C-band, Ku-band, or Ka-band BLOS links (for aircraft that are so equipped), so as to provide connectivity to at least two satellites simultaneously with each satellite link engineered to deliver at least 99.8 % message delivery probability (thereby achieving 99.999 % delivery probability for the diversity pair). Larger UA with BLOS capability are also expected to support LOS capability for short-range and low-altitude operation where terrestrial infrastructure exists to support them. This also partially mitigates the problem of overcoming rain attenuation at low altitude (i.e., for takeoff and landing operations), where the satellite link (if used) might encounter a significant amount of rain attenuation along the radio path.

With regard to LOS ground stations, a regional or nationwide deployment is assumed in order to provide basic coverage (similar to VHF voice radio and ATC radar today) as well as to provide adequate space diversity to meet expected RCP requirements. A notional deployment is illustrated in Fig. 32.12. In this concept, 12 frequency subbands are assigned to nominally hexagonal sectors and reassigned such that co-channel interference between sectors using the same subband is minimized (note: this is for sectors that have relatively high-altitude limits; a dedicated frequency plan for smaller UA might use a “four-color mapping” instead of a “twelve-color mapping”). Within the strawman concept, the total number of ground stations is equal to the total number of sectors (an actual deployment could

ground station pilot signals, the UA will associate itself with a particular sector and ground station (this could also be commanded by the ground system in some cases). This access process involves authentication of the UA in the system and assignment of a low-rate or high-rate channel (or subset of a channel) based on UA data link requirements. When a smaller UA is required to transmit, it selects the antenna/radio that has recently exhibited the best uplink performance from *any* of the ground stations in view, and transmits through this antenna/radio pair *on its assigned channel* (i.e., the one associated with its “primary” CNPC data link).

In a similar vein, each ground station monitors all traffic channels in the system – not only its “assigned subset.” This is a little different from cellular systems today, but a receiver bank is relatively inexpensive, and the benefits are enormous (as will now be explained). When a ground station receives a message on a channel outside its primary responsibility, it reads the header and routes the message to the primary ground station to which it was addressed (or a suitable concentrator node that communicates with a regional grouping of ground stations). In this way, each ground station acts as a potential “diversity receiver” for each UA downlink transmission, thereby achieving the diversity goals of the system without increasing spectrum utilization (spectrum is expensive; receivers are cheap). The primary ground station (or a suitable concentrator node – which in the limit could even be the UA control station) merges the return link messages from each UA into a correlated stream by selecting the first copy of each message received and discarding later copies. The correlated stream is routed to the UACS (if the concentrator node is not the UACS itself).

Return link message reception statistics are gathered at each ground station for each UA and shared among the regional ground stations or merged at a concentrator/controller node (which could be the UACS if this is the concentrator node for downlink traffic). In principle, this sharing or merging of data allows for optimum selection of a ground station for forward link transmissions to each UA (i.e., the ground station with greatest probability of message delivery). The selection algorithm has not been defined or agreed, but options include (a) deterministic use of the “primary” until a handoff is executed (this either fails to optimize message delivery probability or requires frequent handoffs which incurs its own overhead penalty), (b) selection of an optimum ground station with uplink transmission on the UA’s preassigned frequency (this requires the non-primary ground station to transmit on the primary’s channel, while the primary remains quiet) or (c) selection of an optimum ground station with uplink transmission on a channel associated with the selected ground station (this requires additional control traffic and a time delay to alert the UA that an unexpected frequency channel should be monitored for an imminent message). Each of these options has advantages and disadvantages that are yet to be assessed and traded off.

For larger UA that operate at both L-band and C-band in order to achieve 99.999% message delivery probability within the LOS subsystem, critical messages are duplicated at L-band and C-band, and parallel processes to those described above are implemented at C-band. Hence, there will be a minimum of two message transmissions (one in each frequency band) and nominally four or more received

copies. It is assumed that each ground station location will host both an L-band and a C-band subsystem. But even though the ground stations host dual-frequency systems, multipath will be statistically independent between the two frequencies. UA manufacturers can also strive for statistical independence of antenna/airframe directive gain patterns as well (although the extent to which this can be achieved in practice is TBD).

For wideband and less critical traffic, such as video to enhance situational awareness, and weather radar, and some low-rate control traffic, only the C-band subsystem is used by the larger UA's. The achieved message delivery probability for these information flows is 99.8 %. By limiting video and weather radar traffic to C-band, the relatively scarce spectrum resource at L-band is conserved.

The waveform and message format are assumed to include forward-error-correction coding which both improves message delivery probability at a given transmitter power level and offers a degree of integrity (i.e., protection against undetected error). As noted earlier and depending on RCP requirements ultimately assessed, the inherent level of integrity offered by the system can be augmented with additional CRC checks, message cross-checks, and selective message repeats (and cross-checks) in order to meet RCP requirements. A selective message repeat can be achieved in less than 500 ms, thereby preserving low latency, and slow frequency hopping from message to message will mitigate multipath fading with statistically independent fade statistics on the repeated message.

Overall, the strawman architecture serves as an existence proof that stringent RCP requirements potentially levied on future UAS CNPC data links will be achievable with existing technology at acceptable cost. Excursions from this strawman, and other CNPC architectures entirely, could be considered and compared on the basis of performance and cost, as the community moves toward a consensus on a preferred approach.

With regard to the BLOS subsystem that could be used by some larger UA, the strawman architecture is somewhat less complete due to uncertainty regarding eventual frequency band support. However, it is clear that 99.8 % message delivery probability on each individual satcom link can be achieved within reasonable design guidelines, for flight above certain threshold altitudes that are dependent on regional rain statistics and time of year. Hence, it is clear that larger UA could operate "at altitude" using a BLOS system for critical CNPC data. A message delivery probability of 99.8 % can be achieved with a single BLOS link, and 99.999 % can be achieved with two BLOS links (one operating at Ku- or Ka-band and the other operating at L- or C-band). Another possibility is to operate with a single BLOS system plus a LOS subsystem in high-density domestic airspace, in order to achieve 99.999 % message delivery probability in this airspace and revert to the single BLOS system in remote areas where a lower RCP requirement could be justified.

It should be noted that even in domestic airspace, a hybrid LOS and BLOS system could prove beneficial for certain users. For example, a suitably designed BLOS subsystem could support CNPC data flows in a protected frequency band (AMS(R)S), along with high-data-rate payload data flows in a neighboring

frequency band without special protection. Such a system, combined with an L-band or C-band LOS link for critical CNPC data flows, could prove to be particularly cost effective.

Reducing Downlink Data Load: Reporting by Exception. It is desirable to minimize the data load associated with each UA – if it can be done safely – since this allows for a greater number of UA to share a given spectrum resource. As noted earlier, in many cases, the sense-and-avoid downlink data – the target tracks for potentially threatening nearby aircraft – dominate the bandwidth used. In the most basic case, each UA transmits a track report on every aircraft it can “sense,” and most of these aircraft are reported by a large number of UA. Hence, there is redundancy on the return link, and the system engineer should consider ways to minimize this redundancy. With this thought in mind, assume that the ATC ground infrastructure broadcasts a local or regional correlated surveillance picture on a “traffic information service-broadcast” (TIS-B) forward link, accessible by all UA in the airspace, and also makes this information available to all UA pilots via terrestrial means (i.e., an authenticated internet stream via the “wired ATC network”). This is a single data flow which requires much less bandwidth than the aggregate of all the sense-and-avoid return links of all the UA in the airspace. The pilot could indicate to the UA that he or she is receiving this data feed – such an indication requires only one very short forward link message every second. The UA can compare its own target track data to that contained in the correlated surveillance picture received from ATC via the TIS-B forward link. If they match within predefined accuracy bounds, for aircraft within a suitable “range of interest” from the UA, and if the UA knows that the pilot has also received this same picture, no return link reporting is necessary (except for a small heartbeat message indicating that the system is operational). If there is a discrepancy, either in an estimated target position or in the detection of a target not reported by ATC, only these differences need to be reported. Of course, if the UA fails to receive the TIS-B forward link or fails to receive an indication that the pilot has received it, the full target list would be reported. This would be a normal event in remote areas where no ground network exists. Of course, in remote areas, there are few aircraft in the airspace and little competition for spectrum resources.

As may be seen, reporting by exception can significantly reduce bandwidth requirements for the system without any loss in safety (i.e., since the fallback modes are fully functional) but at the expense of greater ground system complexity.

32.6 Conclusion

In order to ensure an adequate level of safety for unmanned aircraft operating in shared airspace, the communication link between the unmanned aircraft and the ground-based pilot must carry certain safety-critical data and must be extremely robust and reliable. Initial estimates of channel loading and required communications performance have been developed, and candidate frequency bands for control and non-payload communications have been identified. While further work is needed to fully characterize the communications channel and achieve international consensus

on frequency allocations and other regulatory aspects, the current consensus view is that a safe and workable implementation appears likely to be achievable through a combination of data link diversity and other error correction and error recovery techniques. The precise set of techniques that will be employed, and the extent to which those techniques will be mandated in the system and subsystem performance standards remains to be worked out.

References

- A general model for VHF aeronautical multipath propagation channel. ICAO/AMCP/WG-D/WP6, Presented by Arnaud Dedryvere, Prepared by Benoît Roturier and Beatrice Chateau, Jan 1999
- Air traffic organization safety management system manual, Version 2.1, May 2008
- CCIR, *Handbook of Curves for Radio Wave Propagation Over the Surface of the Earth* (International Telecommunication Union, Geneva, 1991)
- G. Colby et al., Test Results of the Joint FAA/DoD Investigation of GPS Interference, ION (1996)
- Guidance Material and Considerations for Unmanned Aircraft. RTCA DO-304 March 2007
- International Civil Aviation Organization, ICAO Manual on Required Communications Performance. Document 9869 AN/462 (2008)
- International Telecommunications Union, Propagation data and prediction methods required for the design of terrestrial line-of-sight systems. ITU-R Recommendation P.530-12 (2007)
- International Telecommunications Union, Characteristics of unmanned aircraft systems and spectrum requirements to support their safe operation in non-segregated airspace. ITU-R M.2171 (2009)
- International Telecommunications Union, Results of studies of the AM(R)S allocation in the band 960-1 164 MHz and of the AMS(R)S allocation in the band 5 030-5 091 MHz to support control and non-payload communications links for unmanned aircraft systems. ITU-R M.2205 (2010)
- Next Generation Air/Ground Communication System (NEXCOM) Safety and Performance Requirements. RTCA/DO-284, 23 Jan 2003
- A. Rehmann, J.D. Mestre, Air ground data link VHF airline communications and reporting system (ACARS) preliminary test report. FAA Technical Center, Feb 1995
- UAS control and communications link performance – latency. RTCA issue paper SC203-CC009_UAS CC Performance_vD_21Apr09 (2009a)
- UAS Spectrum, RTCA issue paper SC203-CC011_UAS Spectrum_vN_16April09, 2009, and October 2009b revisions
- UAS control and communications link performance – availability and continuity. RTCA issue paper SC203-CC016_UAS_CC_Availability_vK_12Jul2011 (2011)
- W.J. Wilson, Strawman design for terrestrial unmanned aircraft control links, in *Proceeding of the 2011 Integrated Communication, Navigation and Surveillance (ICNS) Conference* (IEEE, Piscataway, 2011)

Christian Wietfeld and Kai Daniel

Contents

| | | |
|--------|---|-----|
| 33.1 | Introduction and Related Work | 750 |
| 33.2 | Design Challenges | 751 |
| 33.2.1 | Key Building Blocks | 751 |
| 33.2.2 | Basic Principles of Context-Aware UAV Swarm Control | 752 |
| 33.2.3 | Reference Scenarios and Key Performance Indicators (KPIs) | 753 |
| 33.3 | Modular Service Platform for Unmanned Aerial Systems | 754 |
| 33.3.1 | Requirements for a Flexible UAS Service Architecture | 754 |
| 33.3.2 | Key Building Blocks of the Service Architecture | 755 |
| 33.3.3 | Validation and Key Learnings | 761 |
| 33.4 | Model-Based Development of Cognitive Steerings | 762 |
| 33.4.1 | Multi-scale-Simulation Environment | 762 |
| 33.4.2 | Validation by Experiments and Measurements | 763 |
| 33.4.3 | UAV-Specific Channel Models | 764 |
| 33.5 | Communication-Aware Networking Algorithms for UAV Swarms | 766 |
| 33.5.1 | Inter-swarm Behavior Controlled by Microscopic Steerings | 766 |
| 33.5.2 | Wide-Area Mobility Controlled by Macroscopic Steerings | 769 |
| 33.5.3 | An Integrated Steering Approach Using Communication-Aware Potential Fields | 770 |
| 33.5.4 | Interference-Aware Positioning of Aerial Relays (IPAR) | 771 |
| 33.6 | Performance Comparisons of Different Steering Algorithms | 771 |
| 33.6.1 | Aerial Sensor Network Scenario | 771 |
| 33.6.2 | Ad-Hoc Aerial Relay Network Scenario Using Channel-Aware Potential Fields | 773 |
| 33.6.3 | Optimal Aerial Relay Positioning Using IPAR | 773 |
| 33.6.4 | Analyzing the Impact of Role-Based Connectivity Management | 776 |
| 33.7 | Conclusions and Outlook | 777 |
| | References | 777 |

C. Wietfeld (✉) • K. Daniel

Communication Networks Institute, Faculty of Electrical Engineering and Information
Technology, TU Dortmund University, Dortmund, Germany

e-mail: christian.wietfeld@tu-dortmund.de; kai.daniel@tu-dortmund.de

Abstract

A new generation of lightweight and small Unmanned Aerial Vehicles (UAVs) enables the design of networked aerial robotic systems for a wide range of applications. In this book chapter challenges and solution approaches with respect to self-organized and robust multi-UAV systems are discussed. In order to achieve a flexible deployment of Unmanned Aerial Systems (UAS), a novel service-oriented system architecture is introduced. The design aspects of mobility control algorithms addressing the competing requirements of communication reliability and spatial distribution of UAV swarms are demonstrated using two reference scenarios with diverging requirements: aerial sensor networks and ad-hoc aerial relay networks. The novel solution approaches presented in this chapter rely on agent-based control of UAV swarms operating autonomously in dynamically changing environments. Different communication-aware algorithms for microscopic as well as macroscopic mobility control are presented, such as Cluster Breathing, Smart Cube, Potential Fields, and Role-Based Connectivity Management. For the positioning of UAV relays, the Interference-Aware Positioning of Aerial Relays (IPAR) algorithm is introduced. The system design and optimization of cognitive networking for UAS requires a dedicated multi-scale simulation environment, which includes a detailed physical channel model and real-world experiments. The chapter discusses how the network and application task-specific performance requirements are met with the proposed mobility control algorithms even in the cases of temporarily unavailable communication links. The self-healing capabilities therefore allow for reliable networking and control of aerial robot swarms in diverse use cases, such as emergency response, environmental monitoring, and ad-hoc network provisioning.

33.1 Introduction and Related Work

The advances in embedded systems technologies have enabled broad research and first deployments of networked sensors/actors to fulfill monitoring and control tasks (Akyildiz et al. 2002). While many of these cyber-physical systems incorporate mobile components and therefore rely on wireless communications, they are typically ground-based. Hence, they are limited in their scope of operation and movement. The research in the area of networked aerial robotic systems focuses on enabling teams of communicating UAVs (Michael et al. 2006; Frew et al. 2008) to autonomously fulfill a given task (How et al. 2009; Chung et al. 2011), such as the exploration of chemical or nuclear plumes (Spears et al. 2009; White et al. 2008; Daniel et al. 2009; Corrigan et al. 2008). In addition to exploration tasks, teams of flying robots may also provide an ad hoc relay network to serve ground-based users in case of network overload or outage situations (Cheng et al. 2008; Pengcheng et al. 2011).

In contrast to many ground-based networking applications in which the mobility of the communication nodes is mandated by the application scenario, the mobility of aerial robot swarms can be actively influenced. These controlled mobility algorithms aim to meet both application task requirements and communication-related performance indicators (Wang et al. 2006; Rooker et al. 2007; Stump et al. 2008; Akkaya et al. 2010; Ghaffarkhah et al. 2011). While exploration tasks may, for example, suggest spreading out in different directions, the need for continuous data transfer may rather demand for a behavior that keeps the swarm close together. Given a known and static environment, it is possible to steer the swarm with preplanned schemes, which fulfill the competing goals. However, in many real-life scenarios, the swarm needs to be prepared for the unexpected, in particular when disaster management applications or component failures need to be considered. To enhance the robustness and flexibility of the robot swarm, current research seeks to distribute the control intelligence across the different platforms, leading to an autonomous swarm behavior controlled by software agents implemented on each UAV (How et al. 2009; Chung et al. 2011).

This book chapter provides an insight into promising new solution approaches for cognitive networking capabilities to solve the trade-off between application task- and communication-related goals. As a specific contribution of this book chapter, cognitive networking algorithms are discussed as part of a generic service platform concept supporting diverging UAS application services. In the following chapter, the key design challenges in terms of building blocks, reference scenarios, and key performance indicators are presented. Then a new modular service platform for UASs is presented, which leverages embedded Web services technologies to allow for the flexible deployment of multipurpose UAV swarms. Before cognitive networking algorithms are discussed in detail, a model-based development approach, which encompasses analytical modeling, multi-scale system simulation as well as experiments, is introduced. Finally, various new communication-aware mobility steering are presented and discussed based on performance evaluation results.

33.2 Design Challenges

33.2.1 Key Building Blocks

The generic design of a swarm-based aerial robotic system considered in this book chapter (cf. Fig. 33.1) consists of UAVs that are equipped with sensors/actors and act as mobile communication nodes at the same time. Depending on the use case, the aerial robotic swarm connects via base stations to ground-based devices, such as mission control entities, databases, and network gateways. In this chapter the focus lies on the air-to-ground (A2G), air-to-air (A2A), and air-to-user (A2U) links, while the choice of the back-end ground connections is left out of scope. Different communication technologies for A2G, A2A, and A2U links can be considered:

- Local area networks (in particular IEEE 802.11a/b/g/n, p and s)
- Cellular networks such as HSPA, IEEE 802.16e, and LTE

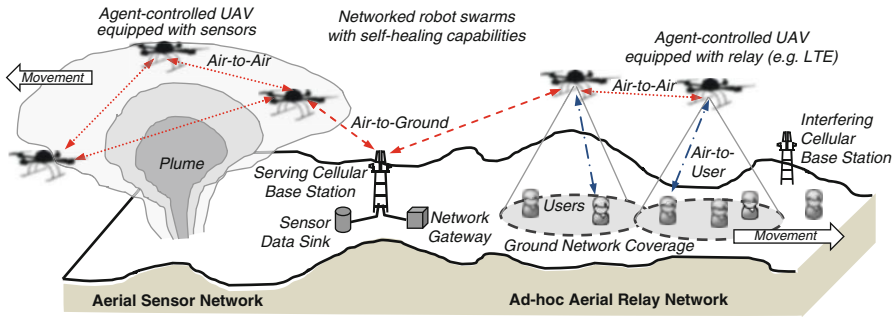


Fig. 33.1 Architecture and reference tasks for networked aerial robot swarm

The primary task of the communication network is to transfer application task-related data, namely, sensor data or user data. At the same time, the swarm control data required to steer the members of the swarm are communicated via the network. To achieve maximum robustness, aerial communication nodes may act as relays in order to provide multiple hop connections and meshing capabilities, as described in Lloyd et al. (2007) and Jung et al. (2010).

33.2.2 Basic Principles of Context-Aware UAV Swarm Control

Due to the dynamically changing, unpredictable system environment leading to potentially unreliable wireless communication links, an autonomous behavior of the swarm is desired. The agent-based control of the robot swarm proposed in this book chapter can be distinguished in two main steering types:

- The microscopic steering controls the behavior of the individual UAV so that a coherent, interconnected swarm is maintained (also called cluster). At the same time, the members of the swarm should not interfere with each other and achieve a maximum impact related to the given swarm task. The microscopic steering is of high importance for the A2A links, as it will self-configure and determine the availability of communication links within the swarm.
- The macroscopic steering controls the swarm according to a given task through a larger-scale scenario, namely, to explore an unknown area or to move from point A to point B. The macroscopic steering will typically be the main influencing factor for the availability of A2G and A2U links.

The key contribution of this book chapter is that both steering types are combined by an autonomous, agent-based decision process on each UAV. Thereby the robot swarm can perform its application task independently of a central ground station. Self-healing capabilities allow that UAVs, which are cut off from the swarm due to obstacles and channel impairments, will automatically reintegrate into the swarm.

33.2.3 Reference Scenarios and Key Performance Indicators (KPIs)

To illustrate the concrete challenges and performance indicators of cognitive networking in aerial robot swarms, two application areas are introduced: the exploration of tropospheric plumes as an example for a monitoring task and the ad hoc network provisioning for outage compensation in terrestrial networks (cf. Fig. 33.1).

33.2.3.1 Scenario 1: Plume Exploration

In the case of a chemical or nuclear accident, disaster managers require information about the current and future development of a plume to decide on potential counter measures for protecting inhabitants and rescue personnel. The aerial sensor system should provide as fast as possible, but not necessarily in real-time, accurate information about the concentration of harmful substances.

This chapter discusses in particular the following key performance indicators:

- **Spatial Exploration Ratio (SER):** The SER is determined by dividing the scenario in a regular spaced grid. If a grid element is accessed by a UAV and once the information was successfully transferred to the data sink, this grid element is counted as being covered. The sum of all covered grid elements divided through the total number of grid elements delivers the SER value.
- **Cluster Separation Ratio (CSR):** Among the variety of communication-related parameters to be considered, the CSR value serves as an indicator to assess the communication performance: the time span, in which one or more UAVs are not connected to the main part of the swarm (the so-called cluster), is determined and related it to the overall mission time.

The key challenge for the control of the aerial robot swarm in this scenario is to allow for continuous communication within the swarm by meeting a minimum CSR target (e.g., $CSR < 1\%$) while at the same time as much space as possible is explored (indicated by a high SER value).

33.2.3.2 Scenario 2: Ad-Hoc Aerial Relay Network

The handling of temporary network outages or overload situations is a major challenge for network operators. Aerial network provisioning is a very interesting option to quickly provide ad hoc network coverage, even if parts of the ground-based infrastructure are not operational or overloaded.

The considered key success criteria for the networked swarm are:

- **Ground Network Coverage (COV):** The COV is determined by cumulating all coverage areas of each individual UAV excluding overlaps, obstacles, and isolated UAVs that are not connected to the swarm.
- **Throughput (TP):** The amount of communication traffic handled by the system in terms of TP is determined by summing up the raw data rates of all individual A2U communication connections within the swarm.

Table 33.1 Overview of challenges and proposed solution approaches

| Scenario | Challenges and design considerations | New channel-aware swarm control algorithms | |
|-----------------------------|--|--|--|
| | | <i>Microscopic</i> | <i>Macroscopic</i> |
| Aerial sensor network | Trade-off between CSR and SER: loose connectivity among the UAVs leads to a high SER, but at the same time the CSR may grow too high | <i>Cluster Breathing (CB), Smart Cube (SC), C.-A. Potential Fields (CAPF)</i> | <i>Cluster Repelling Walk (CRW), Role-Based Connectivity, Return and Release</i> |
| Ad-hoc aerial relay network | Trade-off between TP and COV: a stable TP can be achieved by reducing the COV to adapt to bad channel conditions | <i>Communication-aware Potential Fields (CAPF)</i> <i>Interference-aware positioning of relays (IPAR)</i> | |

The scenario-specific challenge will typically be to achieve a minimum throughput to match the requirements of specific communication service (e.g., $TP > 5 \text{ Mbit/s}$) while at the same time the COV value should be maximized to serve as many users as possible.

Table 33.1 summarizes the challenges and design considerations addressed in this book chapter. It provides an outlook of the corresponding solution approaches discussed later on. Before the proposed swarm control algorithms are discussed, the overall system architecture is introduced in the next section.

33.3 Modular Service Platform for Unmanned Aerial Systems

The research for context-aware communication within Unmanned Aerial Systems (UAS) has focused not only on the mobility control algorithms (see in subsequent sections) but also on the development of a generic and modular service architecture, which supports heterogeneous, multipurpose UAS. In the following section the requirements and solution approach for a generic UAS service platform are introduced.

33.3.1 Requirements for a Flexible UAS Service Architecture

In order to serve different use cases, an Unmanned Aerial System should build on a flexible software and communication architecture. Several architecture approaches have been presented and implemented to address multi-sensor, multi-network-technology and multi-vehicle systems for unmanned autonomous systems (Tisdale et al. 2006; Dias et al. 2008; Love et al. 2009; Elston et al. 2009, 2011). As an

evolution and alternative to these approaches, in this chapter an architecture building on embedded Web services technologies to ease the integration of highly diverse services beyond sensor tasks is introduced.

While service-oriented architectures (SoA) are state of the art for IT services, in the embedded and real-time control systems area, SoA architectures are still an emerging topic. The key benefit of adopting the SoA concept for embedded systems is that the complexity and heterogeneity of different hardware and software components is hidden by well-defined Web Services and clearly structured XML data formats. This allows for the reuse of software components and flexible integration of new components. Recent developments have overcome the concern regarding the overhead introduced by SoA for resource-constraint systems. Prominent examples for the adoption of SoA concepts for embedded systems are the Device Profiles for Web services (DPWS) (Discroll et al. 2009) as well as Efficient XML (EXI) (Schneider et al. 2011). Research projects such as the European MORE project (Wolff et al. 2007) have shown the feasibility of using DPWS for geo-monitoring services and remote control of embedded devices. Other areas, in which SoA-based embedded Web services gain importance, are distributed energy systems and Smart Grid control (Wietfeld et al. 2011).

The design of a future-proof UAS service platform should therefore leverage the concept of a service-oriented architecture while at the same time the real-time constraints of Unmanned Aerial Systems shall be taken into account. The service architecture should be flexible to support diverse aerial services ranging from sensor-based monitoring to communication relaying. Furthermore, different types of Unmanned Aerial Vehicles (quadcopters, fixed wing, tiltwing) as well as different wireless networking technologies (ranging from Wi-Fi to public cellular networks) have to be taken into account for the system architecture design. Appropriate system capabilities are required to allow for autonomous operation in unknown environments and for providing robustness against component and node failures. End users shall be able to operate the system with minimum training, allowing them to focus on the application task to be fulfilled. Therefore, automatic self-configuration and health management shall be supported. Finally, the implementation needs to comply with real-time, safety and security requirements. Hence, state-of-the-art, highly efficient Web services-based concepts, such as optimized IP transport, DPWS, and EXI, are considered to meet the resource constraints of embedded HW/SW platforms.

Within the research project AVIGLE (Rohde et al. 2010), an example UAS service platform fulfilling these requirements has been designed, implemented, and validated by real-life experiments. Key building blocks addressing these requirements are introduced in the next section.

33.3.2 Key Building Blocks of the Service Architecture

As a proof of concept of the flexibility of the architecture, two diverging UAS use cases are considered in the following: a 3D Visual Exploration Service (as an

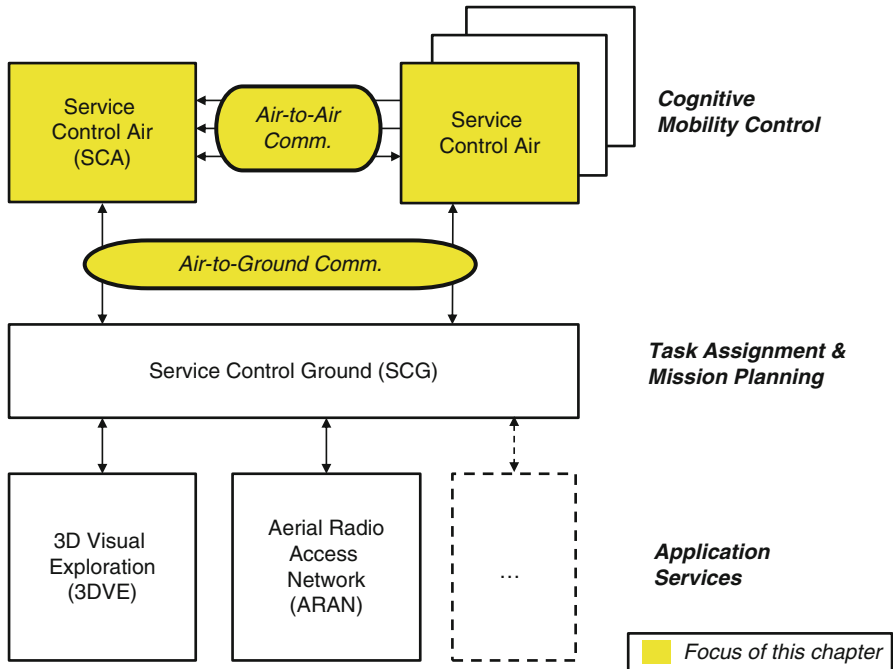


Fig. 33.2 Key components of UAS service platform

example for a generic Sensor Exploration Service) and an Aerial Radio Access Network Service. These reference application services are linked with two core UAS control services groups: the Service Control Ground (SCG) and the Service Control Air (SCA); see Fig. 33.2. The core services provide the generic control functionality to operate the UAS independent of a specific application task, such as navigation services. In addition, communication services are an essential part of the system architecture: on one hand local buses provide onboard communication (e.g., intra-SCA), while wide-area wireless communication services allow for the inter-UAV communication as well as the link between UAVs and the ground-based functionalities.

It should be mentioned at this stage that the architecture is worked out in this chapter in great detail for Unmanned Aerial Systems, but it is flexible enough to cater also other types of autonomous vehicles, such as ground-based or underwater vehicles. In the following, the key services of a UAS are introduced in detail.

33.3.2.1 Service Control Air (SCA)

The Service Control Air (SCA) provides all functionalities to enable the operation of multiple, even heterogeneous UAVs within a UAS. A basic assumption is that each UAV has implemented specific flight control mechanisms, which allows it to be steered by the SCA to a specific position. Therefore, the detailed local control

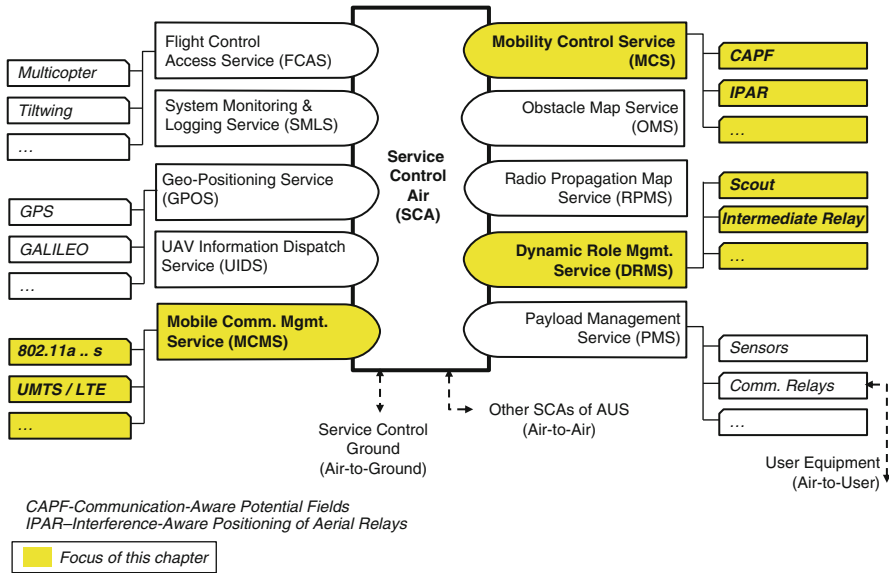


Fig. 33.3 Overview of *Service Control Air (SCA)*

algorithms and components, such as the individual control of the UAV engines, are out of scope for the SCA. The following services are essential parts of the SCA (cf. Fig. 33.3):

- In line with the idea of service-orientation, the complexity of the UAVs' capability to stay in the air and fly into a given direction is hidden through a corresponding Flight Control Access Service (FCAS). The parameters of this service are either global geographical positions or directions.
- The main user of the FCAS is the Mobility Control Service that implements different strategies to achieve predefined performance indicators. The mobility control of a UAS is context depending, while the context is defined by the application task, the communication requirements within the system, and the physical environment (topology, weather, etc.). The in-depth discussion of the appropriate design of the Mobility Control Service is covered in subsequent sections, as different applications task demand for specific Mobility Control Services. Example algorithms are the Communication-Aware Potential Fields (CAPF) and the Interference-Aware Positioning of Aerial Relays (IPAR).
- Another essential service is the Payload Management Service (PMS), which provides the access to different components carried by the UAV to fulfill a given application task. Example payload components are optical sensors (such as infrared cameras), environmental sensors (such as gas or radioactivity detectors), and wireless communication relays, to allow for ground coverage and provisioning of A2U links.

- Via the Payload Management Service, component-specific software libraries are integrated. The modular approach allows for the integration of future payloads, such as new sensor types or robotic components.
- The Geo-Positioning Service (GPOS) provides accurate information about the current position of the UAV. This information is required both for mobility control as well as for application services, which link sensor data with geo-positions. This function will typically leverage satellite-based positioning technologies, such as GPS, GALILEO, and combinations thereof. The accuracy of the positioning information provided by the existing Global Positioning System (GPS) as well as the future GALILEO system is typically in the range of several meters which is sufficient for most application scenarios (Niehöfer et al. 2011). Due to the inaccuracies of satellite-based positioning, additional sense-and-avoid capabilities are required to reliably avoid collisions with obstacles and within the swarm. If needed, additional positioning technologies can be integrated via the GPOS (e.g., for indoor usage).
- The Dynamic Role Management Service (DRMS) manages the UAV's application task capabilities and the task assignment. Through this service available capabilities are offered to the system, while at the same time specific task assignments within the overall UAS may require the deactivation of certain components. For example, a resource-consuming sensor can be switched off, in case the UAV serves only as a communication relay at a given point in time. The Dynamic Role Management Service therefore interacts with the Payload Management Service and the Mobility Control Service.
- Finally, Mobile Communication Management Services (MCMS) are available to allow for the transmission of UAS internal control data as well as payload data, such as pictures, videos, or ground user traffic. Similarly to the Payload Management Service, different wireless communication technologies can be integrated via the service-oriented approach. The MCMS allows integrating mobility management functionality to realize handovers and dynamic mesh routing. The CMS interacts with the MCS as mobility control often needs to take into account communication-related parameters (such as Radio Signal Strength parameters). Throughout the system operation, the CMS may gain information to feed the Radio Propagation Map Service.

The Mobility Control Service is supported by additional services within the SCA, such as:

- Obstacle Map Service (OMS): provides geo-referenced information about the physical characteristics of the environment for proactive, wide-area collision avoidance, in addition to near-distance sense-and-avoid systems of the UAV. The Obstacle Map Service may use either stored map data as well as dynamically generated data gathered by the UAS during exploration tasks.
- Radio Propagation Map Service (RPLS): provides geo-referenced information about the channel characteristics of ground base stations for proactive planning to avoid communication outages between UAVs and the ground station. Similar to the above mentioned Obstacle Map Service, pre-stored data as well as dynamic data can be used.

Support functions such as a System Monitoring and Logging Service (SMLS) as well as an Information Dispatch Service (IDS) ensure the reliable operation of the system.

33.3.2.2 Service Control Ground (SCG)

The Service Control Ground (SCG, cf. Fig. 33.4) is the counterpart of the Service Control Air and therefore contains various corresponding services, such, the Mobile Communication Management Service (MCMS), System Monitoring and Logging Service (SMLS), UAS Obstacle Map Service (UOMS), UAS Radio Propagation Management Service (UPMS), and UAV Payload Management Service (UPMS). While the Service Control Air will be present in multiple instances (one per UAV), the standard system will comprise one SCG instance. The UAV Payload Management Service serves as global repository for the different service capabilities of the various UAVs within the UAS. Due to resource constraints in the UAV, the aerial instances of the map services OMS and RPLS will contain a subset of the information handled in the ground-based instance (UOMS and UPMS). For robustness, the Service Control Ground can be realized in a redundant, physically distributed way, but this consideration is out of scope in this presentation of the architecture.

There are a number of services, which are specific to the ground-based system, compared with the aerial control components:

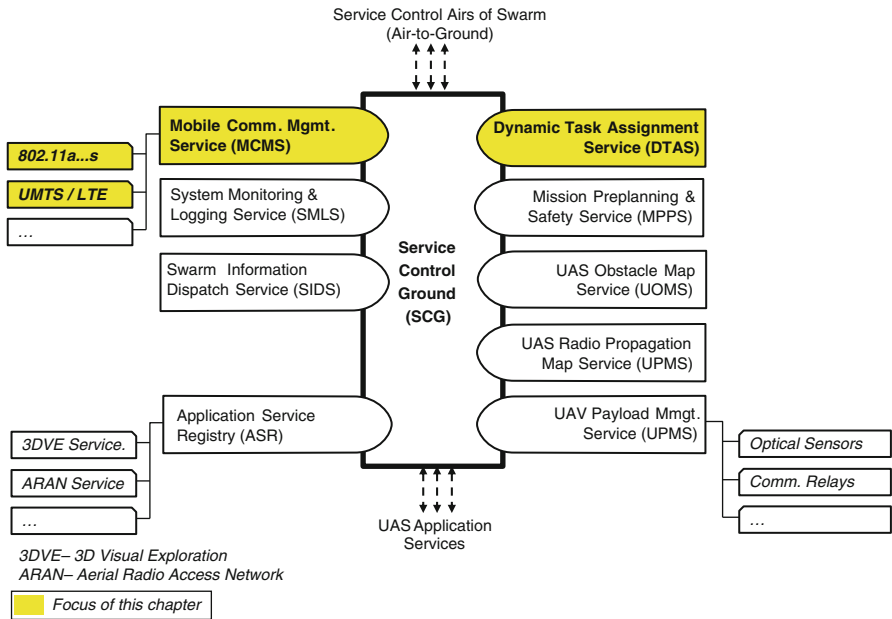


Fig. 33.4 Overview of Service Control Ground (SCG)

- The Dynamic Task Assignment Service (DTAS) provides a user interface toward the end user and allows for the specification of the application details to be fulfilled by the UAS. This service is context sensitive, that is, it must consider the payload components of the UAVs, the flight capabilities of the UAVs, the environmental conditions, etc.
- The DTAS interacts closely with the Mission Pre-Planning and Safety Service (MPPS), which will – based on the task requirements specified by the end user and the available UAV characteristics – breakdown the overall in tasks to be fulfilled by the UAVs' Mobility Control Service. The MPPS leverages information provided by OMS and RPLS.
- The Application Service Registry (ASR) manages the interaction with different application services of the UAS. Different application service will require access to different payload components and mobility control strategies. The ASR therefore interacts with the UPMS, DTAS, and MPPS in order to match the application task requirements with the available service capabilities.
- The Swarm Information Dispatcher Service (SIDS) efficiently routes any data between the ground-based services and the aerial components, may it be control information or payload data. It comprises scheduling and prioritization algorithms to master potential overload situations.

33.3.2.3 Example UAS Services

In the following, two service types in line with the reference scenarios defined in Sect. 33.2 are introduced (cf. Fig. 33.5).

3D Sensor Exploration Service (3DSE)

The 3D Sensor Exploration Service (3DSE) serves as an example for a distributed aerial sensor network application. The UAVs are equipped with sensors, such as

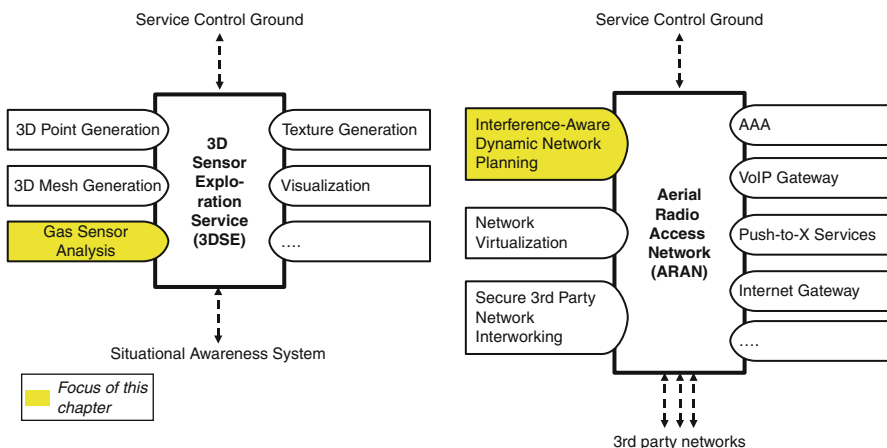


Fig. 33.5 Example application services for communication-aware UAV operations

gas sensors or cameras as a payload, and deliver geo-referenced gas sensor data or aerial photographs of the earth's surface from different perspectives. This service may comprise very specialized functionality, such as 3D point generation, 3D mesh generation, and texture generation, in order to finally visualize the results for the users of the service (visualization). A detailed description of the image processing and virtualization approach is provided by Rohde et al. (2010). An example for an aerial gas sensor application is described in Daniel et al. (2011). Potential users of the 3DSE are rescue organizations, which require detailed information of a city environment after an earthquake or after a large-scale chemical incident. Using an up-to-date 3D map as part of situational awareness systems allows efficient planning and optimization of rescue actions. But the results of this service can also be leveraged within the UAS itself, because it provides current data for the mission planning through the obstacle and radio propagation services.

Aerial Radio Access Network Service (ARAN)

The Aerial Radio Access Network Service (ARAN) provides ad-hoc ground network coverage by aerial relays. It requires communication relays as payload on the UAVs' side and specific mobility control algorithms. Examples for standard functionalities are AAA services, a VoIP gateway, and an Internet gateway.

UAS-specific service functionalities are related to:

- Interference-Aware Dynamic Network Planning which needs to take into account both coverage requirements as well as interference mitigation. This aspect is addressed in subsequent sections.
- Push-to-X services which are required to support IP-based, multimedia group communications. These services are of specific importance for ad-hoc communication of emergency response organizations (Subik et al. 2011).
- Secure Third Party Network Interworking is required to integrate the Aerial Radio Access Network with ground-based networks, such as a public cellular network, which shall be complemented with the aerial network to compensate network outages.
- Network virtualization functionalities are required in case the aerial network shall be used by multiple network operators. In this case, the available system capacity needs to be shared amongst the different networks based on service contracts. Therefore, scheduling and monitoring of Quality of Services parameters are part of this building block. The network virtualization is subject of ongoing research and out of scope for this chapter.

33.3.3 Validation and Key Learnings

The implementation of the proposed architecture has been carried out on embedded platforms (such as Vortex86DX and ARM Cortex-A8) and validated for flexibility and efficiency in experiments (see below). It has shown the embedded Web services approach is very suitable for the integration of services, such as different sensor types or communication relays. Nevertheless, in case of specialized sensors (such

as IR cameras), the implementation of software connectors to integrate proprietary drivers with the generic architecture requires significant effort. Therefore, the benefits of a Web services-based approach can be only fully leveraged, when the suppliers of the UAV payload components will provide a Web service compliant-interface off the shelf. Due to the usage of highly optimized binary EXI coding, the actual size and processing times of data messages to be exchanged within the system were similar to a comparable, non-Web services-based implementation. For real-time critical parts of the system, the data message structures have been hard-coded in order to avoid encoding delays.

In summary, the embedded Web services approach has proven to be feasible for UAS, as it provides a framework for a common, well-established formal description of service characteristics and data formats and at the same time allows to trade flexibility with efficiency by using highly efficient data encoding techniques.

33.4 Model-Based Development of Cognitive Steerings

33.4.1 Multi-scale-Simulation Environment

For the performance evaluation of cognitive UAV networks and the corresponding steering algorithms, it is essential to consider multiple system components and characteristics (cf. Fig. 33.6):

- The mobility model delivers the dynamically changing position of the UAVs.
- The wireless channel model’s output – the Radio Signal Strength – is derived from the current position of the UAV taking into account a realistically modeled

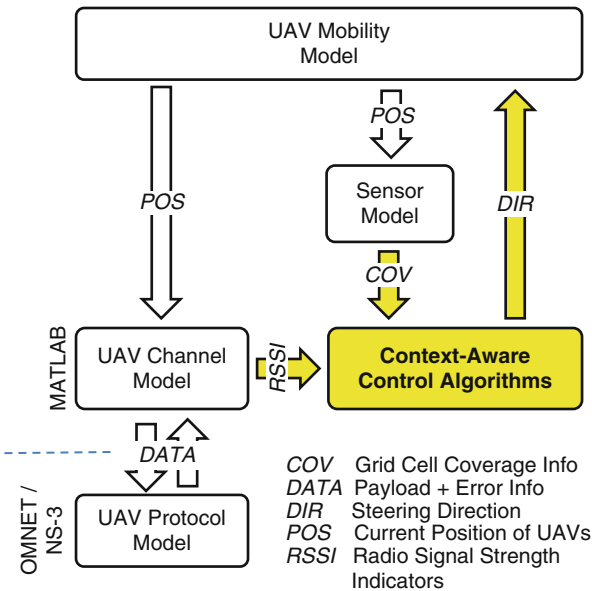


Fig. 33.6 Example information flows in multi-scale system simulation environment

system environment (topology, obstacles, interferers, antenna characteristics, etc.).

- The sensor model deployed with the UAV allows to determine the spatial coverage of the sensor based on its current position. In case of a plume exploration task, it is assumed for simplification that once the UAV has entered a grid cell, sensor measurement values for the complete grid cell are instantly available.
- The communication protocol models deliver detailed information about the timing and length of data transmissions.

To address these different system aspects, a multi-scale simulation environment is required, which combines various, specialized simulation models and real-life HW/SW implementations into a complete system model (Daniel et al. 2011). Figure 33.6 provides an overview of major information flows between the various subsystem models and the control algorithms to be investigated.

A special focus of the performance analysis is placed on the dynamic meshing capabilities, as short-life routing paths have to be reconfigured and re-established as fast as possible to avoid separations within the aerial swarm network. Subsequently, MANET and VANET protocols have been integrated into the simulation environment to allow the analysis of the interdependencies among channel, mobility, and meshing capabilities.

With regard to an appropriate trajectory calculation, a UAV-specific kinematic movement model is utilized: while the agent-based steering control algorithms mandate desired future positions of the UAV, the physical characteristics (mass, acceleration) need to be taken into account to achieve realistically modeled swarm behavior. The system component models are developed and validated based on dedicated experiments (e.g., channel measurements), which are introduced in the next section.

33.4.2 Validation by Experiments and Measurements

Although the multi-scale simulation models provide a powerful and realistically modeled environment to evaluate the system, experiments are essential to validate both dedicated system component models as well as the overall system performance.

In the research projects AirShield (Daniel et al. 2011) and AVIGLE (Goddemeier et al. 2012), different types of UAVs, ranging from multi-copter to tilt-wings and blimps, are deployed to validate mobility control algorithms. Swarm control data (telemetry) and payload-related data (gas sensor values) were successfully transmitted and routed over a Wi-Fi (A2A mesh) and UMTS-HSPA (A2G) network (cf. Fig. 33.7).

With respect to the relevance of the mesh protocol for the air-to-air links, several experiments in an EMC anechoic room have been conducted and analyzed with regard to the 802.11 performance in an interfered environment (Daniel et al. 2010). Furthermore, the impact of interferences on the reliability of air-to-ground links has also been analyzed for LTE and Mobile WiMAX by means of different network and channel emulators (e.g., R&S CMW 500, EB Prosim C8). The air-to-ground



Fig. 33.7 Example experiments carried out in the AirShield project: validation of A2G and A2A links (*left*), validation of sensor data transfer (*middle*), gas sensor and communication payload (*right*)

availability of cellular networks in higher altitudes (up to 500 m) has been surveyed with a captive balloon in urban and suburban environments. Since the channel characteristic is a crucial system property for UAV networks, results of recent studies (Goddemeier et al. 2012) are presented in the following section.

33.4.3 UAV-Specific Channel Models

The UAV channel is influenced by numerous factors such as antenna characteristics, attenuation, multipath propagation, and interference. Extensive work has been performed on ground-based wireless systems as well as satellite systems, which can be partly applied for the UAV channel. However, the UAV channel bears certain specifics due to its different geometry.

The air-to-air (A2A) channel is characterized by a dominant direct path and additional interference due to reflections from the ground (Allred et al. 2007). The characteristics of the air-to-ground (A2G) channel are highly dependent on the used communication technology in use. Leveraging publicly available cellular networks are useful to provide wide-area air-to-ground coverage without additional infrastructure. Therefore, a dedicated UAV channel model for A2G links is of specific interest for the system modeling.

Base station antennas in cellular network systems usually show directional characteristics providing reliable ground coverage for the end users.

Figure 33.8 depicts the specific characteristics of the height-dependent UAV channel, based on measurement data, ray-tracing results, and a specifically developed analytical channel model (Goddemeier et al. 2012). The channel can be characterized by 4 zones: at low heights up to 10–20 m, shadowing effects are observed (Zone I), while in heights up to approx. 50 m, a strong direct path interferes with reflections of the ground respectively the roof tops (Zone II). At heights above 50 m, the UAVs antenna leaves the main antenna lobe and a reflected path from

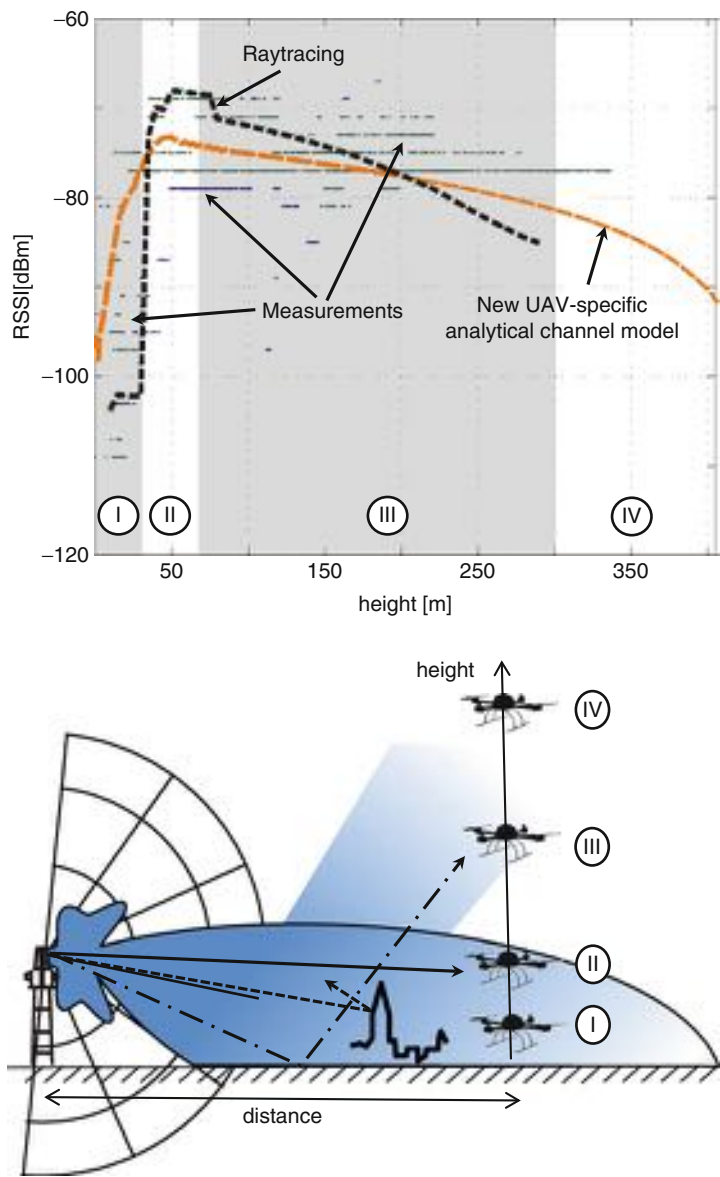


Fig. 33.8 UAV channel characteristics: air-to-ground link

the ground dominates (Zone III). Finally, for altitudes above 300 m, the received power for direct paths as well as reflected paths decreases. Consequently, no reliable communication is possible (Zone IV). The results illustrated in Fig. 33.8 have been validated with channel measurements in real-life UMTS/GSM networks in Germany

by utilizing a captive balloon ($h < 500$ m). The key conclusion from this synopsis of a far more detailed analysis (Goddemeier et al. 2012) is that the exploration at heights above approx. 250 m requires communication-aware aerial relay nodes with meshing capabilities to be able to extend the operational range by establishing multi-hop communication links.

33.5 Communication-Aware Networking Algorithms for UAV Swarms

The networked UAV systems discussed in this book chapter rely on autonomously controlled mobility. Recent research has come up with several mobility control strategies which have often been inspired by nature or found an illustrative counterpart in nature. An in-depth survey of connectivity-dependent algorithms to control the position of mobile sensor nodes is provided in (Younis et al. 2008). Examples of relevant work related to robot mobility control have been presented by Wang et al. (2006), Rooker et al. (2007), Stump et al. (2008), Elston et al. (2009), Han et al. (2009), Akkaya et al. (2010), and Ghaffarkhah et al. (2011). This chapter focuses in particular on algorithms addressing the interdependency and desired equilibrium between coverage and connectivity requirements in UAV swarm missions.

33.5.1 Inter-swarm Behavior Controlled by Microscopic Steerings

For the distributed control of a swarm, in Reynolds (1987) three basic rules are established: separation, alignment, and cohesion. Based on the physical distances between neighbors, the behavior of the swarm is controlled to avoid collisions and to ensure harmonized movements for the computer-based modeling of biological swarms and herds. This fundamental work has inspired also the research on controlled mobility for robot swarms; see, for example, Cao et al. (1997) and Ryan et al. (2004). In the context UAVs performing jointly an exploration task (see scenario 1), the coverage of the swarm shall be maximized (leading to separation) while at the same time connectivity between the UAVs needs to be preserved (leading to cohesion). In Daniel et al. (2011) a corresponding channel-quality-aware mobility control algorithm, which introduces the so-called Cluster Breathing (CB) rules, has been proposed and studied. This algorithm considers the received power from neighboring vehicles, the radio Signal Strength Indicator (RSSI) as space and time-variant communication parameter (cf. flow chart shown on top of Fig. 33.9). As the RSSI will vary due to the dynamically changing channel characteristics, the swarm will “breathe” to a certain degree. Similar to other highly dynamic control processes in mobile radio networks (such as power control or handover decisions), the impact of short-term fading must be compensated appropriately, for example, by introducing medium-term averaging of RSSI values and hysteresis. To achieve a better spreading of the UAVs when fulfilling a swarm task, such as a plume

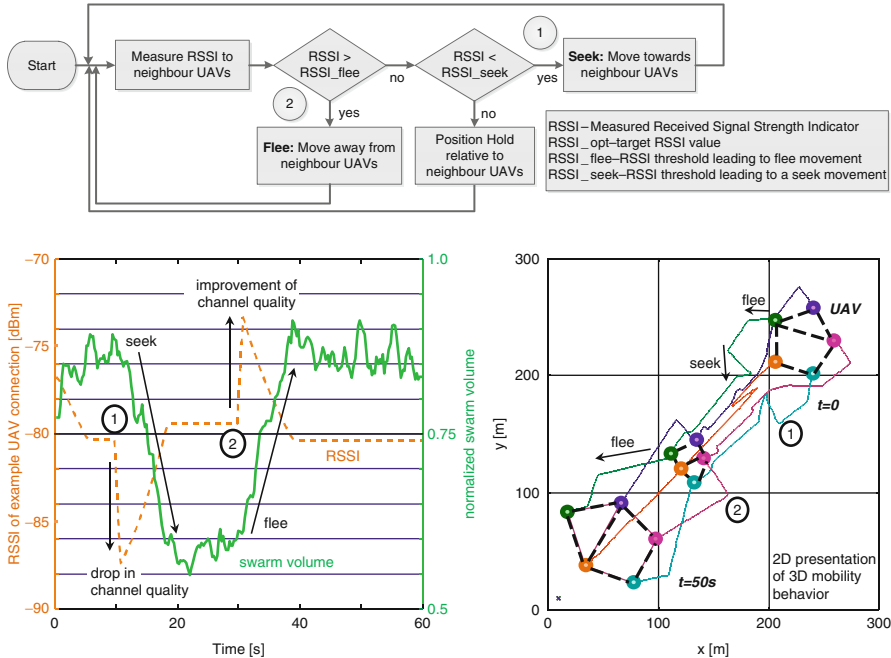


Fig. 33.9 Example of cognitive steering due to changing channel characteristics (Cluster Breathing): Flow Chart, RSSI vs. Swarm Volume Chart, Spatiotemporal 2D Chart Of Swarm Behavior

detection, the number of required neighbors can be limited, for example, to two or three.

Figure 33.9 shows an example of the behavior of an aerial robot swarm: after the robots have been separated according to the given RSSI boundaries, an area with stronger impact of fading is entered at 10 s (1), which leads automatically to a reduction of the distances between the robots respectively of the volume created by the network nodes. With improved channel quality (2), a flee movement is initiated, until the target RSSI value is reached.

Despite careful selection of the control parameters, it might nevertheless happen that the connection to one or more UAVs is completely lost. These situations require cluster-self-healing capabilities: one option is, for example, that the UAVs of the swarm move back to the position, where the connection was lost. In some cases, it might be intended to split up the swarm for a given time to fulfill the swarm task more efficiently. This special situation is discussed below in more detail (cf. Sect. 33.6.4).

The Cluster Breathing (CB) steering as described above is able to meet the communication-related goals on a microscopic level, but has the drawback that it does not exploit the full potential regarding the local exploration coverage of the swarm: the UAVs within the swarm will not or very rarely move into the edges of a cube, even if this would be possible from a communication perspective.

A static configuration can be ruled out, as it will not be possible to react on the dynamically changing physical channel characteristics. Consequently, a more advanced algorithm has been developed, namely, the Smart Cube algorithm.

The Smart Cube (SC) (Daniel et al. 2011) approach aims to optimize the 3D distribution of the UAV agents within a given cube by using Voronoi tessellation (Ammari and Das 2010) among other techniques. This algorithm provides a better local coverage in case of spatial exploration tasks. To cater for RSSI variations, the Smart Cube concept adapts the size of the cube to be covered dynamically according to current physical channel conditions. However, the performance evaluation of larger-scale scenarios has shown that the overall performance of given exploration tasks is strongly impacted by the macroscopic behavior, which might efficiently compensate for deficiencies in the local coverage of the swarm.

In order to maintain the connectivity of the UAV swarm, the maximum distances between the nodes needs to be limited. Instead of using distances the Smart Cube uses a virtual cube which size is determined by an RSSI threshold (cf. Fig. 33.10a)

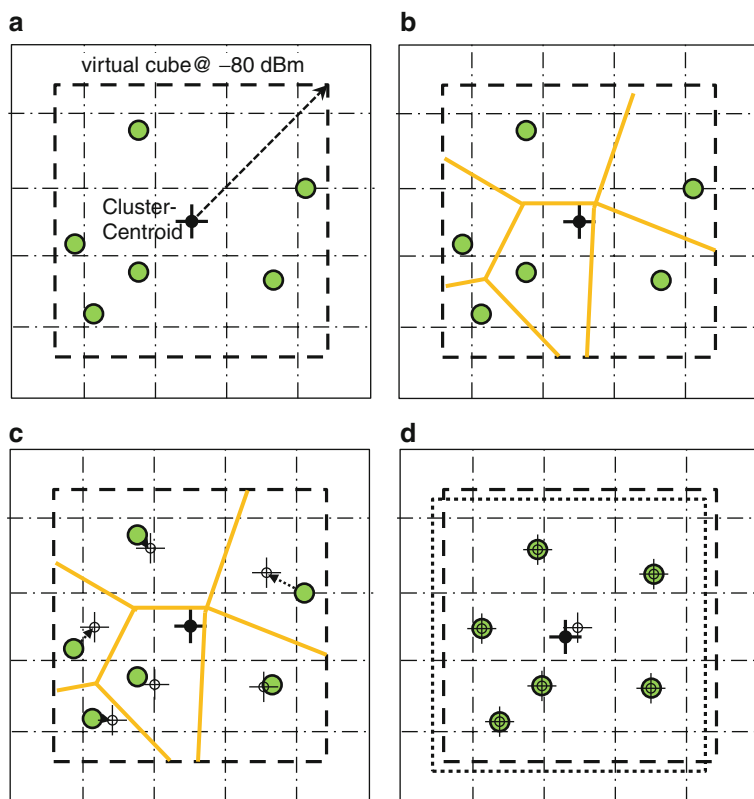


Fig. 33.10 Achieving better spatial distribution through *Smart Cube* algorithm. (a) Initial state. (b) Voronoi tessellation. (c) Lloyd movement. (d) Improved topology

for achieving communication awareness. Within this cube a Voronoi tessellation is done by each agent (cf. Fig. 33.10b). On basis of the Voronoi tessellation, the spatial volume difference between the single Voronoi cells can be evaluated. This information is utilized by each agent to iteratively increase or decrease the owned space in order to achieve an equal volume distribution (cf. Fig. 33.10c). This algorithm is known as Lloyd or k-means-algorithm (Du et al. 2006). The Lloyd algorithm itself is not communication aware. For avoiding any separation, the virtual cube restricts possible the target positions to an area where the RSSI between the nodes will be sufficient.

33.5.2 Wide-Area Mobility Controlled by Macroscopic Steerings

The macroscopic behavior of the swarm is highly dependent on the task to be fulfilled by the swarm. In case of an exploration task (see reference scenario 1), the swarm needs to move as quickly as possible through a given volume, bearing in mind the key performance indicators, in particular exploration efficiency. In completely static and homogenous scenarios, a deterministic movement of a fixed configuration of UAVs working row by row through the scenario can be considered. Such a mechanism is neither communication-aware nor would it allow for a dynamic and self-organizing adaption to complex scenarios and UAV outages. From an exploration perspective, the well-known multiple traveling salesman problem (mTSP) provides a solution approach to find the shortest path to visit all relevant grid cells within a known volume. This centralized algorithm can serve as a best-case reference in terms of exploration efficiency (Daniel et al. 2011), but it operates completely independent from communication-related boundary conditions as well as from application data gathered during the exploration. Subsequently, it is necessary to develop a situation-aware behavior, which takes an eventually dynamically changing and unknown environment into account:

- The Self-Repelling Walk (SRW) presents a known algorithm to steer the behavior of individual mobile agents, avoiding intersections with their own paths as much as possible. In its basic form, this approach can serve as a reference for an autonomous behavior of the individual UAVs within the swarm, but it takes neither communication requirements nor cooperative behavior into account.
- The Cluster Repelling Walk (CRW) applies the principles of the SRW to a complete communication cluster being superimposed by microscopic steerings at the same time. On a macroscopic level the swarm moves synchronously while being still able to compensate for changing channel environments through microscopic control algorithms (cf. Fig. 33.11).

An alternative to the combination of different microscopic and macroscopic steerings can be provided by a novel approach, namely, Communication-Aware Potential Fields (CAPF), which is described in more detail below.

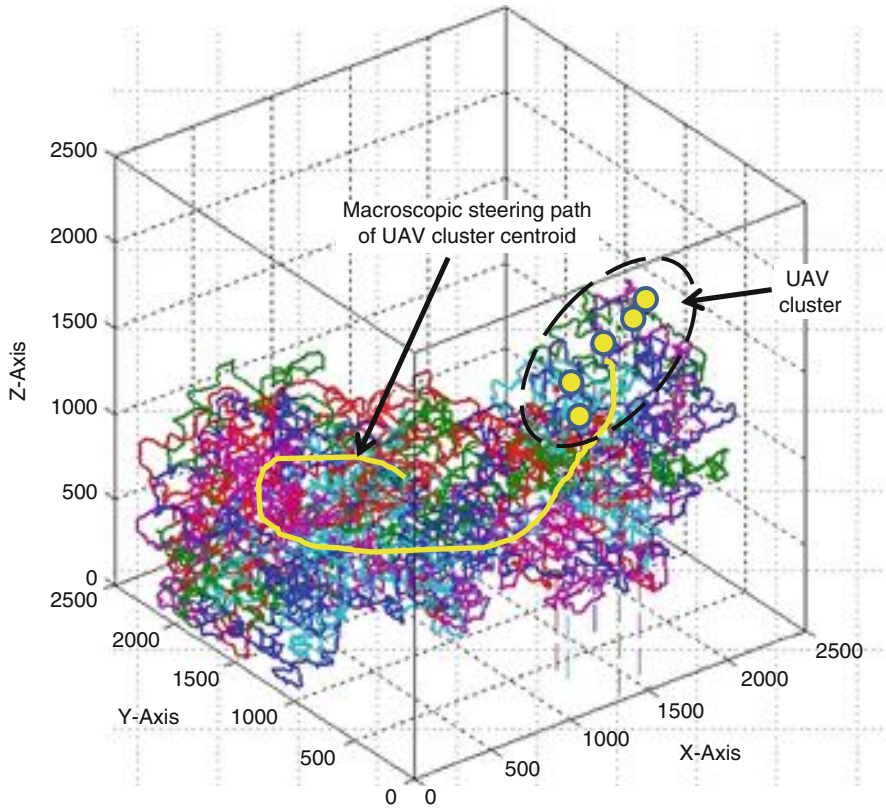


Fig. 33.11 Exploration by macroscopic steering (e.g., Cluster Repelling Walk)

33.5.3 An Integrated Steering Approach Using Communication-Aware Potential Fields

The Communication-Aware Potential Fields (CAPF) algorithm expands the potential fields approach used to guide UAVs. Originally (Howard et al. 2002), the Potential Fields have been introduced to allow the mobile agents to derive attracting and repelling forces from a given environment. Thereby obstacles can be associated with a Potential Field creating repelling forces, while points of interest trigger attracting forces. The combination of potential forces with connectivity targets has been further evolved, for example, by Zou et al. (2003), Wang et al. (2005), Garetto et al. (2007) and Tan et al. (2008). In Goddemeier et al. (2011) UAVs have been associated with potential fields, thereby creating respective attracting as well as repelling forces (corresponding to low and high RSSI values when applying the Cluster Breathing method). In case of aerial sensor networks, the Communication-Aware Potential Fields (CAPF) are used for microscopic control combined with exploration-oriented macroscopic steerings. In case of the ad-hoc relay network

scenario, the CAPF approach is suited to provide both micro- as well as macroscopic steering behavior: the area to be provisioned with a wireless communication service can, for example, be associated with a potential field generating an attracting force. In addition, base stations required to ensure connectivity with other networks induce attracting forces. In case of several UAVs providing the ground-based communication service, repelling forces between the UAVs are introduced to ensure a sufficient spacing to avoid interference and allow for maximum ground coverage.

33.5.4 Interference-Aware Positioning of Aerial Relays (IPAR)

The introduction of terrestrial relays and femto-cells to increase the spectral efficiency of broadband cellular networks has gained much attention in recent years, for example, in Pabst et al. (2004) and Andrews et al. (2011). A recently proposed system concept is based on a swarm of UAVs acting as mobile relays to offload traffic to surrounding terrestrial cells with free resources (Rohde et al. 2012). Thereby cell outages as well as cell overload situations can be compensated. The optimal position of the aerial relays is highly dependent of the inter-cell interference (ICI) introduced by the mobile relays: in case that a mobile relay is operated in the vicinity too close to a ground-based station (cf. Fig. 33.1), the mobile relay will cause harmful interference resulting in reduced traffic capacity. Therefore, the specific Interference-Aware Positioning of Aerial Relays (IPAR) algorithm was introduced (Rohde et al. 2012). The IPAR algorithm's goal is thus to find positions that yield a maximized throughput of the cellular network for the aerial relays while considering interference and available traffic load. The IPAR algorithm uses an interference and load aware analytical model, inspired by Elayoubi et al. (2008), to evaluate the system performance (fitness) and iteratively improves an initial population of relaying scenarios through mutation and recombination of the relay positioning. Example results for the positioning of UAVs leveraging the IPAR algorithm are presented below (cf. Sect. 33.6.3).

33.6 Performance Comparisons of Different Steering Algorithms

In the following, the performance of different algorithms presented above is presented for the two reference scenarios: aerial sensor network and aerial relay networks.

33.6.1 Aerial Sensor Network Scenario

In this section different algorithms to address the aerial sensor network scenario are discussed. Figure 33.12 shows example results related to the Spatial Exploration Ratio (SER – percentage of grid cells visited by the swarm) and the Cluster

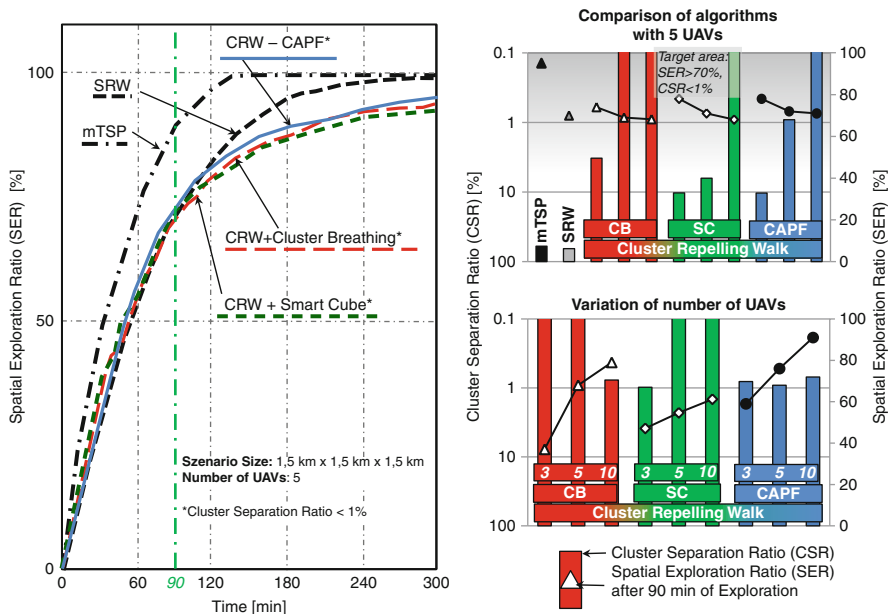


Fig. 33.12 Comparison of different steering algorithms: CRW combined with CB, SC, and CAPF for different parameterizations and swarm sizes (3, 5, and 10 UAVs)

Separation Ratio (CSR – percentage of simulation time, in which the swarm is split up in two or more communication clusters) for selected combinations of steerings corresponding to reference scenario 1. The swarm shall explore a given volume as fast as possible while maintaining stable communication links within the swarm. In order to demonstrate the dependency between inter-swarm communication and spatial exploration performance, this study focuses on the A2A connectivity (Wi-Fi at 2.4 GHz links). The top right-hand part of the figure shows selected SER and CSR after 90 min of exploration for different RSSI thresholds. The left-hand part shows the SER over time for those simulation runs, in which CSR values below 1 % were achieved. The results for a swarm with five UAVs show that the exploration task can be efficiently performed even in the presence of communication related boundary conditions. While the centrally planned reference mTSP delivers very good results for the exploration task, it shows a high number of cluster separations at the same time. The SRW without communication-awareness fulfills the exploration task with very good performance, but it also fails in terms of communication performance. The dedicated, communication-aware strategies (Cluster Repelling Walk combined with CB, SC, or CAPF) show that – given a suitable parameterization – an appropriate equilibrium between low Cluster Separation Ratio (below 1 %) and Spatial Exploration Ratio (around 70 % after 90 min of exploration) is possible. The lower-right hand part of the figure shows the results for CSR and SER for different swarm sizes, ranging from three to ten UAVs.

The selected results shown in Fig. 33.12 have been produced based on numerous simulations with different parameterizations of the algorithms, in particular by varying the RSSI thresholds as well as the weighting of macro- and microscopic mobility control. They indicate that a trade-off between communication and exploration performance is possible. The study also shows that the communication performance is much more sensitive than the exploration success: for example, the CAPF algorithm can achieve a very good communication performance ($CSR < 0.1\%$ instead of 10%) when accepting a 10% lower Spatial Exploration Ratio (SER 71% instead of 78% ; see top right-hand part of Fig. 33.11). With regard to the swarm size, the number of UAVs helps to improve the SER for all algorithms (up to 91% SER for ten UAVs; see results in Fig. 33.11, lower right-hand part). The results therefore show that the system scales well, but at the same time the SER gain per additional UAV decreases with the number of UAVs. This is an example of sub-modularity, which is a property that can be proven for most search problems; see, for example, Krause et al. (2008). One interpretation of these results is that the additional communication constraints preserve this property and limit the impact of additional UAVs on the search efficiency. For the given use case it can be concluded that already with a smaller number of UAVs (three or five) satisfactory SER values are achieved. The cost of deploying more UAVs needs to be carefully assessed against the benefit of improved exploration efficiency.

33.6.2 Ad-Hoc Aerial Relay Network Scenario Using Channel-Aware Potential Fields

With regard to the ad-hoc aerial relay network use case, the first scenario considers an emergency response scenario, in which terrestrial networks are not available anymore (Goddemeier et al. 2011). Figure 33.13 shows example results related to a network provisioning scenario for a crowd of users (e.g., firemen) which moves through an area with larger obstacles/urban canyons (cf. left part of Fig. 33.13). In this case, the channel-aware potential field algorithm controls the micro as well as macro mobility of the UAV swarm. The results (cf. right part of Fig. 33.13) show that different channel qualities lead to different ground coverage areas while at the same time allowing for constant throughput TP. The example results also demonstrate the impact of obstacles: when the distances between UAVs grow, the ground coverage is higher, but the risk for a splitting of the swarm due to obstacles also grows, leading to temporary reduction of the ground coverage area.

33.6.3 Optimal Aerial Relay Positioning Using IPAR

While the above scenario did not consider interference from other networks, the following use case focuses on the optimal positioning in the presence of a terrestrial cellular network. In the left part of Fig. 33.14, the performance of a system with a regular hexagonal layout (frequency reuse factor 1) and different numbers of aerial

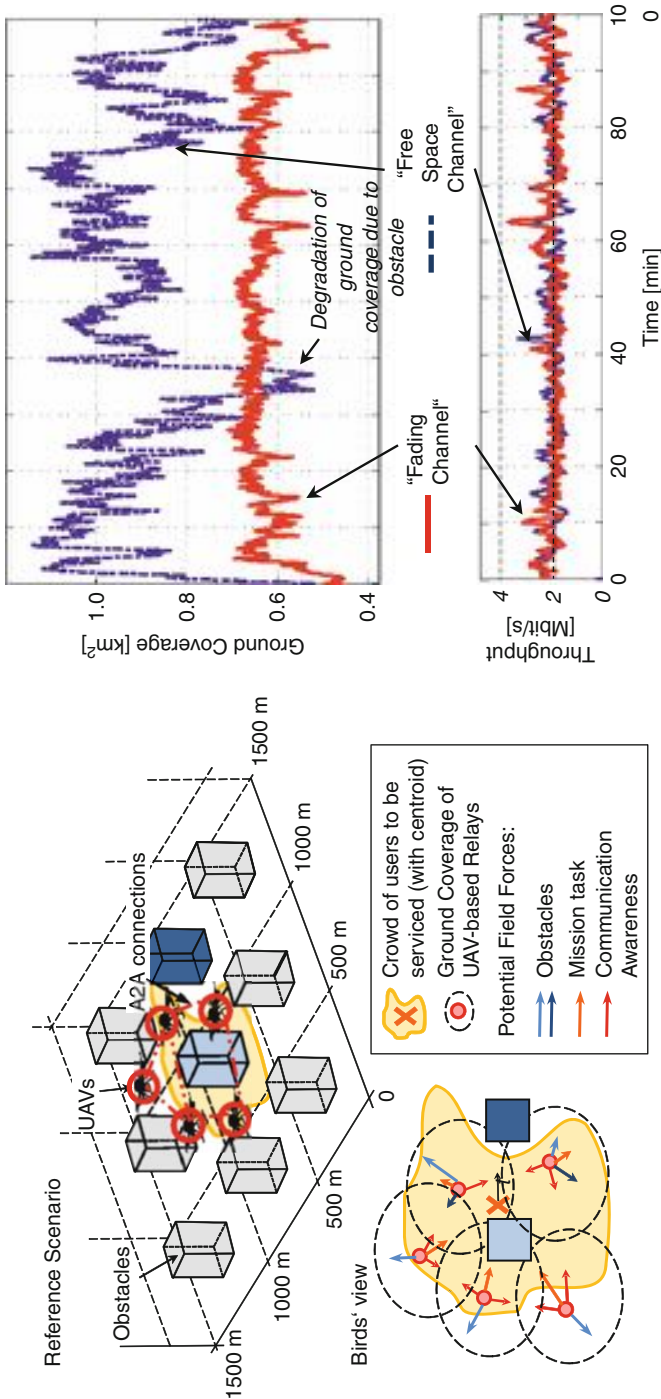


Fig. 33.13 Providing ground network coverage by aerial robot swarm (e.g., Scenario and results)

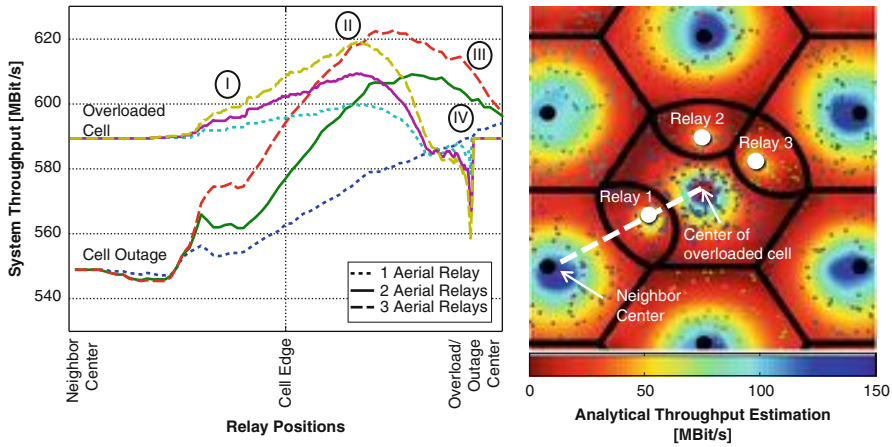


Fig. 33.14 Positioning of aerial relays based on IPAR

relays (1, 2, and 3) is shown for an overload situation as well as a complete outage of the cell. The total downlink system throughput for all connections to users within the considered scenario is shown for different relay positions. The relays move synchronously, starting from the center of the neighbor cell.

The results for the overload scenario show that with a movement of the UAV(s) toward the center of the overloaded cell (cf. (I) in Fig. 33.14), there is a slow increase in total throughput while cell overload and thus packet loss is compensated. It is visible that the method scales quite well for a small amount of relays, leading to a maximum system throughput when three relays are in use (cf. (II) in Fig. 33.14). The most important observation is that the best position of the aerial relay(s) is not at the cell edge, but at around two thirds of the distance from neighbor cell center to the center of the overloaded cell (Rohde et al. 2012). Once the UAV relay(s) further approach the center of the overloaded cell, the throughput drops significantly due to increased interference (cf. (III) in Fig. 33.14).

In case of the cell outage, the results also demonstrate the dependence of the optimal position on the level of interference. For the case of one UAV, the aerial relay (see lower dotted line, which shows a maximum throughput at the cell center; see (IV) in Fig. 33.14) is best positioned at the center of the failed and overloaded macro cell while in case of two and three aerial relays the UAVs should best be positioned in an appropriate distance from the failed macro cell. This underlines how the interference imposed by the relays and macro cells on each other strongly affects the optimal relay positioning. While the left part of Fig. 33.14 shows the full spectrum of possible relay positions and their corresponding system throughput, the right part of the figure shows an example throughput heat map for three aerial relays resulting from UAV positions determined by the IPAR optimization algorithm. The cell boundaries shown in the figure are derived by associating mobile equipment with those terrestrial cells or relays which provide the strongest RSSI (Rohde et al. 2012).

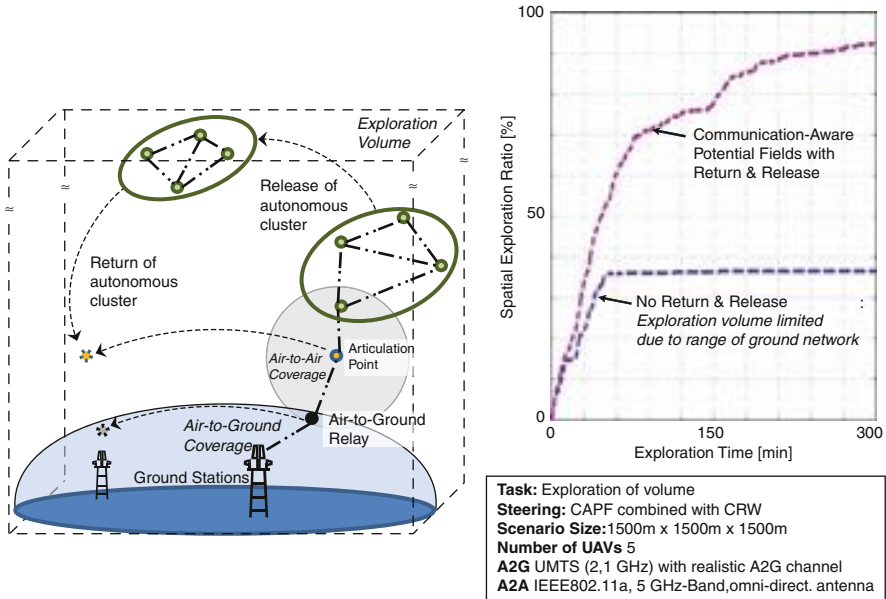


Fig. 33.15 Illustration of different roles within an aerial robot network with limited base station coverage and example results showing an improved spatial exploration ratio

33.6.4 Analyzing the Impact of Role-Based Connectivity Management

Various application scenarios, in which robot swarms can be very useful, require the operation of the swarm across larger distances and altitudes. In those cases, limitations in the range of the communication links need to be addressed by assigning different roles (Odell et al. 2004). In this last use case to be addressed in this chapter, a concept of Role Based Connectivity Management for UAV swarms is presented, taking into account the specific channel characteristics as outlined above. Figure 33.15 shows an example scenario in which publicly available cellular networks shall provide A2G coverage, while Wi-Fi technology is used for the A2A links. The swarm task requires operating in heights of up to 1.5 km, which is out of range of the cellular network coverage (as discussed above in the context of the A2G channel model). To extend the operating range of the swarm, multi-hop links are established: dedicated UAVs stay within the range of the A2G system, while others move beyond to explore the given volume. Thus, different roles are assigned to UAVs, such as an A2G relay node, an articulation point as relay node, and meshed sensor nodes, which are called scouts (cf. Fig. 33.15). While the swarm of scouts performs the exploration task, the relays ensure connectivity to the ground station: the A2G relay node maintains the direct link to the ground station. The articulation point identifies the intermediate node which is required to interconnect

the mesh network and the A2G relay. However, even the use of multi-hop links will not allow for the exploration of the complete scenario depending on the number of UAVs used. Hence, controlled temporary link interruptions are introduced: with a dedicated release and return strategy (RRS), a part of the swarm moves beyond the communication range of the articulation point and reintegrates according to an autonomous control algorithm. With such an approach, dynamically growing search volumes can be accommodated leveraging a given, eventually limited ground-based communication infrastructure. Figure 33.15 (right) shows an example comparison of the spatial exploration ratio over time, in which the RRS enables increasing the SER by exceeding the restricted communication range. In-depth investigations have proven the feasibility and benefits of this approach (Goddemeier et al. 2012).

33.7 Conclusions and Outlook

In this chapter key challenges and essential building blocks for the design of networked aerial robot swarms using self-organized, communication-aware mobility control are introduced. The contribution focuses on multipurpose UAV systems and therefore introduces a modular, service-oriented architecture, which allows supporting example services such as an Aerial Radio Access Network as well as a 3D Sensor Exploration Service. The work presented in this chapter shows that agent-based distributed intelligence is suited to perform the continuous trade-off between the competing requirements resulting from the swarm's application task and the communication-related requirements. Different channel-aware strategies for microscopic as well as macroscopic mobility algorithms have been introduced and discussed based on results, which have been produced by a dedicated multi-scale UAV network system simulation environment. Through the assignment of roles within the swarm, it is possible to cope with advanced scenarios, in which a continuous air-to-ground communication is not available. Implementation and validation experiments with UAV test systems have shown that the technology of task-driven cognitive networking for aerial robot swarms has advanced to be technically mature enough for future deployment. Nevertheless, there are many open research questions, for example, regarding the reliable networking of different UAV types and the communications-sensitive mobility control for networked heterogeneous teams of unmanned aerial and ground robots.

References

- K. Akkaya, F. Senel, A. Thimmapuram, S. Uludag, Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility. *IEEE Trans. Comput.* **59**(2), 258–271 (2010)
- I.F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks. *IEEE Commun. Mag.* **40**(8), 102–114 (2002)
- J. Allred, A. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, K. Mohseni, Sensorflock: an airborne wireless sensor network of micro-air vehicles, in

- Proceedings of the 5th international conference on embedded networked sensor systems*, Sydney, Australia, pp. 117–129, 2007
- H. Ammari, S. Das, A study of k-coverage and measures of connectivity in 3D wireless sensor networks. *IEEE Trans. Comput.* **59**(2), 243–257 (2010)
- J. Andrews, F. Baccelli, R. Ganti, A tractable approach to coverage and rate in cellular networks. *IEEE Trans. Commun.* **59**, 3122–3134 (2011)
- Y. Cao, A. Fukunaga, A. Kahng, Cooperative mobile robotics: antecedents and directions. *Auton. Robot.* **4**(1), 7–27 (1997). Springer
- X. Cheng, D. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks. *Wirel. Netw.* **14**(3), 347–355 (2008)
- H. Chung, S. Oh, D.H. Shim, S.S. Sastry, Toward robotic sensor webs: algorithms, systems, and experiments. *Proc. IEEE* **99**(9), 1562–1586 (2011)
- C.E. Corrigan, G.C. Roberts, M.V. Ramana, D. Kim, V. Ramanathan, Capturing vertical profiles of aerosols and black carbon over the Indian Ocean using autonomous unmanned aerial vehicles. *Atmos. Chem. Phys.* **8**(3), 737–747 (2008)
- K. Daniel, B. Dusza, A. Lewandowski, C. Wietfeld, AirShield: a system-of- systems MUAV remote sensing architecture for disaster response, in *Proceedings of the IEEE systems conference (SysCon)*, Vancouver, Canada, pp. 196–200, March 2009
- K. Daniel, A. Wolff, C. Wietfeld, Protocol design and delay analysis for a MUAV-based aerial sensor swarm, in *IEEE wireless communications and networking conference (WCNC)*, Sydney, Australia, pp. 1–6, April 2010
- K. Daniel, S. Rohde, N. Goddemeier, C. Wietfeld, Cognitive agent mobility for aerial sensor networks. *IEEE Sens. J.* **11**(11), 2671–2682 (2011)
- P. Dias, J. Sousa, F.L. Pereira, Networked operations (with Neptus), in *3rd annual conference on maritime systems and technologies MAST*, Cadiz, Spain, Nov 2008
- D. Driscoll, A. Mensch, *Devices profile for web services*, OASIS, Standard, 2009
- Q. Du, M. Emelianenko, L. Ju, Convergence of the Lloyd algorithm for computing centroidal voronoi tessellations. *SIAM J. Numer. Anal.* **44**, 102–119 (2006)
- S. Elayoubi, O. Ben Haddada, B. Fouresterie, Performance evaluation of frequency planning schemes in OFDMA based networks. *IEEE Trans. Wirel. Commun.* **7**(5), 1623–1633 (2008)
- J. Elston, E.W. Frew, D. Lawrence, P. Gray, B. Argrow, Net-centric communication and control for a heterogeneous unmanned aircraft system. *J. Intell. Robot. Syst.* **56**(1–2):199–232 (2009)
- J. Elston, J. Roadman, M. Stachura, B. Argrow, A. Houston, E.W. Frew. The tempest unmanned aircraft system for in situ observations of tornadic supercells: design and flight test results. *J. Field Robot.* **28**(4), 461–483 (2011)
- E. Frew, T. Brown, Airborne communication networks for small unmanned aircraft systems. *Proc. IEEE* **96**(12), 2008–2027 (2008)
- M. Garetto, M. Gribaudo, C.-F. Chiasserini, E. Leonardi, A distributed sensor relocation scheme for environmental control, in *Proceedings of the IEEE international conference on mobile adhoc and sensor systems (MASS)*, Torino, Italy, pp. 1–10, October 2007
- A. Ghaffarkhah, Y. Mostofi, Communication-aware motion planning in mobile networks. *IEEE Trans. Autom. Control* **56**(10), 2478–2485 (2011)
- N. Goddemeier, S. Rohde, J. Pojda, C. Wietfeld, Evaluation of potential fields mobility strategies for aerial network provisioning, in *Proceedings of the IEEE globecom workshop on wireless networking for unmanned autonomous vehicles*, Houston, TX, USA, Dec 2011
- N. Goddemeier, K. Daniel, C. Wietfeld, Role-based connectivity management with realistic air-to-ground channels for cooperative UAVs. *IEEE J. Sel. Areas Commun. (JSAC)* **30**, 951–963 (2012)
- X. Han, X. Cao, E. Lloyd, and C.-C. Shen, Fault-tolerant relay node placement in heterogeneous wireless sensor networks, in *Proceedings of the 26th IEEE international conference on computer communications (INFOCOM)*, pp. 1667–1675, May 2007
- Z. Han, A.L. Swindlehurst, A.L., K. Liu, K., Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles. *IEEE Trans. Veh. Technol.* **58**(7), 3533–3546 (2009)

- J. How, C. Fraser, K. Kulling, L. Bertuccelli, O. Toupet, L. Brunet, A. Bachrach, N. Roy, Increasing autonomy of UAVs. *IEEE Robot. Autom. Mag.* **16**(2), 43–51 (2009)
- A. Howard, M. J. Mataric, G.S. Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, in *Proceedings of the 4th international symposium on distributed autonomous robotic systems (DARS' 02)*, Fukuoka, Japan, pp. 299–308, 2002
- J.H. Jung, S. Park, S.-L. Kim, Multi-robot path-finding with wireless multihop communications. *IEEE Commun. Mag.* **48**, 126–132 (2010)
- A. Krause, A. Singh, C. Guestrin, Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res. (JMLR)* **9**, 235–284 (2008)
- E. Lloyd, G. Xue, Relay node placement in wireless sensor networks. *IEEE Trans. Comput.* **56**(1), 134–138 (2007)
- J. Love, J. Jariyasunant, E. Pereira, M. Zennaro, K. Hedrick, C. Kirsch, R. Sengupta, CSL: a language to specify and re-specify mobile sensor network behaviors, in *15th IEEE real-time and embedded technology and applications symposium*, pp.67–76, 2009
- N. Michael, C. Belta, V. Kumar, Controlling three dimensional swarms of robots, in *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, pp. 964–969, Orlando, Florida, USA, May 2006
- B. Niehöfer, A. Lewandowski, C. Wietfeld, Evaluation of the localization accuracy of satellite systems for traffic flow predictions, in *24th international technical meeting of the satellite division of the institute of navigation (ION GNSS 2011)*, Portland, OR, USA, Sept 2011
- J.J. Odell, H. Parunak, M. Fleischer, Temporal aspects of dynamic role assignment, in ed. by P. Giorgini, J. Müller, J. Odell. *Agent-oriented software engineering IV*, Lecture Notes in Computer Science (Springer, Berlin/New York 2004), pp. 185–214
- R. Pabst, B. Walke, D. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D. Falconer, G. Fettweis, Relay-based deployment concepts for wireless and mobile broadband radio. *IEEE Commun. Mag.* **42**(2004), 80–89 (2004)
- Z. Pengcheng, Y. Kai, A.L. Swindlehurst, Wireless relay communications with unmanned aerial vehicles: performance and optimization. *IEEE Trans. Aerosp. Electron. Syst.* **47**(3), 2068–2085 (2011)
- C. Reynolds, Flocks, herds and schools: a distributed behavioral model, in *SIGGRAPH '87: proceedings of the 14th annual conference on computer graphics and interactive techniques*, Anaheim, CA, USA, July 1987
- S. Rohde, N. Goddemeier, C. Wietfeld, F. Steinicke, K. Hinrichs, T. Ostermann, J. Holsten, D. Moormann, AVIGLE: a system of systems concept for an avionic digital service platform based on micro unmanned aerial vehicles, in *Proceedings of the IEEE international conference on systems, man, and cybernetics (SMC)*, Istanbul, Turkey, Oct 2010
- S. Rohde, M. Putzke, C. Wietfeld, Ad hoc self-healing of OFDMA networks using UAV-based relays. *Ad Hoc Networks* (Elsevier), Available online 11 July 2012, ISSN 1570-8705, 10.1016/j.adhoc.2012.06.014. (<http://www.sciencedirect.com/science/article/pii/S157087051200131X>)
- S. Rohde, C. Wietfeld, Interference aware positioning of aerial relays for cell overload and outage compensation, in *IEEE 76th vehicular technology conference*, Québec, Canada, Sept 2012
- M.N. Rooker, A. Birk, Multi-robot exploration under the constraints of wireless networking. *Control Eng. Pract.* **15**(4), 435–445 (2007)
- A. Ryan, M. Zennaro, A. Howell, R. Sengupta, J.K. Hedrick, An overview of emerging results in cooperative UAV control, in *43rd IEEE conference on decision and control*, pp. 602–607, Dec. 2004
- J. Schmutzler, S. Gröning, C. Wietfeld, Management of distributed energy resources in IEC 61850 using web services on devices, in *2nd IEEE international conference on smart grid communications (SmartGridComm)*, Brussels, Belgium, Oct 2011
- J. Schneider, T. Kamiya, Efficient XML Interchange (EXI) Format 1.0, W3C Std., 2011

- D.F. Spears, D.R. Thayer, D.V. Zarzhitsky, Foundations of swarm robotic chemical plume tracing from a fluid dynamics perspective. *Int. J. Intell. Comput. Cybern.* **2**(4), 745–785 (2009)
- E. Stump, A. Jadbabaie, V. Kumar, Connectivity management in mobile robot teams, in *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, Pasadena, USA, pp. 1525–1530, May 2008
- S. Subik, C. Wietfeld, Integrated PMR-broadband-IP network for secure realtime multimedia information sharing, *IEEE international conference on technologies for homeland security (HST)*, Boston, USA, pp. 20–25, Nov 2011
- G. Tan, S.A. Jarvis, A.-M. Kermarrec, Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks, in *The 28th international conference on distributed computing systems ICDCS*, pp. 429–437, June 2008
- J. Tisdale, A. Ryan, M. Zennaro, X. Xiao, D. Caveney, S. Rathinam, J.K. Hedrick, R. Sengupta, The software architecture of the Berkeley UAV Platform, in *IEEE conference on control applications*, Munich, Germany, pp. 1420–1425, Oct 2006
- G. Wang, G. Cao, T. La Porta, W. Zhang, Sensor relocation in mobile sensor networks, in *24th annual joint conference of the IEEE computer and communications societies INFOCOM*, Miami, USA, pp. 2302–2312, March 2005
- G. Wang, G. Cao, T.L. Porta, Movement-assisted sensor deployment. *IEEE Trans. Mobile Comput.* **5**(6), 640–652 (2006)
- B. White, A. Tsourdos, I. Ashokaraj, S. Subchan, R. Zbikowski, Contaminant cloud boundary monitoring using network of UAV sensors. *IEEE Sens. J.* **8**(10), 1681–1692 (2008)
- C. Wietfeld, H. Georg, S. Gröning, C. Lewandowski, C. Müller, J. Schmutzler, Wireless M2M communication networks for smart grid applications, in *Proceedings of the European wireless 2011 (EW2011)*, Vienna, Austria, pp. 275–281, April 2011
- A. Wolff, S. Michaelis, J. Schmutzler, C. Wietfeld, Network-centric middleware for service oriented architectures across heterogeneous embedded systems, in *The eleventh international IEEE EDOC conference workshop*, Annapolis, MD, USA, pp. 105–108, October 2007
- M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks: a survey. *Ad Hoc Netw.* **6**(4), 621–655 (2008). Elsevier
- Y. Zou, C. Krishnendu, Sensor deployment and target localization based on virtual forces, *24th annual joint conference of the IEEE computer and communications societies INFOCOM*, San Francisco, USA, pp. 1293–1303, March 2003

Jack Elston, Maciej Stachura, Cory Dixon, Brian Argrow,
and Eric W. Frew

Contents

| | | |
|--------|---|-----|
| 34.1 | Introduction | 782 |
| 34.2 | Physical/Transport | 784 |
| 34.2.1 | IEEE 802.11 Radio | 786 |
| 34.2.2 | IEEE 802.15.4 Radio | 786 |
| 34.2.3 | 900-MHz Spread-Spectrum Point-to-Point Radios | 787 |
| 34.2.4 | Satellite Communications | 787 |
| 34.3 | Data Routing and Network Configuration | 789 |
| 34.4 | Sensor, Communication, and Control Fusion | 791 |
| 34.5 | Service Implementation and Discovery | 794 |
| 34.5.1 | Application Layer Networking Components | 794 |
| 34.5.2 | Service Discovery | 796 |
| 34.5.3 | Service Subscription and Command Interface | 798 |
| 34.6 | Mission-Level Control | 799 |
| 34.6.1 | Human Operator Input | 799 |
| 34.6.2 | High-Level Control Algorithms | 802 |
| 34.7 | VORTEX2 Implementation | 803 |
| 34.7.1 | VORTEX2 Experiment and Results | 805 |
| | References | 809 |

J. Elston (✉) • M. Stachura • E.W. Frew
Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, USA
e-mail: jack.elston@colorado.edu; maciej.stachura@colorado.edu; stachura@colorado.edu;
eric.frew@colorado.edu

C. Dixon
College of Engineering and Applied Science, University of Colorado at Boulder, Boulder,
CO, USA
e-mail: cory.dixon@colorado.edu

B. Argrow
Director Research and Engineering Center for Unmanned Vehicles, University of Colorado,
Boulder, CO, USA
e-mail: brian.argrow@colorado.edu

Abstract

Different networking hardware, protocols, and sensors can be combined to create a diverse and complex unmanned aircraft system through a layered design approach with modular supporting software. A layered design simplifies both testing and system reconfiguration, lending itself to incremental verification while only requiring the maintenance of standard interfaces. Critical software components, such as service discovery, simplify the inclusion of a diverse set of subsystems and sensors. Maintaining the modularity of these software components ensures that the system can be expanded while requiring minimal software changes. An example of these design approaches is provided through the description of a system that enabled flight operations of a multi-vehicle unmanned aircraft system for performing targeted, in situ sampling of supercell thunderstorms during the 2010 VORTEX2 field campaign. This network's flexible nature facilitated the complex interworking of many subsystems while accommodating the strict deployment time constraints and generally unknown starting configurations typical of nomadic field operations. Data required for situational awareness and mission-level decision making was retrieved from multiple remote and local sources, and mechanisms were provided for dissemination to any local participant. The architecture's modular design allowed for changes in the field and actively supported the addition and subtraction of nodes without requiring reconfiguration of the remainder of the system.

34.1 Introduction

Most unmanned aircraft (UA) operations are conducted using a single aircraft and rely on simple point-to-point static network configurations (Hipskind et al. 2002; Halverson et al. 2005; Lin and Lee 2008; Office of the Secretary of Defense 2009). This configuration is sufficient to meet the needs of the scientific, operational, and regulatory requirements for many applications. However, as unmanned aircraft systems (UAS) continue to advance and their applications are expanded, there is a need to provide communication beyond a single command and control link. This need is not solely restricted to operation of multiple aircraft (Tisdale et al. 2009; How et al. 2004; Beard et al. 2006) but also includes applications where ground vehicles or stationary nodes are involved in the system and must maintain their own communication requirements.

There are compelling reasons to have sensing systems composed of multiple vehicles with very different capabilities in communication range and bandwidth, sensor payloads, and onboard computational power. Bird-sized miniature UA have significant advantages in unit cost and safety of operation. However, they are burdened with limited payload size/mass, low flight speed/duration/range, limited computational resources, and small communication range and data bandwidth. These aircraft are well suited to multiple, dispersed, short-range sensing tasks, where attrition is expected, or the aircraft are considered disposable. However, such a flock or swarm cannot self-deploy to distant locations, self-organize to

perform complex tasks, or communicate data from distant regions. The addition of a few aircraft with greater capabilities, e.g., small UA, provides a system with the combined advantages of both platforms.

Communication is further complicated when applied to these heterogeneous unmanned aircraft systems. The vastly different payload capacity, avionics capability, communication range, and flight endurance limitations of unmanned aircraft classes lead to fundamentally different types of sensing and communication. For example, swarming miniature air vehicles will likely use local flooding schemes over short distances to disseminate data (Allred et al. 2007), while larger aircraft will use mobile ad hoc and delay-tolerant networking in conjunction with non-line-of-sight radio transmission. Combining these heterogeneous capabilities into a single system with robust functionality remains an open challenge.

Even with single vehicle operations, there has been an increasing need to interface vehicle subsystems with the communication link. In this manner, end users and remote systems can interact with particular subsystems to achieve mission goals. This is typically performed through a dedicated payload link, where sensing data is kept separate from vehicle command and control. This is not always an advantageous configuration, as it is difficult to expand beyond a point-to-point communication link. Furthermore, certain applications require the combination of command and control with payload information to optimize sensing. Examples include various applications of adaptive sampling (Leonard et al. 2010), the construction of communication-based relay chains (Dixon 2010), and the characterization and localization of WiFi sources (Stachura and Frew 2011).

Data-centric, targeted, in situ meteorological sampling is an example application for a complex UAS that requires the use of a heterogeneous, multi-node network that combines inter- and intra-vehicle communications. Conducting targeted sampling is particularly challenging for severe local storm phenomena given the large number of participating vehicles and widely distributed data sources for situational awareness (Elston et al. 2011). This configuration differs from previous uses of UAS for meteorological sampling (Hipskind et al. 2002; Halverson et al. 2005; Lin and Lee 2008) which typically occur over water with observations made over a period of days. Challenges created by the fast evolution of local storms are compounded with the differences in the regulatory environment for the rapid deployment of small UAS over land (Frew et al. 2012). These challenges led to the complex network and system design for the Tempest UAS fielded in 2010 during the second Verification of the Origins of Rotation in Tornadoes Experiment (VORTEX2) (Elston et al. 2011).

To address the difficulties associated with communications in complex UAS, the development of an inter- and intra-vehicle network has been decomposed using a bottom-up layered design approach (Fig. 34.1). This approach allows for the design of UA and supporting systems to reflect and enhance design decisions made at the lower layers. Given that the success of a UAS is based heavily on networked communications, this approach ensures the mission-level control algorithms can be implemented on the underlying network architecture. Subsequent decisions at higher levels of the design process are based in part on the decisions made below. The layers designed here include: *physical/transport*; *data routing*

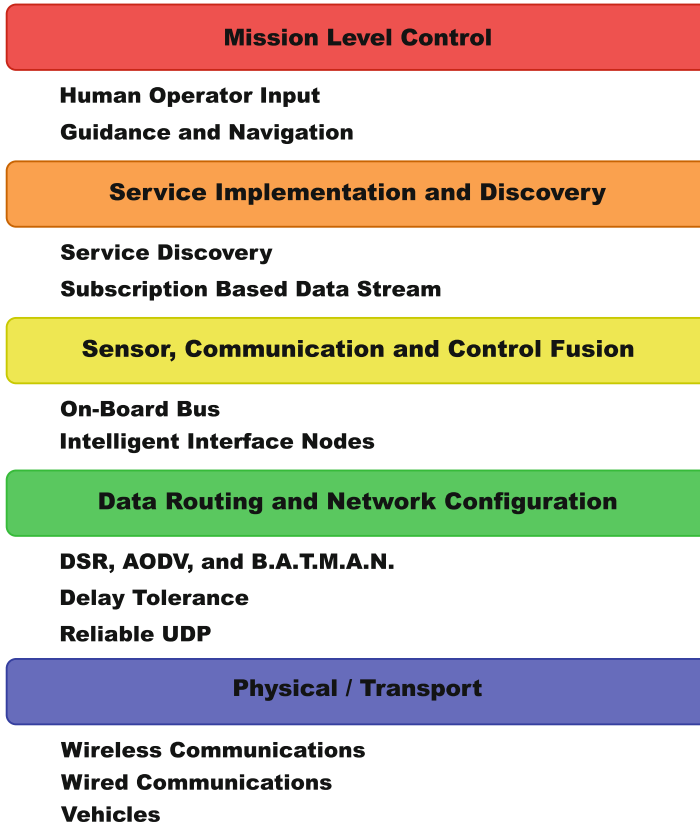


Fig. 34.1 Bottom-up design approach to hardware, communication, sensing, and control for networked UAS

and network configuration; integration of intra-vehicle communication for sensor, communication, and control fusion; application layer communication protocols for service implementation and discovery with data-stream publication; and mission-level control.

34.2 Physical/Transport

The *physical/transport* layer (Fig. 34.1) is the base layer that is directly responsible for moving data. This layer is comprised of three distinctly different transport media: mobile vehicles, wired communications, and wireless radios. Typically, design of the subsystems in the *physical/transport* layer is driven by the desire to use commercial off-the-shelf (COTS) hardware as much as possible to allow rapid development and easy deployment. Figure 34.2 contains examples of possible components and their associated protocols.



Fig. 34.2 The various transport methods supported by the network-centric architecture and resulting software

A complex UAS will use a large number of transport methods since many subsystems such as autopilots, sensors, ground stations, end users, etc., will need to share data or issue commands. Providing a system that seamlessly accommodates a large number of transport methods is critical to an adaptive and modular network-centric architecture and presents a significant challenge. It is also necessary to properly abstract this layer from the rest of the network design since these subsystems will typically change for different mission requirements and evolve over time.

Each transport method has its unique benefits and challenges. Mobile vehicle-based transport involves coupling onboard storage with mobility to transport large amounts of data from one geographical location to another, but requires the use of delay-tolerant networks (Brown and Henkel 2006). Wired communications systems are typically limited to intra-vehicle connections, but generally provide highly reliable transmissions. Wireless communications subsystems are required for UAS and can greatly influence system performance. Therefore, wireless communications are typically the focus of the design of the *physical/transport* layer for most UAS applications.

To aid in the wireless subsystem selection process, several of the more popular communications methods are presented here, all of which have been used with the NetUAS system. These include IEEE 802.11 (WiFi) and IEEE 802.15.4 (ZigBee) radios, 900-MHz modems with both RS-232 and IEEE 802.3 (Ethernet) interfaces, and Iridium satellite communications with an RS-232 interface. These systems are typically used in small UAS for their relatively small size and weight, low power

Table 34.1 Specifications for the EnGenius EMP-8602 PLUS-S

| | |
|-----------------------|------------------------------------|
| EMP-8602 PLUS-S | |
| Operating frequency | 2.4000–2.484 GHz (802.11b) |
| Output power | 100 mW (802.11b) |
| Range | up to 140 m (802.11b) |
| Configurations | Ad hoc, managed, master, secondary |
| Data rate | Up to 11 Mbps (802.11b) |
| Interface protocol | Mini PCI |
| Operating voltage | 3.3 VDC |
| Operating temperature | 40–85 °C |
| Dimensions | 59.6 × 44.45 mm |
| Weight | 15 g |

draw, and ability to operate at limited radiated power levels without a license. Each communications method utilizes different network topologies including multi-hop, point-to-point, and centralized, and the system design must be able to accommodate communications through these different schemes. Furthermore, selection of communications hardware will be restricted by the resources, dynamics, and sizes of the nodes in the network.

34.2.1 IEEE 802.11 Radio

IEEE 802.11 offers an ad hoc mode which allows for any single device to connect to others in proximity without requiring a centralized access point. This behavior is desirable in UAS where network nodes might frequently move in and out of range of other nodes. Furthermore, the 802.11 protocol allows high-bandwidth connections, for large amounts of data to be sent over the links. Unfortunately, without amplification, the radiated power of a COTS radio covers only a relatively short distance. Generally, 802.11 cards containing an Atheros chipset are a good choice because of the level of driver support in Linux, a commonly used embedded operating system for UAS avionics and ad hoc networks. One frequently used card is the EMP-8602 PLUS-S from EnGenius (EnGenius Tech Inc 2011) (Table 34.1). Characterization of the performance of a similar card in a UAS network was performed during the Ad hoc UAV Ground Network (AUGNet) experiment (Brown et al. 2007).

34.2.2 IEEE 802.15.4 Radio

In smaller UAS, where weight and power savings are paramount, a protocol like IEEE 802.15.4 is commonly used (see specifications in Table 34.2). This protocol is limited in bandwidth compared to 802.11, but its lower rate allows for lower power, and for implementation on much simpler devices (Zheng and Lee 2006). Furthermore, the radios implement the ad hoc on-demand distance

Table 34.2 Specifications for the Digi XBee-Pro radio

| | |
|-----------------------|----------------------------------|
| XBee-PRO | |
| Operating frequency | 2.4000–2.4835 GHz |
| Output power | 63 mW |
| Range | 90 m |
| Configurations | Master, slave, or slave/repeater |
| Data rate | 250 Kbps |
| Interface protocol | 3 VDC CMOS UART |
| Operating voltage | 2.8–3.4 VDC |
| Operating temperature | 40–85 °C |
| Dimensions | 24.38 × 3.294 mm |
| Weight | 3 g |

vector (AODV) routing protocol in hardware, allowing for mesh networks to be formed without special software. This protocol has been used with micro UA projects (Allred et al. 2007) and has been installed in larger UAS providing gateways to other wireless networks to enable participation of micro UA in larger scenarios (Elston et al. 2009b).

34.2.3 900-MHz Spread-Spectrum Point-to-Point Radios

For several different scenarios involving small UAS, a longer-range solution is necessary. Even with amplification and steerable antennas, the range of COTS 802.11 components remains limited by the relatively high frequency of its carrier. Although 900-MHz radio modems generally only offer point-to-point or centralized communications schemes, and are limited in bandwidth, they greatly increase the communication range over 2.4-GHz options in the unlicensed industrial, scientific, and medical (ISM) radio bands. The setbacks in the routing can be overcome with application layer networking software running on a computer paired with each modem. To save weight on the UA, smaller OEM models of the radios can be used such as the FreeWave MM2. At the ground station or in ground vehicles where power and weight requirements are significantly relaxed, a model such as the HTPlus could be used, which contains significantly more functionality. Detailed specification of these units can be found in Table 34.3.

34.2.4 Satellite Communications

For significantly long-range missions, satellite communications are required. These are typically expensive, unreliable, and very low bandwidth. However, given a sufficiently autonomous aircraft with a relatively static flight pattern, this small amount of communication can be sufficient to perform a mission. Before the exact range of operations required for supercell sampling missions was known, the Tempest UA was equipped with the A3LA-X Iridium modem from Nal Research

Table 34.3 Specifications for the FreeWave MM2 and HTPlus radios

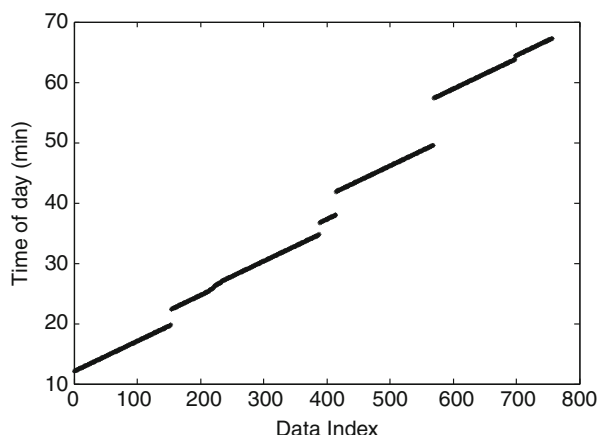
| | |
|-----------------------|---|
| HTPlus | |
| Operating frequency | 902–928 MHz |
| Output power | 5 mW up to 1 W |
| Range | Point-to-point Up to 48 km Point-to-multipoint 24 km |
| Configurations | Master, slave, or slave/repeater |
| Data rate | Up to 867 Kbps |
| Interface protocol | IEEE 802.3 or RS232 |
| Operating temperature | –40 to 60 °C |
| Dimensions | 165 × 74 × 59 mm |
| Weight | 427 g |
| MM2 | |
| Operating frequency | 902–928 MHz |
| Output power | 5 mW up to 1 W |
| Range | Up to 96 km |
| Configurations | Master, slave, or slave/repeater |
| Data rate | Up to 153.6 Kbps |
| Interface protocol | IEEE 802.3 or RS232 |
| Operating voltage | 3.3–5.5 VDC |
| Operating temperature | –40 to 85 °C |
| Dimensions | 36 × 50.8 mm |
| Weight | 14 g (TTL version) |

Table 34.4 Specifications for the Nal research A3LA-X satellite modem

| | |
|-----------------------|--------------------|
| A3LA-X | |
| Operating frequency | 1,616–1,626.5 MHz |
| Output power | 0.6–7 W |
| Range | Works globally |
| Configurations | Slave |
| Data rate | 2.4 Kbps |
| Interface protocol | RS232 |
| Operating voltage | 4.0–32.0 VDC |
| Operating temperature | 30–70 °C |
| Dimensions | 162 × 67 × 27 mm |
| Weight | 340 g (0.75 pound) |

(see specifications in Table 34.4). Due to the regulations required for operating UA in the U.S. National Airspace System (Davis 2008), reliability became an issue for ensuring safe operations and a different solution was developed combining IEEE 802.11 and 900-MHz radios. Figure 34.3 shows the timeouts experienced during one mission, and Fig. 34.4 compares the amount of telemetry received through the modem’s limited bandwidth to a point-to-point 900-MHz link.

Fig. 34.3 Index of each data packet vs. the time of day it was received at the ground station using the Iridium modem. Note the four timeouts, around 20, 35, 38, and 50 min. At these points, the modem lost communications and the ground station was forced to reestablish the connection with the UA, a process that can take several minutes



34.3 Data Routing and Network Configuration

The *data routing and network configuration* layer (Fig. 34.1) is built on top of the COTS components used in the transport layer and is used to determine how to deliver data between nodes. For several of the network connections, it can be sufficient to use manufacturer-provided routing and configuration methods. However, for wireless connections to the aircraft and between the ground vehicles, most scenarios dictate the use of unique or modified protocols that can handle multi-hop connections and network topology reconfiguration Christmann and Johnson (2007). The managed mode of 802.11 fails quickly in these scenarios, and the ad hoc mode fails to provide for routing beyond single-hop communications (Brown et al. 2007). There is significant value in fielding a true multi-hop, ad hoc solution when deploying systems consisting of multiple cooperating vehicles, and several projects have employed these architectures (Brown et al. 2007; Elston et al. 2009b; Allred et al. 2007). Providing this unique environment generally requires the use of third-party routing protocols. Presented below are protocols that have been used with the NetUAS system along with the particular experiment each was employed in.

During the flights performed for the AUGNet experiment, an 802.11b card was combined with a Linux single-board computer and GPS to construct a mesh network radio (MNR) (Brown et al. 2007). To simplify development, the core of each MNR node for this experiment was identical, with only the packaging differing depending on whether the MNR is mounted at a fixed ground site, on a ground vehicle, or in an aircraft. Each MNR ran the *dynamic source routing* (DSR) protocol (Johnson et al. 2001): an on-demand protocol that only seeks a route from source to destination when data must be sent. Thus, radio nodes do not waste bandwidth trying to establish routes they do not need at the moment. When a node needs to send a packet, it initiates a route request process among nodes in the network to establish a route. DSR uses source routing whereby a packet source

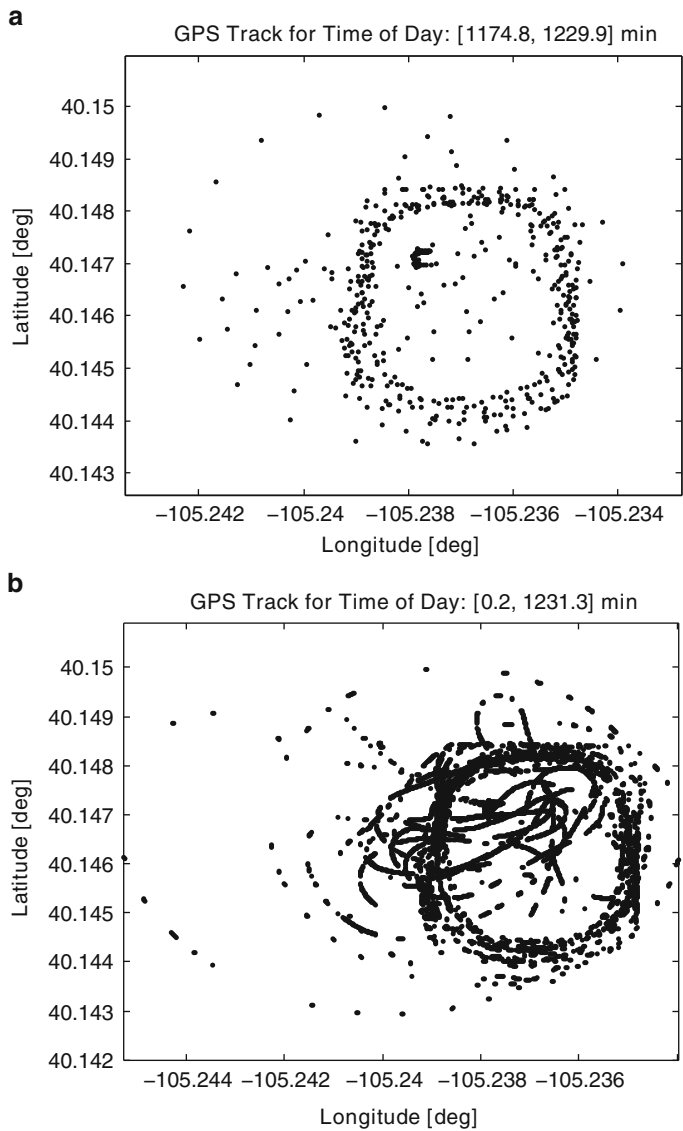


Fig. 34.4 The difference in spatial sampling with the (a) lower data rate from the Iridium and (b) higher data rate from the 900-MHz modem. Notice the reduced data points and missing sections in the stream transmitted via the Iridium modem

precisely specifies in the header packet which route the packet will follow. DSR was implemented using the Click modular router (Kohler et al. 2000), a software-defined protocol that enables the modification of the routing scheme as needed. During the experiment, it was determined that timeouts and hop lengths had to

be changed from the protocol's default values to accommodate the combination of node motion with the communications ranges from COTS 802.11 cards. However, even with these modifications, DSR was found to fail more often in highly mobile networks, especially when transmitting large amounts of data (Brown et al. 2007).

The *ad hoc on-demand distance vector (AODV)* (Chakeres and Belding-Royer 2004) protocol was used in the Collaborative Colorado-Nebraska Unmanned Aircraft System Experiment (CoCoNUE) to combat the issues encountered during the AUGNet experiment (Elston et al. 2009a). Unfortunately, although the AODV implementation created by CoReSoftware (AODV-UU 2010) provided the desired functionality in the lab and during some flight experiments, its use in the field uncovered several problems. Most of the issues came from the lack of updates since 2007, which mandated the use of older kernels and drivers. At that time, no other implementations were available, so the group was forced to use another protocol. AODV could still be a suitable protocol for use in UA networks, especially given that a hardware implementation of AODV is employed by some modes of operation in 802.15.4 radios. For IP-based networks, the use of Click could provide for an acceptable alternative to the CoReSoftware package, as at least one example of an implementation of AODV in Click already exists (Performance Analysis of Telecommunication Systems 2012).

An implementation of the *better approach to mobile ad hoc networking (B.A.T.M.A.N.)* protocol (B.A.T.M.A.N. 2010) was used to replace the AODV implementation in the CoCoNUE project (Elston et al. 2009a; Houston et al. 2011). This protocol was chosen based on the results of characterization of multiple different routing protocols (Abolhasan et al. 2009). In several experiments conducted in both the laboratory and in the field using mobile and stationary nodes, the functionality of B.A.T.M.A.N. was verified for identifying and properly selecting routes. It was later employed in the complex network system fielded in VORTEX2 (Elston et al. 2011), and provided effective routing despite the large number of participating nodes.

For certain UAS applications, suitable third-party implementations do not exist. In these cases, routing protocols must be designed to address the unique networking issues. As mentioned in Sect. 34.2, work has been performed to investigate the use of the vehicles themselves as data ferries. It considers optimal mobility and delay-tolerant networking to provide maximum bandwidth in a disconnected network (Brown and Henkel 2006). For UA networks with limited uptime given power or other considerations, work has been performed on modifying the UDP protocol to provide data delivery guarantees without the overhead required by TCP (Henkel et al. 2006).

34.4 Sensor, Communication, and Control Fusion

The *sensor, communication, and control fusion* layer (Fig. 34.1) integrates intra-vehicle communication between subsystems with the external meshed networking environment. In the NetUAS system, a modular architecture has been implemented

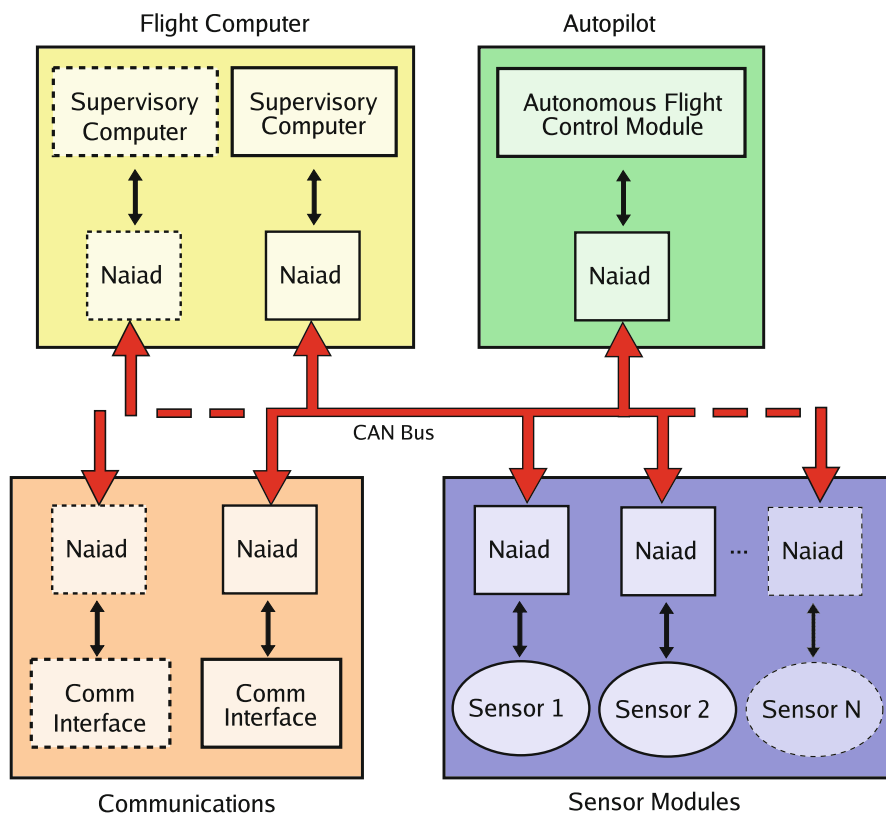


Fig. 34.5 Distributed onboard network architecture. *Dashed lines* indicate areas of system expansion, and *shaded blocks* represent subsystems

for the synthesis of onboard systems needed in mission-level decision making. The architecture employs a lightweight interface node to connect to each component, and provide an interface through a shared bus. Each node contains some intelligence and is responsible for initialization, data fetching, and data manipulation needed for a particular device. By enforcing this paradigm, the bus traffic remains high level and enables significant system scaling with no reprogramming of the existing nodes. Particular subsystem upgrades may also be performed with ease, assuming the high-level data being pushed to the bus by its interface node remains unchanged. A typical configuration consists of a supervisory flight management computer, the autonomous flight system, extra-vehicle communication hardware, and sensors (Fig. 34.5).

Using this distributed architecture, better fault tolerance and flexibility is achieved than by using a centralized approach. Without the hardware limitations of a single-board solution, only the inter-node bus protocol limits the number of times subsystems and interfaces can be replicated. The use of addressing-by-type allows for multiple nodes to both send and receive the same message type, and enables

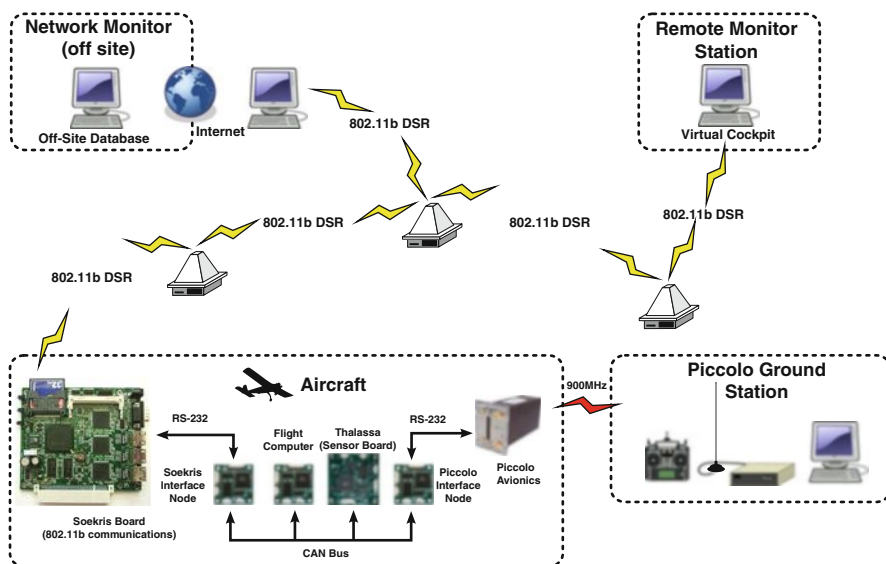


Fig. 34.6 Typical deployment scenario for the ares UA networked communication, command, and control system. The aircraft electronics consist of a distributed avionics system connected through a controller area network (CAN) bus using the Naiad interface nodes (Elston et al. 2009b)

transmission without knowing the hardware address of a destination node. Combining this functionality with the use of a health-and-status subsystem to identify and disable malfunctioning nodes makes changing between redundant subsystems transparent to the rest of the system. It provides a level of fault tolerance equivalent to the nearly independent subsystems of a federated architecture while maintaining the advantages of modular systems.

The integration of intra- and inter-vehicle communication plays a significant, albeit subtle, role in the performance of the unmanned aircraft system. By connecting subsystem information, e.g., telemetry from an autopilot or images from an onboard sensor to the external meshed network, the aircraft can exchange data directly to the subsystems that need it (Fig. 34.6). An example includes the interface of an ad hoc communications system to the autopilot to provide greater ground-station-to-UA communication range through a multi-hop connection, or to allow for control hand off between multiple remote operators via a mesh network. Including payloads in the intra-vehicle bus allows network participants to view mission-specific data generated from anywhere within communication range. This becomes very important when considering cooperative tasks shared between multiple vehicles.

Within NetUAS, the Naiad is the first incarnation of the interface nodes used to implement this architecture (Elston et al. 2005). A Naiad node is approximately 1.5 in. on each side, weighs about 0.5 oz, and nominally uses 50 mA at 5 V. The node contains a watch-dog timer with brown-out detection. The microcontroller is capable of about 14-MIPS throughput, but requires very little power to operate and

is relatively inexpensive. Each node has mounting holes to enable easy placement about the aircraft, or to be stacked in a convenient location using small standoffs. Inter-node communications are made through the message type-based controller area network (CAN) protocol. Each node contains a fault-tolerant CAN transceiver to guarantee data transmission and allows the system to run on a single wire in the case of a separated bus line. Furthermore, the use of high differential voltage levels by the transceiver protects the bus from interference.

34.5 Service Implementation and Discovery

Components at the *service implementation and discovery* layer (Fig. 34.1) define how elements of the UAS can interact with one another, enabling seamless command and control across a heterogeneous network. This layer accommodates functionality such as the use of more than one ground station, interaction with the UA from outside of the ad hoc network, in flight reconfiguration of the entire UAS, and communication between different UA types and heterogeneous subnetworks. Components at this layer ensure that network configuration (e.g., the existence of nodes and their addresses) no longer has to be specified before the system is deployed. New UA can be injected into the system or new ground stations may be brought up, and the network configures itself appropriately in response. All of this functionality is enabled by mechanisms for service discovery, data-stream subscription, and command issuance.

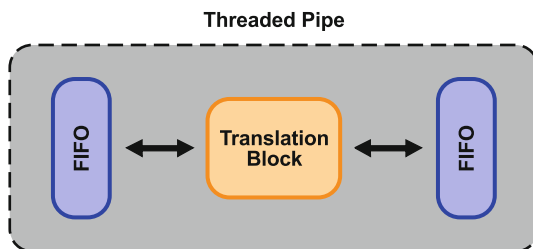
In the NetUAS system, the *service implementation and discovery* functionality is implemented by several user-level threads running on a Linux computer. These threads not only provide interfaces to the network and system components but also provide support functionality such as message logging and a graphical interface for users to either visualize available data streams or inject control messages into the UAS. Components are programmed in a way to provide a simple application programming interface (API) for the development of cooperative algorithms based upon the information handled by the underlying mechanisms. The software maintains a level of modularity that allows for reliability, ease of upgrading, and the ability for programmers to contribute functionality without needing to understand all of the components in the system.

34.5.1 Application Layer Networking Components

The application layer networking software uses an object-oriented approach to define component modules. These modules are based upon the idea that any task implemented on a particular node can be represented by a combination of up to three different types of blocks. These blocks are as follows:

- *Communication Block.* The communication block provides an interface to some physical device. It might be a device on a serial port, a socket, or a video camera. It is responsible for initialization, reading from, and writing to the device.

Fig. 34.7 The *threaded pipe* object. The block in the *center* can be instantiated as a communication, translation, or computation thread



- *Translation Block*. Considering the non-standard internal message format, most device interfaces require translation from the device's protocol to the internal messaging scheme. Two examples are XML parsing/serialization from a socket interface or conversion of proprietary binary GPS data from a serial interface.
- *Computation Block*. Occasionally the interface will also require that some computation is performed using the data made available by the interface/translation block combination. An example of this could be a filter implemented to condition the data coming in from a sensor. There are also cases where internal data needs to be processed and then pushed back to the system (i.e., no interface or translation blocks).

All three of these blocks are specialized instantiations of a single class. This class is implemented as a thread, allowing each block to perform its function asynchronously with the remainder of the system. Because of their asynchronous nature, data transfer between a series of blocks is performed through first-in-first-out buffers (FIFOs). A *threaded pipe* comprises a combination of FIFOs and a block. The *threaded pipe* is the basis for application-based networking and is shown in Fig. 34.7.

The combination of up to three of these *threaded pipes* (communication, translation, and computation), connected through their FIFOs, creates an *Interface*, shown in Fig. 34.8. Each one of these interfaces completely defines the mechanisms necessary to implement a particular capability. Examples include providing an interface to a network, reading from a sensor, actuating a controller, or providing decision making about a specific task based upon system parameters.

A critical component of the *interface* is the *client list manager*. This block is used by interfaces that provide communications to maintain a list of clients whose traffic has been seen on the network. It provides the ability to store information about each client that is frequently used on a system-wide basis, including service discovery information, last known GPS location, time since last received transmission, etc. It can store multiple addresses for a particular node, which is required to implement a gateway between two different types of networks.

By linking two or more of these interfaces together, and sharing the client list between each, data can be transferred from one interface, such as a sensor, to

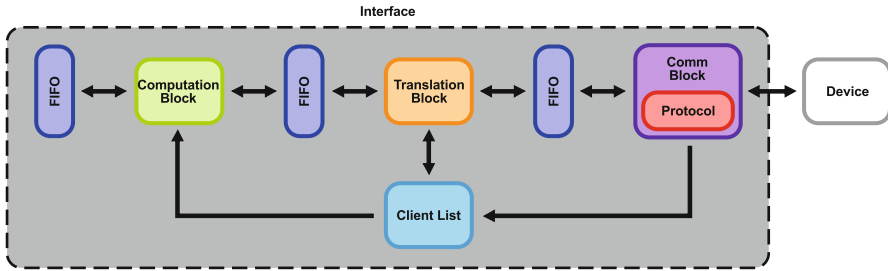


Fig. 34.8 An *interface* object, which enables a device to communicate with the remainder of the UAS. The interface is composed of up to three different *threaded pipes* (computation, translation, and communication) and a client list stored in shared memory

another interface, such as a socket to an ad hoc network (Fig. 34.9a). The intelligent routing of messages between interfaces is performed by a *data distribution* object. This object uses a destination address associated with each message and looks it up in the client list to determine the proper interface to send the message. All internal messages have to go through the *data distribution* object, and it is therefore used to provide logging functionality when desired.

Service discovery implementation is performed through the *capabilities manager* object. As each *interface* is invoked, it registers the capabilities it enables (autonomous flight, gateway to external networks, etc.) with the *capabilities manager* object (Fig. 34.9a). This process of collecting the various capabilities allows for the complete specification of all data streams available for subscription and all accepted commands. This information is broadcast over the entire network at a fixed interval providing the basis for the service discovery mechanism.

Combining the *capabilities manager*, *data distribution* object, and some multiple of *interface* objects composes a *network appliance* as shown in Fig. 34.9. A large number of interfaces may be added to the *network appliance* enabling more functionality for the particular node. Complicated nodes can be managed in this way, with the *network appliance* providing interfaces to multiple types of networks and translating between different addressing schemes or data transport methods. A specific example of this would involve a node that takes in XML command packets from an 802.11 network, parses the XML and translates the source address, then forwards it to an 802.15.4 network with the proper packet-based format and addressing scheme. Another example is managing the complex interfaces to provide a human operator system-wide status through a GUI.

34.5.2 Service Discovery

In order to facilitate self-coordination in a complex UAS, a significant amount of infrastructure must be in place. A vital component of this infrastructure is the ability of other UA to obtain information about the status and capabilities of

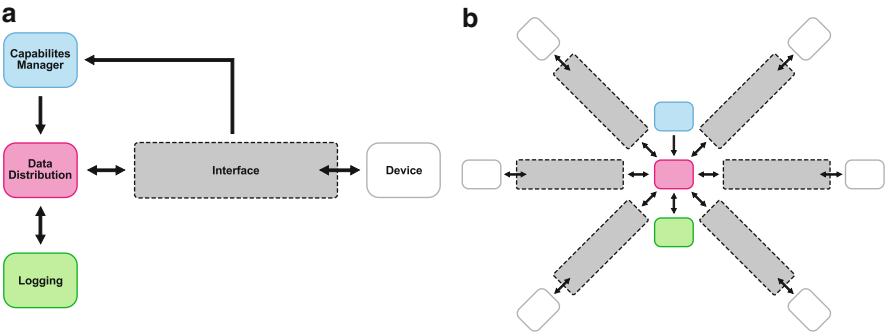


Fig. 34.9 The *network appliance* object. (a) Depicts a simple, one-device *network appliance*, where (b) shows how this can easily be expanded to accommodate a large number of devices

Table 34.5 Comparison of service discovery protocols

| | Distributed directory | Central directory | Flat | Hierarchical | Local cache | Unicast | Multicast | Broadcast | Push | Pull |
|-------------|-----------------------|-------------------|------|--------------|-------------|---------|-----------|-----------|------|------|
| DEAPspace | X | | X | | X | | | X | X | |
| UPnP | X | | X | | X | X | X | | X | X |
| Konark | X | | X | | X | | X | | X | X |
| PDP | X | | X | | X | | | X | X | X |
| Salutation | | X | | X | | X | | | | X |
| Jini | | X | | X | | X | X | | | X |
| SLP | | X | X | | | X | X | | X | X |
| Sailhan2005 | X | X | | X | | X | X | | | X |
| DSDP | X | X | | X | | X | X | | | X |
| VIA | X | X | | X | X | X | X | | | X |
| Superstring | X | | | X | | X | | | | X |
| GSD | X | | X | | X | X | | X | X | X |
| ALLIA | X | | X | | X | X | | X | X | X |
| Twine | X | | X | | | X | X | | | X |
| Pastry | | X | X | | | X | | | | X |

surrounding aircraft. It is only when this information is known that aircraft may then employ neighboring assets to collaboratively complete mission objectives. The concept of service discovery is well established; however, few service discovery protocols exist for ad hoc networks. Many of the protocols still have no method for dealing with high levels of node mobility which typically leads to sporadic disconnections and the need for continual updates of routing information. Furthermore, many protocols might be able to adapt to a highly dynamic environment, but were designed with another architecture in mind and not tested in such a manner. A summary of some existing protocols is given in Table 34.5, and further detailed in Elston (2011). Each contains some component that makes it unsuitable for highly dynamic ad hoc networks, and more significantly, no protocol allows for the easy definition

of services typically used in a UAS. Most were designed to establish the presence of services such as file sharing or a networked printer on a local area network.

A service discovery protocol was implemented from scratch, as no satisfactory third-party solution was available. To define each vehicle's services, a description language was implemented which combines the vehicle's subsystem definitions with system-level information. The functionality of the vehicle's subsystems are broken into two components: data that it can transmit to clients on a periodic basis (data streams) and commanded actions or configuration settings that each subsystem accepts (accepted commands). System-level information includes the network address of any onboard communication nodes, a unique vehicle ID to enable translation between different addressing schemes at gateway nodes, and a subset of the vehicle's telemetry for higher-level queries, such as when looking for "Neighboring nodes containing a GPS."

The *capabilities manager* is in charge of implementing the service discovery protocol. Each interface registers its capabilities with this block by indicating the particular subsystem type it represents and which streams and commands are associated with it. After 10 s has elapsed, the computation block assembles all known subsystem capabilities with the system-level information into a small packet that is broadcast to all connected networks. This heartbeat message enables other nodes to track the connectivity status of their neighbors and identify their various associated capabilities. Since this packet is reconstructed each time it is sent out, capabilities can be changed in real time, effectively accommodating system failure or reconfiguration.

34.5.3 Service Subscription and Command Interface

After the reception of a capabilities packet, a node has all the information it needs to either issue a command to a particular subsystem, or subscribe to the data provided by it. Command issuance is performed through the transmission of a request addressed using the system-level data stored in the client list. The request includes the unique subsystem ID to be commanded, the accepted command ID, and the desired actions. The method of confirmation for successful reception and execution of the command varies by subsystem. This is typically performed through the transmission of a simple ACK or NACK packet; however, some systems might require more complex responses. An example would be acknowledgement of the reception of a flight plan by echoing the entire plan back to the sender. Each subsystem is responsible for maintaining its own acknowledgement protocol, along with rules for retransmission assuming no acknowledgement is received.

Stream subscription is conducted through the transmission of a request containing the subsystem ID defined in the capabilities packet and a "Subscribe" message. Upon reception of this request, the remote node's *data distribution* object directly handles each subscription. Subscription status is indicated by adding the unique ID of the subsystem and its associated stream the client wishes to receive to the info in the client list. An ACK or NACK packet is then sent back to the subscriber

indicating its subscribed status. Data transmission to the subscriber occurs whenever the *threaded pipe* representing the subsystem of interest decides to publish data. To determine recipients, the *threaded pipe* requests the list of subscribers from the *data distribution* object. When no clients have subscribed to the data, no data is transmitted, preserving network bandwidth for other uses. At any time, a subscriber can send an “Unsubscribe” packet to remove it from a data service.

34.6 Mission-Level Control

The *mission-level control* layer (Fig. 34.1) defines algorithms and rules that allow for the UAS to achieve an overarching goal. Elements on this layer are generally the focus of research work and tend to drive system design Love et al. (2009), which can cause significant issues when fielding a system that has not considered the underlying infrastructure. A myriad of control architectures have been developed based on simple communication models (Behnke et al. 2011; Alighanbari and How 2006; Beard et al. 2006). In many cases, these models assume perfect communication or a graph-theoretic disk approximation such that nodes can communicate perfectly when within some maximum range of one another. In any case, realistic communication issues such as data routing and quality of service are mainly ignored. Construction of this layer as the last component of a bottom-up methodology ensures the underlying layers can support the needs of the mission-level control algorithms, and that the algorithms themselves maintain realistic expectations for network performance.

34.6.1 Human Operator Input

Human operators usually provide mission-level control through a graphical user interface. The challenge of a GUI for a complex system is to allow the user the interaction and information they need to perform specific tasks without overwhelming them. This can be achieved by abstracting out the complexities of the system to give a broad overview of the system while allowing the user to access more specific information when necessary.

For the NetUAS system, the GUI is implemented as a computation block in the application layer networking and therefore can be invoked from any machine running the software, including multiple computers on the same network. This allows the flexibility of supporting redundant human operators, or independent human operators in charge of their own subsystem. The GUI can act as a gateway to allow users outside of the internal network to invoke a fully functional instance. It can also be limited to a read-only state where an operator is not allowed to issue commands over the network.

The NetUAS GUI is divided into five spaces as seen in Fig. 34.10 and is designed to be as simple as possible while still relaying pertinent information to the operator. The toolbar across the top enables user interaction with the map space using



Fig. 34.10 Annotated NetUAS GUI screenshot depicting the components (Elston 2011)

quick-reference icons. The upper left pane displays all service discovery, stream subscription, and command status. The lower left pane displays all map layers and allows each one to be toggled by the user. The map space provides the user with spatial situational awareness and can show all of the layers controlled by the lower left-hand pane. The bottom bar gives quick status, such as the time, location of the cursor on the map page, and that cursor location's distance from the current position of the node running the GUI.

Status of the service discovery mechanisms is provided as a list to the user and separated by node. As each node announces itself to the system, its human-readable ID is displayed. Should the node fail to communicate its status for longer than a timeout interval, which is set depending on the mission and underlying network, its background will turn red to indicate this status to the user. By selecting the arrow next to each client, the services/commands that each particular node provides/accepts are displayed to the user (Fig. 34.10). Requests for subscriptions to a particular data stream are performed by clicking on the check mark next to each stream. Once the first packet is received, the background of the stream will turn green to indicate that the subscription went through, and certain streams will show up as information in the rest of the GUI (i.e., the map space).

If the user desires further interaction with any particular node, clicking the check box next to the human-readable name will bring up a separate window (Fig. 34.11). This window is populated with tabs depending on the services provided by the node. Figure 34.11 shows two windows from two separate nodes. One node is an aircraft carrying a pressure, temperature, and humidity (PTH) sonde, and the other

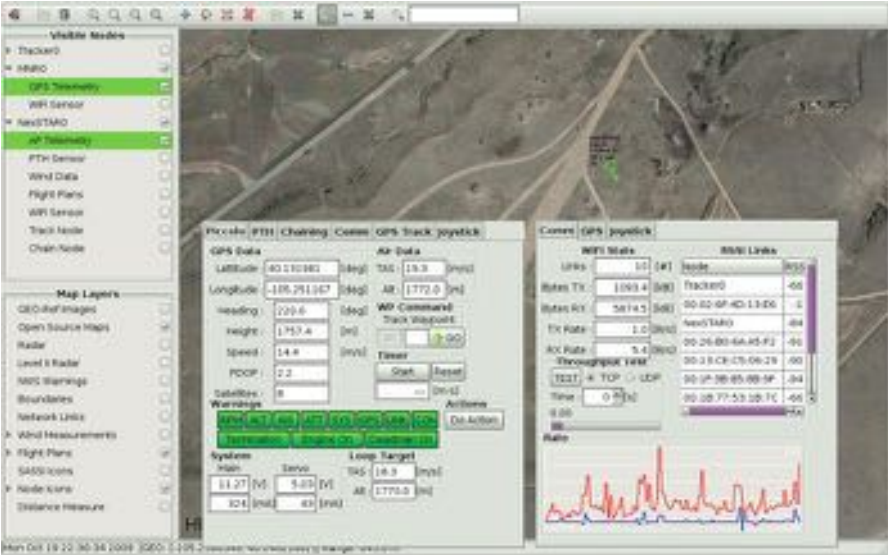


Fig. 34.11 NetUAS GUI with node windows displayed. Each window provides tabs that correspond to capabilities of the particular node and allows the user to see detailed status or issue commands

is a simple meshed network radio. This diversity demonstrates how very different sets of capabilities can be displayed in the GUI. The tabs and their content are all determined dynamically using the service discovery protocol, and their content is defined using XML, making it easy to add a GUI for new services/accepted commands.

Issuing a command to a particular node is simple, requiring the user to open the associated node window and click the appropriate buttons. As an example, the user can command the UA to track an MNR by selecting the appropriate tab of the UA node window and clicking the “Track” button to issue the command. Furthermore, the orbit parameters can be adjusted, such as the altitude or relative distance of the center point from the tracked node. This is particularly nice when attempting to track an aircraft from a moving vehicle. By allowing a user in the car to position the UA to any side of the car, the user can better keep it in view and away from visual obstructions.

Some functionality for commanding nodes is performed outside of the node windows. For example, each node accepting flight plans can be sent a flight plan from the map plane. Clicking on the appropriate menu bar item allows for a plan to be drawn. Upon completion, the user may left click the plan they have drawn (or opened from a file) and using the correct modifier key, translate it, rotate it, or change the position of one waypoint. Once the plan is ready, the user simply has to right click on it, choose “send waypoint,” and then pick the particular client to receive it.

The map plane of the GUI (Fig. 34.10) displays several geo-referenced layers of data including the following: maps, weather information, flight boundaries, network status, wind measurements, flight plans, and node telemetry. The first of these layers, “Open Source Maps,” contains one of several selectable forms of maps: satellite imagery, road network, topographical, and aviation charts. These maps provide background imagery for spatial reference and supply pertinent local information such as road names and elevation. This data is presented in a tiled, scrollable interface similar to what Google Maps offers, allowing the user to easily navigate an area by dragging and zooming with the mouse. Right clicking on the imagery allows the user to switch between the different types of maps.

Each component of the NetUAS GUI was designed to provide for management of a multi-UA system, while presenting the necessary data in such a way to avoid overwhelming the operator. Every component allows for the user to select which data he desires to display, as is evident with the ability to toggle the visibility of map plane layers, the expandable nodes and services lists, and the ability to open windows for more in-depth interaction with a specific node. Intuitive mouse and keyboard control also simplifies the user interaction process, allowing for essential tasks to be completed quickly and easily.

34.6.2 High-Level Control Algorithms

Automated mission-level control is provided through cooperative algorithms. A well-designed network-centric UAS architecture facilitates vehicle coordination which greatly simplifies the creation of high-level control algorithms. Networking components such as service discovery provide for easy identification of nodes with a particular capability, so that partners can be identified. Once these partners are identified, command and data exchange interfaces provide the algorithm with the necessary means for accomplishing coordinated tasks.

The Research and Engineering Center for Unmanned Vehicle (RECUV) Heterogeneous Unmanned Aircraft System (R-HUAS) is an example of a system that has implemented cooperative algorithms using the NetUAS architecture. The layered nature of the NetUAS enabled the R-HUAS to combine both small and miniature UA through a multi-tier heterogeneous airborne network to perform cooperative tasks. The R-HUAS has supported research efforts on a large variety of topics that include cooperative search, acquisition, and tracking (Frew et al. 2008); flocking of miniature air vehicles (Allred et al. 2007); volumetric atmospheric sampling (Argrow et al. 2005); ad hoc networking on small UA (Brown et al. 2007); controlled mobility in ad hoc networks (Brown and Henkel 2006); mother-ship/daughtership control architectures; the construction of communication-based relay chains (Dixon 2010); and the characterization and localization of WiFi sources (Stachura and Frew 2011). Successful demonstration of each of these algorithms have been performed through either flight tests or hardware-in-the-loop simulations.

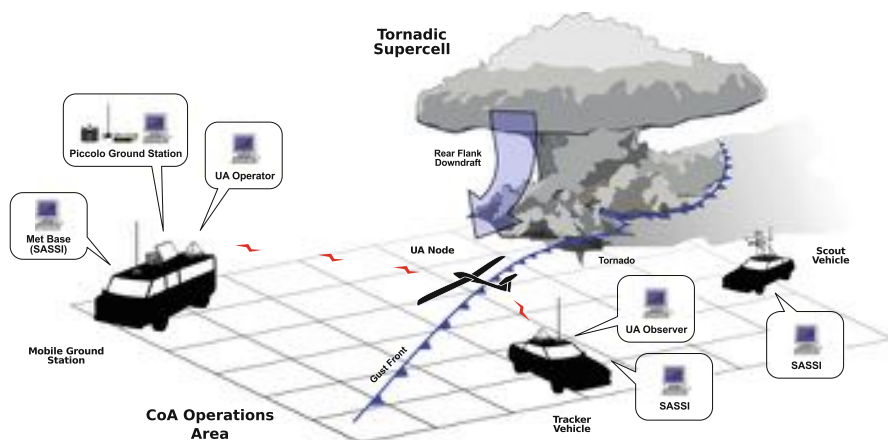


Fig. 34.12 Schematic of the Tempest UAS and ground vehicles for in situ sampling of severe storms (Elston et al. 2011)

34.7 VORTEX2 Implementation

The value of the layered design of the communication, command, and control network described here was demonstrated through successful sampling of severe supercell thunderstorms as part of the VORTEX2 field campaign (Elston et al. 2011). The concept of operations (CONOPS) for sampling supercell storms required the most complex incarnation of the NetUAS architecture to date (Fig. 34.14). This was mainly due to the large number of networked vehicles, and the large number of subsystems and interfaces per vehicle. The unmanned aircraft system (Fig. 34.12) consisted of a total of four vehicles: (1) the Tempest unmanned aircraft, (2) a mobile ground control station used to command the UA, (3) a ground “tracker” vehicle tasked to observe the aircraft, (4) and a “scout” vehicle outfitted with a mobile mesonet (Straka et al. 1996) sensor suite used to provide situational awareness for the ground station.

The Tempest UAS (Tempest 2010) was designed as a low-cost UAS for collecting in situ data in supercell thunderstorms. Two key mission requirements were the ability to launch within 10 min of arrival at the launch site and to maximize sampling time while operating in severe atmospheric conditions. The UA is flown semiautonomously with a Piccolo SL autopilot (Cloudcap 2011), and in situ meteorological data is collected with a Vaisala RS-92 sonde (Hock and Franklin 1999). Networking is provided through a custom version of Linux running on an Overo SBCs from Gumstix (2012). The Overo platform has an onboard 802.11 solution and worked well with an implementation of B.A.T.M.A.N. protocol. The ARM processors required cross compiling the NetUAS software, but the persistent storage is provided through an SD card, making data logging and reconfiguration of the operating system relatively simple.

A mobile ground control station (GCS) is used to support the deployment of the UAS by providing the necessary tools for operation and maintenance of the system. Its crew consists of a driver, meteorologist, UAS manual pilot (who can control the UA through joystick commands issued to the autopilot from a conventional radio-control handset), UAS operator, UAS technician, and pilot in command (PIC). Four different networked computer systems provide interfaces for the meteorologist, UAS operator, and PIC to visualize data from a variety of sources and make mission-level decisions. UAS command, control, and communication are maintained with the UA up to the 20-mile (32 km) operational range through two tracking antennas positioned on the roof of the GCS. These consist of a mechanical system with a high-gain 900-MHz patch antenna and a phased array antenna for directed 802.11 communications. A small Linux single-board computer automatically points the antennas using GPS feeds from the GCS and the UA in conjunction with local magnetometer readings. This helps ensure continuous communication over the range of operations, and although the GCS typically remains stationary during a flight, this enables the GCS to move in the case of an emergency while maintaining communications with the UA.

A wide area network (WAN) interface is available in the mobile GCS through the use of two types of cellular connections. This provides access to the Internet, which allows the head meteorologist to make navigation and targeting decisions based on real-time radar data using the Gibson Ridge Level 3 (GR3) (GRLevel3 2010) software package. It also allows for visualization of the positions and real-time data of other assets in the VORTEX2 armada using the SASSI tool (Rasmussen 2010) for severe weather situational awareness. The Internet is also used by the UAS operator for dynamic map requests and for the real-time publication of the UA location and meteorological measurements.

During flight operations, the UA is commanded to orbit within 1,000 ft vertically and 1/2 mile horizontally of the tracker ground vehicle that carries the UA observer. This is done to satisfy Federal Aviation Administration (FAA) see-and-avoid requirements (Elston et al. 2011). The tracker vehicle contains two systems that allow the UA to follow with a high level of autonomy and that free the observers to focus on airspace monitoring and UA observation. The first system is a modified version of the MNR, based on the Alix 3d2 single-board computer. The MNR contains a GPS receiver and provides the location of the tracker vehicle to subscribers. By allowing the UA flight computer to subscribe to this GPS location, a controller on the UA can track and orbit the tracker. This provides two advantages: the UA remains within the required distance of the tracker, and the sampling of the storm is simplified to directing the driver of the tracker using voice commands over a VHF radio, which indirectly moves the UA. The second system is a laptop computer running the networking software and a limited-functionality graphical user interface. This interface provides status of the system and position of the UA at all times, allowing the personnel in the tracker to provide an offset to the UA orbit. This tracker-relative orbit is chosen to allow easy, full-time observation by one of the designated UA observers through a side window or sunroof.

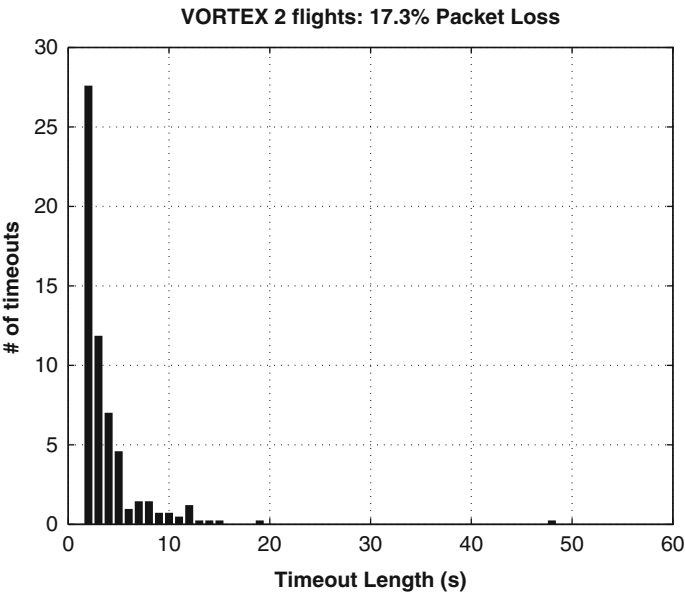


Fig. 34.13 Aggregate histogram of communication timeouts using finalized system over 4 VORTEX2 deployments from June 6, 7, 9, and 10, 2010 (Elston et al. 2011)

34.7.1 VORTEX2 Experiment and Results

System development and flight testing for the VORTEX2 project was conducted over 2 years and 69 flights of the Tempest UAS. During each of the flights, in addition to recording pressure, temperature, humidity, and wind information, logs were kept of all traffic through the NetUAS system, routing and network statistics (Fig. 34.13), the locations of the vehicles, telemetry from the autopilot, and data from the electronic speed controller (ESC). The field component of the VORTEX2 project was composed of the final 21 of the 69 Tempest UA flights and used two different aircraft from 2 May to 10 June, 2010.

All flights during the VORTEX2 campaign occurred without incident despite being flown near and beneath supercells and other severe convective storms, and through light precipitation. A total of six flights in the proximity of supercell thunderstorms were performed, including one flight (10 June, 2012) conducted shortly after the supercell produced two tornadoes. Basic information for each flight is provided in Table 34.6, which includes date, aircraft flown, time of flight, COA area for flight, and whether or not the UAS team was performing the mission in coordination with the rest of the VORTEX2 armada. This coordination is significant as it provides the opportunity for UAS data to be combined with measurements of the storm taken from the other sensor platforms in the armada. The UAS team was not always able to coordinate with the armada given the limited area for UAS operations compared to the entire range of the VORTEX2 project (Elston et al. 2011).

Table 34.6 Summary of significant VORTEX2 deployments

| Date | With armada | Flight (mm:ss) | COA area |
|--------------|-------------|----------------|-------------|
| 6 May 2010 | Yes | 44:15 | 2009-CSA-37 |
| 26 May 2010 | Yes | 45:01 | 2009-WSA-23 |
| 6 June 2010 | No | 19:35 | 2009-WSA-13 |
| 7 June 2010 | No | 27:30 | 2009-CSA-9 |
| 9 June 2010 | Yes | 33:30 | 2009-CSA-6 |
| 10 June 2010 | Yes | 34:00 | 2009-WSA-33 |

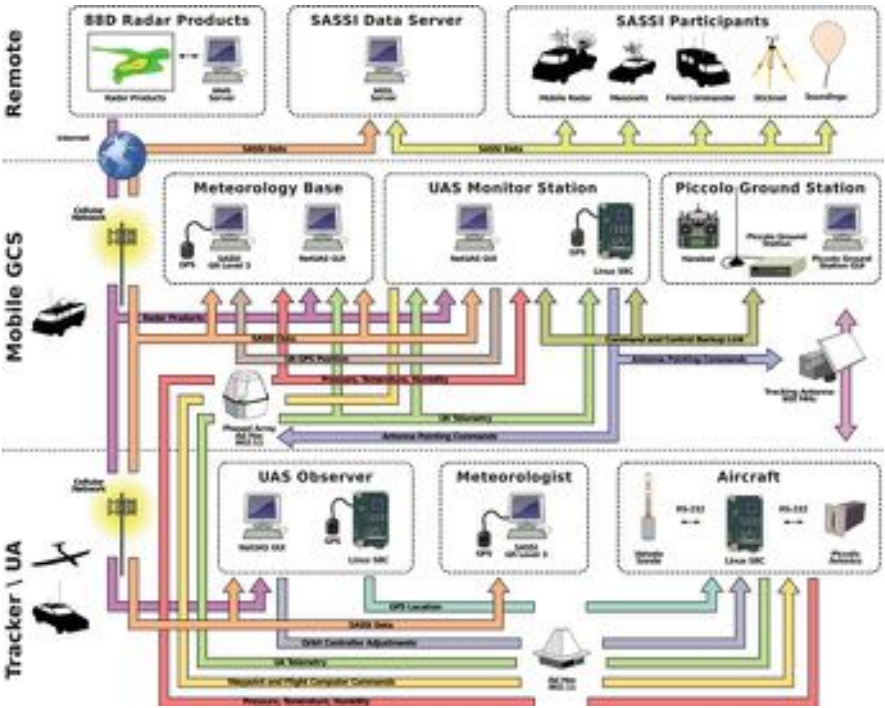


Fig. 34.14 System diagram for entire UAS including remote data sources (Elston et al. 2011)

Figure 34.14 represents the complete, ideal system configuration for the VORTEX2 experiment, with each arrow representing a directed data stream that is provided and managed through the NetUAS service discovery functionality. In practice, portions of the data streams would go down, e.g., from loss of connectivity due to separation, and the architecture provided seamless transitions between configurations. The ad hoc networking capabilities of the NetUAS accommodated a large number of networked subsystems and vehicles, while also allowing for switching out aircraft, vehicles, computers, and nodes in the field without requiring reprogramming of the system.

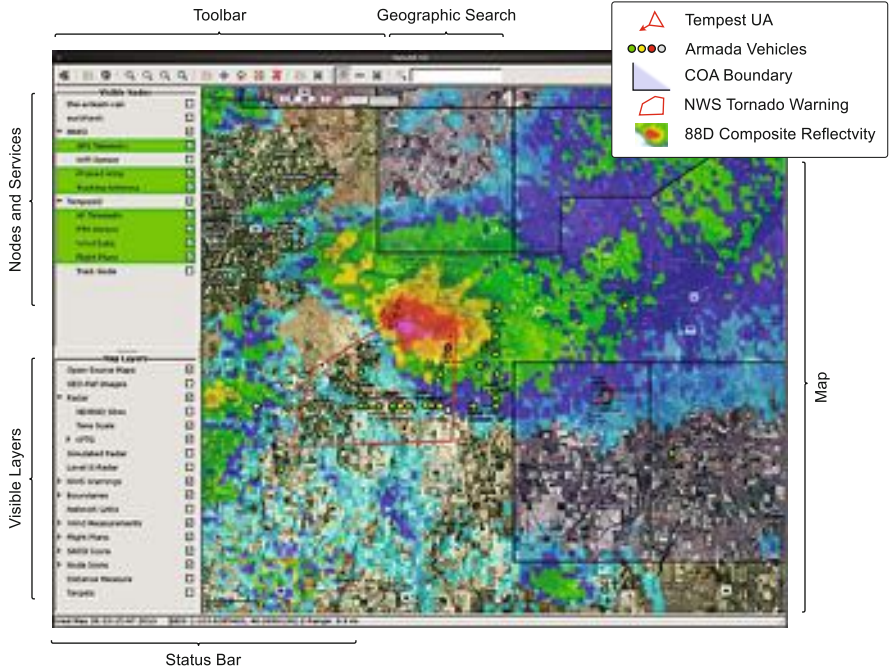


Fig. 34.15 Annotated NetUAS GUI screenshot taken during 26 May 2010 operations. The *left panel* contains a list of connected nodes and their respective services, along with a list of displayable layers. The *right panel* shows the overlay of WSR-88D image of the supercell, COA boundaries, and locations of other deployed VORTEX2 vehicles (Elston et al. 2011)

The modular nature of the NetUAS architecture easily accommodated new enhancements to the system. As an example, interfaces to external data sources were added at several points in the system, and functionality was added to the GUI to display this data (Fig. 34.15 shows a screenshot of the GUI taken during 26 May 2010 operations). Several new GUI layers were added including real-time WSR-88D weather radar data, National Weather Service warnings, flight region boundaries, VORTEX2 vehicle locations (available through SASSI (Rasmussen 2010)), and real-time UA wind measurements. In addition to being able to display new data sources, the GUI was also enhanced to satisfy the needs of the atmospheric sampling mission. Support was added for centering the map using city or road name searches, quick flight plan generation, and the ability to send waypoint plans to all vehicles (including ground vehicles).

The NetUAS also accommodated software changes to increase system functionality while in the field. These changes were generally implemented between flight experiments, a capability that was only achievable due to the ability to quickly change modules without affecting the rest of the system. During the deployment, a module was added to the operator interface to include battery usage estimates for determining a safe time to terminate operations with enough power to return to the

GCS and land the aircraft. A module was also added to the operator station in the GCS that subscribed to UA telemetry and converted it to an NMEA string. This string was transmitted over a USB interface. In this manner, a computer connected to the other end of the USB saw it as a GPS device. This allowed for easy integration of UA telemetry into the GR3 software, which already had support for reading GPS devices.

The layered design of the network also allowed for easy system reconfigurations to address issues encountered during the experiments. The design accommodated one deployment which required the UA operator station in the GCS to be field swapped when the graphics card failed with a laptop used as a backup in the tracker vehicle. It also allowed for the tracker's entire system to be reconfigured several times following the discovery that the ad hoc link from the UA to the tracker was problematic in both packet loss and, more importantly, occasional lengthy communication timeouts. During these timeouts, the GCS could still command the UA; however, if the UA was not tethered to the tracker, it was much more difficult for the observer in the tracker to maintain a visual fix on the UA (Elston et al. 2009a; Houston et al. 2011).

Due to the time constraints of the project, a more thorough investigation of this problematic link was not possible, so a solution had to be found using the available hardware. Changes were made on multiple levels of the design, but due to the modular and abstracted nature of the NetUAS architecture, no software changes were required, and little reconfiguration was necessary for the rest of the system. One of the configurations consisted of removing the MNR and connecting the observer's laptop to a USB GPS unit to provide the location to the UA. By using the publish/subscribe architecture and service discovery, this was seamlessly accomplished, requiring only that the operator at the mobile GCS identify the laptop as the target to be tracked instead of the MNR. Another configuration was attempted using a wired interface through an Ethernet cable from the observer laptop to limit WiFi interference with the UA-to-tracker link. In this configuration, NetUAS provided the functionality necessary to establish a gateway between the wired and wireless networks. Because of this ease of reconfiguration, in relatively little time several configurations were tested, and one was eventually found that proved a solution that worked well enough to successfully complete the scientific experiments (Elston et al. 2011).

Conclusion

The bottom-up design approach described here enables the creation of a net-centric command and control architecture for a complex unmanned aircraft system. The design philosophy was centered on using modular and off-the-shelf technology as much as possible. This philosophy is evident across the layers. At the lowest hardware layer, several types of COTS networking components and radios were used to provide communication between aircraft, ground vehicles, and static nodes. The establishment of links between the radios was performed using commercial routing protocols wherever possible, with the

exception of a few custom solutions to accommodate unique applications. Intra-vehicle communication was enabled between a commercially available autopilot, communication solution, and sensing payload using modular interface nodes with a reliable bus protocol. Finally, these were tied together through a unique application layer networking solution which provided mechanisms for service discovery, a publish/subscribe data architecture and the issuing of commands.

Given this layered architecture, the development and implementation of cooperative algorithms was greatly simplified. Algorithms were constructed without worrying about details of data and information flow. In some cases, like tracking a ground vehicle, the aircraft communicated directly with the target, while in others, such as when working with heterogeneous vehicles and networks, the data was routed through several nodes before reaching its destination. Service discovery mechanisms allowed for cooperative algorithms to easily identify nodes capable for performing the tasks necessary, request their participation, and exchange the data necessary for each vehicle to complete its task.

The benefits of this architecture and layered design approach were demonstrated through the successful deployment of a complex UAS for use in the VORTEX2 project. The Tempest unmanned aircraft system accomplished several milestone goals including the first ever use of a UAS to collect data in close proximity to a supercell on 6 May 2010, and the first ever sampling of a supercell rear-flank downdraft airmass by a UAS on 9 June 2010. The network-centric nature of the operation enabled data to be gathered about the storm environment and UAS (such as observed winds, control surface deflections, and control-loop tracking) that will be used as the basis for the design of future networked unmanned aircraft systems.

References

- M. Abolhasan, B. Hagelstein, J.C.-P. Wang, Real-world performance of current proactive multi-hop mesh protocols, in *Proceedings of the IEEE Asia Pacific Conference on Communications*, Shanghai (IEEE, Piscataway, 2009)
- M. Alighanbari, J.P. How, Robust decentralized task assignment for cooperative uavs, in *Collection of Technical Papers – AIAA Guidance, Navigation, and Control Conference*, Keystone, vol. 5 (AIAA, Reston, 2006), pp. 3232–3247
- J. Allred, A.B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, K. Mohseni, Sensorflock: an airborne wireless sensor network of micro-air vehicles, in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, Sydney (ACM, New York, 2007), pp. 117–129
- AODV-UU, Ad-hoc on-demand distance vector routing (2010), <http://core.it.uu.se/core/index.php/AODV-UU>
- B. Argrow, D. Lawrence, E. Rasmussen, Uav systems for sensor dispersal, telemetry, and visualization in hazardous environments, in *43rd Aerospace Sciences Meeting and Exhibit* (AIAA, Reston, 2005)
- B.A.T.M.A.N., b.a.t.m.a.n (2010), <https://www.open-mesh.org>
- R.W. Beard, T.W. McLain, D.B. Nelson, D. Kingston, D. Johanson, Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *IEEE Proc. Spec. Issue Multi-Robot Syst.* **94**(7), 1306–1324 (2006)

- D. Behnke, K. Daniel, C. Wietfeld, Comparison of distributed ad-hoc network planning algorithms for autonomous flying robots, in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, 2011, pp. 1–6 doi:10.1109/GLOCOM.2011.6134382
- T.X. Brown, D. Henkel, On controlled node mobility in delay-tolerant networks of unmanned aerial vehicles, in *Proceedings of International Symposium on Advanced Radio Technologies* (Institute for Telecommunication Sciences/National Telecommunications and Information Administration, Boulder, 2006)
- T.X. Brown, B.M. Argrow, E.W. Frew, C. Dixon, D. Henkel, J. Elston, H. Gates, Experiments using small unmanned aircraft to augment a mobile ad hoc network, in *Emerging Technologies in Wireless LANs: Theory, Design, and Deployment*, ed. by B. Bing (Cambridge University Press, Cambridge, 2007), pp. 123–145. ISBN:ISBN-13: 9780521895842. Chapter 28
- I.D. Chakeres, E.M. Belding-Royer, Aodv routing protocol implementation design, in *ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops – W7: EC (ICDCSW'04)*, Washington, DC (IEEE Computer Society, Los Alamitos, 2004), pp. 698–703
- H.C. Christmann, E.N. Johnson, Design and implementation of a self-configuring ad-hoc network for unmanned aerial systems, in *AIAA Infotech@Aerospace*, Rohnert Park (AIAA, Reston, 2007)
- Cloudcap, The cloudcap website (2011), <http://cloudcaptech.com>
- K.D. Davis, Interim operation approval guidance 08–01: unmanned aircraft systems operations in the U.S. national airspace system. *FAA Unmanned Aircraft Systems Program Office* (2008)
- C. Dixon, Controlled mobility of unmanned aircraft chains to optimize network capacity in realistic communication environments. Ph.D. thesis, University of Colorado, 2010
- J. Elston, Semi-autonomous small unmanned aircraft systems for sampling tornadic supercell thunderstorms. Ph.D. thesis, University of Colorado, 2011
- J. Elston, B. Argrow, E.W. Frew, A distributed avionics package for small uavs, in *AIAA Infotech@Aerospace Conference*, Arlington (AIAA, Reston, 2005)
- J. Elston, B. Argrow, A. Houston, J. Lahowetz, Distributed atmospheric sensing using small uas and doppler radar, in *AIAA Infotech@Aerospace Conference*, Seattle (AIAA, Reston, 2009a)
- J. Elston, E.W. Frew, D. Lawrence, P. Gray, B. Argrow, Net-centric communication and control for a heterogeneous unmanned aircraft system. *J. Intell. Robot. Syst.* **56**(1–2), 199–232 (2009b)
- J.S. Elston, J. Roadman, M. Stachura, B. Argrow, A. Houston, E.W. Frew, The tempest unmanned aircraft system for in situ observations of tornadic supercells: design and vortex2 flight results. *J. Field Robot.* **28**, 461–483 (2011)
- EnGenius Tech Inc, The engenius website (2011), <http://engeniustech.com>
- E.W. Frew, D.A. Lawrence, S. Morris, Coordinated standoff tracking of moving targets using lyapunov guidance vector fields. *AIAA J. Guid. Control Dyn.* **31**(2), 290–306 (2008)
- E.W. Frew, J. Elston, B. Argrow, A. Houston, E. Rasmussen, Sampling severe local storms and related phenomena: using unmanned aircraft systems. *Robot. Autom. Mag.* **19**(1), 85–95 (2012)
- GRLevel3, grlevel3 (2010), <http://www.grlevelx.com/grlevel3/>
- Gumstix, The gumstix website (2012), <http://www.gumstix.com>
- J. Halverson, P.L. Azoifeifa, M. Black, S. Braun, D. Cecil, M. Goodman, A. Heymsfield, G. Heymsfield, R. Hood, T. Krishnamurti, G. McFarquhar, J. Molinari, R. Rogers, J. Turk, C. Velden, D.-L. Zhang, E. Zipser, R. Kakar, Nasa's tropical cloud systems and processes (tcsp) experiment: investigating tropical cyclogenesis and hurricane intensity change. *NASA TCSP News* (2005). <http://tcsp.nsstc.nasa.gov/news.html>
- D. Henkel, C. Dixon, J. Elston, T.X. Brown, A reliable sensor data collection network using unmanned aircraft, in *Proceedings of the Second International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN)* (ACM, New York, 2006)
- S. Hipskind, G. Tyrell, G. Holland, J. Curry, Use of the aerosonde uninhabited aerial vehicle (UAV) in the fourth convection and moisture experiment (CAMEX 4), in *AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles*, Portsmouth (AIAA, Reston, 2002)

- T.F. Hock, J.L. Franklin, The near gps dropwindsonde. *Bull. Am. Meteorol. Soc.* **80**(3), 407–420 (1999). ISSN 0003–0007
- A. Houston, B. Argrow, J. Elston, J. Lahowetz, P. Kennedy, The collaborative colorado-nebraska unmanned aircraft system experiment. *Bull. Am. Meteorol. Soc.* **93**, 39–54 (2011) (In Submission)
- J. How, E. King, Y. Kuwata, Flight demonstrations of cooperative control for uav teams, in *Proceedings of the AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, Chicago (AIAA, Reston, 2004)
- D.B. Johnson, D.A. Maltz, J. Broch, Dsr: the dynamic source routing protocol for multi-hop wireless ad hoc networks, in *In Ad Hoc Networking*, ed. by C.E. Perkins (Addison-Wesley, Boston, 2001) pp. 139–172. Chapter 5
- E. Kohler, R. Morris, B. Chen, J. Jannotti, M. Frans Kaashoek, The click modular router. *ACM Trans. Comput. Syst.* **18**, 263–297 (2000). ISSN 0734–2071
- N.E. Leonard, D.A. Paley, R.E. Davis, D.M. Fratantoni, F. Lekien, F. Zhang, Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *J. Field Robot.* **27**(6), 718–740 (2010)
- P.-H. Lin, C.-S. Lee, The eyewall-penetration reconnaissance observation of typhoon longwang (2005) with unmanned aerial vehicle, aerosonde. *J. Atmos. Ocean. Technol.* **25**(1), 15–25 (2008)
- J. Love, J. Jariyasunant, E. Pereira, M. Zennaro, K. Hedrick, C. Kirsch, R. Sengupta, Csl: a language to specify and re-specify mobile sensor network behaviors. (IEEE RTSS), 1 (2009)
- Office of the Secretary of Defense, *FY2009–2034 Unmanned Systems Integrated Roadmap* (Department of Defense, Washington, DC, 2009)
- Performance Analysis of Telecommunication Systems (PATS) research group, An implementation of aodv in click (2012), <http://www.pats.ua.ac.be/software/aodv>
- E. Rasmussen, Situational awareness for severe storms intercept (2010), <http://www.rasmsys.com/Software/page8/page8.html>
- M. Stachura, E.W. Frew, Cooperative target localization with a communication aware unmanned aircraft system. *AIAA J. Guid. Control Dyn.* **34**(5), 1352–1362 (2011)
- J.M. Straka, E.N. Rasmussen, S.E. Fredrickson, A mobile mesonet for finescale meteorological observations. *J. Atmos. Ocean. Technol.* **13**(5), 921–936 (1996). ISSN 0739–0572. doi:10.1175/1520-0426(1996)013<0921:AMMFFM>2.0.CO;2
- Tempest, The recuv tempest ua website (2010), <http://recuv.colorado.edu/tempest>
- J. Tisdale, Z. Kim, J. Hedrick, Autonomous uav path planning and estimation. *IEEE Robot. Autom. Mag.* **16**(2), 35–42 (2009). ISSN 1070–9932. doi:10.1109/MRA.2009.932529
- J. Zheng, M.J. Lee, A comprehensive performance study of IEEE 802.15.4, in *Sensor Network Operations* (IEEE, Piscataway, 2006), pp. 218–237

Timothy X. Brown, Mark McHenry, and Suppapol Jaroonvanichkul

Contents

| | | |
|--------|--|-----|
| 35.1 | Introduction | 814 |
| 35.2 | Basic UAS Communication Architectures | 818 |
| 35.2.1 | UAS Communication | 818 |
| 35.2.2 | Cognitive Radio Architecture | 820 |
| 35.3 | UAS Spectrum Stakeholders and Requirements | 821 |
| 35.3.1 | Incumbent Requirements | 821 |
| 35.3.2 | Entrant Requirements | 822 |
| 35.3.3 | Spectrum Regulator Requirements | 823 |
| 35.4 | Mechanisms for UAS Spectrum Awareness | 823 |
| 35.4.1 | Geolocation and Database Approaches | 824 |
| 35.4.2 | Sensing Approaches | 825 |
| 35.4.3 | Beacon Approaches | 829 |
| 35.4.4 | Hybrid Approach | 833 |
| 35.5 | Mechanisms to Manage UAS Spectrum Access | 833 |
| 35.5.1 | Types of Spectrum Access | 834 |
| 35.5.2 | Control of Spectrum Access | 835 |
| 35.6 | Example | 839 |
| 35.7 | Conclusion | 841 |
| | References | 843 |

T.X. Brown (✉) • S. Jaroonvanichkul
University of Colorado, Boulder, CO, USA
e-mail: timxb@colorado.edu; Suppapol.Jaroonvanichkul@Colorado.EDU

M. McHenry
Shared Spectrum Company, Vienna, VA, USA
e-mail: mmchenry@sharespectrum.com

Abstract

This chapter examines the enabling role of cognitive radio technologies for unmanned aircraft systems to access more spectrum needed for flight operations. Analysis of stakeholders sets the requirements for cognitive radios. With these requirements, the paper describes the different architecture choices available to the UAS cognitive radio designer. These include the communication architecture, the spectrum awareness techniques for assessing what spectrum is available, and the spectrum access techniques for deciding which available spectrum to use. Cognitive radios provide a number of benefits over traditional radios: they increase spectrum management flexibility and automatically resolve conflicting allocations. This promotes more widespread spectrum sharing, increasing the spectrum available. It provides mechanisms to automatically make fine-grained allocations, such as spectrum reserved for specific phases of flight, which increases unmanned aircraft flight safety. It enables new business models of spectrum management that can encourage better allocation of spectrum resources. Information from this chapter is relevant for the development of future unmanned aircraft systems rules and standards.

35.1 Introduction

Unmanned aircraft systems (UASs) have the potential to play an important social, economic, and security role for countries around the globe. Unlike manned aircraft where aircraft flight control systems are an integral part of the aircraft, pilot controls of UAS are physically separated from the aircraft. Further, communication to air traffic control (ATC) and sensors for detect sense and avoid (DSA) will be required to safely integrate UAS into national airspace systems (FAA 2001). This poses a great technical challenge to remotely operate the UAS. Thus, one of the primary concerns regarding the ability of UAS to meet safety regulations without conflicting with existing systems is the availability and allocation of bandwidth and spectrum for control, command, and communication (C3) (FAA 2006; Neale 2007). In addition to C3 for remote piloting and platform safety, UAS communication with payloads, including backhaul of imagery and sensor data, and payload management is required and may in some cases demand more bandwidth than the C3. These communication needs are shown in Fig. 35.1. As noted in the figure, the communication is both with the aircraft and among parties.

It is useful to distinguish internal and external communication (Heppel 2008). Internal communication is within the UAS and includes the C3 connection between the unmanned aircraft and the pilot as well as the payload management. External ATC and DSA communication will use standardized frequencies (e.g., standard ATC voice, ADS-B, weather radar, transponder, and GPS channels). As such they are outside of the scope of this study. It should be noted that these external communications will require additional internal communication back to the pilot

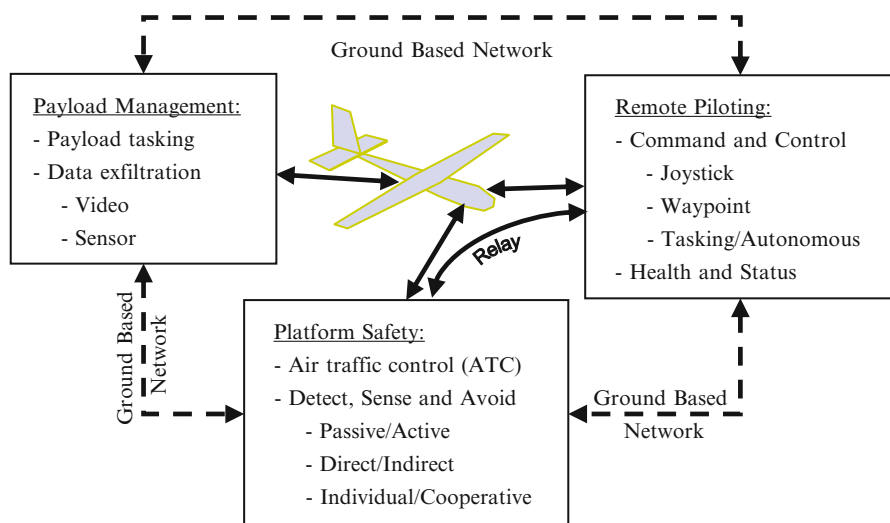


Fig. 35.1 Types of communication in UAS

for pilot situational awareness and proper integration into airspace management systems. So, the focus of this chapter is to support internal communications.

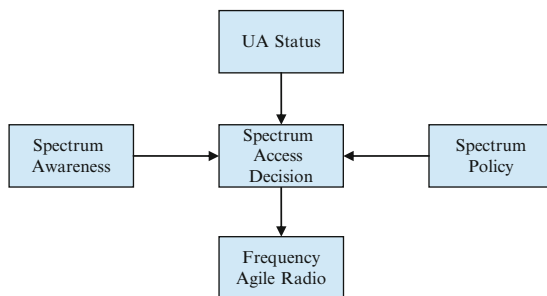
Existing UASs, use technologies ranging from military aviation links, to satellite links, to short-range links adapted from other standards. Regardless of the link technology, addressing this communication need requires radio spectrum. Today UASs face two spectrum problems:

- Spectrum scarcity
- Static spectrum management

The RTCA has estimated that the spectrum requirement for UAS C3 is at least 13MHz and as much as 300MHz (Box et al. 2009). However, there is currently limited spectrum allocated for civilian and military UAS C3, and demand for spectrum from all applications is growing. All useful bands for radio communications have been allocated. Allocation of spectrum on a global basis is controlled by international agreements, which are handled by the World Radio Conference (Griswold 2008). In Agenda 1.5 of the World Radio Conference (WRC) 2015, there will be discussion of use of spectrum allocated to satellites for UAS (ITU 2012). However, in general, spectrum allocation agreements are difficult and slow to be changed. Even if there are additional spectrum allocated for UAS use, there could be years of delay waiting for the existing spectrum users to vacate the bands. Further, any allocation is likely to fall short of long-term UAS spectrum needs.

While spectrum is fully allocated, paradoxically, studies show that spectrum is underutilized for most of the time. According to the Federal Communications Commission in the U.S. (FCC 2003), there is a large variation in the use of spectrum bands in space and time. The utilization varies from 15 to 85 % indicating large

Fig. 35.2 Functional blocks of a cognitive radio



portions of usable spectrum are underutilized or “vacant.” Such unused spectrum is referred to as spectrum *white spaces*. Although artificial, a spectrum scarcity exists and is a side effect of the inefficiencies of fixed assignment. This results from the currently static approach to spectrum management, where a large amount of spectrum is assigned for long periods of time over large regions (Akyildiz et al. 2006; Mitola 2006). Additionally, with “manual” spectrum management, deployment of systems is troublesome. If there are conflicts in frequencies assigned, the time needed to resolve such conflicts can be substantial. For example, Griswold (2008) has illustrated that “In Iraqi Freedom, the Army’s Hunter [UAS] did not operate for the first 30 days in-theater because it was awaiting frequency deconfliction in order to obtain operational frequencies to use.” Worse, such conflicts can go undetected until interference is observed in flight operations.

A significant communication role can be filled by new radio technologies. One such technology is *cognitive radio* (Akyildiz et al. 2006). A cognitive radio determines unused portions of the existing wireless spectrum and adapts the radio’s operating characteristics to operate in these unused portions in a manner that limits interference with other devices. The functional components of a cognitive radio are shown in Fig. 35.2. The cognitive radio is built on top of a frequency agile radio that can change the frequency bands or modulation with which it communicates under software control. The choice of bands is a so-called *spectrum access* decision that is based on several factors. Three factors are considered in this chapter. *Spectrum awareness* is how the cognitive radio discovers what frequency bands are not being used by an incumbent user and thus potentially available. *UA status* is information such as the location of the unmanned aircraft and its current phase of flight. *Spectrum policies* are machine-readable rules that specify in which bands a radio is permitted to (or prohibited from) transmitting and under what conditions. Thus, to transmit in a band requires that the band is within the capabilities of the radio, not occupied by an incumbent user, and there is a policy permitting its use and appropriate for the current location and status of the UA.

Cognitive radios address the problems of spectrum scarcity in three ways. First, through the use of spectrum awareness, a cognitive radio can find unused spectrum to utilize. Second, since the radios are frequency agile, they can use bands that would

otherwise be neglected. The radio can use bands based on location, phase of flight, and other conditions. For instance, several planes can share low-quality unlicensed bands during preflight where packet losses and delays are tolerable leaving more spectrum for higher-quality dedicated bands during more critical phases of flight such as takeoff. Third, policies enable a mechanism to acquire spectrum through spectrum leasing. Spectrum leasing (FCC 2004) is an agreement where a spectrum holder leases its spectrum to another spectrum user. Machine-readable policies can be used to define lease terms in a spectrum-leasing agreement. Machine-readable policies allow lease terms to be defined precisely and be followed strictly. So, the policy-based approach could encourage spectrum holders to lease their spectrum. Generally, the policies are used to define time-limited leases that limit the period for which the spectrum is valid to a specific period (Chapin and Lehr 2007). This time limit limits the risk even if the terms are found later to be unfavorable to either the lessor or lessee.

More generally, policies address the problem of static spectrum management. Cognitive radios enable fine-grained time, frequency, and location allocations of radio spectrum for communication. Current approaches allocate large swaths of spectrum over large regions for decades at a time. Finer-grained allocations foster more efficient use of the spectrum with greater communication capacity to support future UAS growth and innovation. More importantly, cognitive radio can make such allocations automatically while still adhering to stakeholder requirements. This automation provides government regulatory flexibility that is not currently present and promotes better use of the spectrum. Spectrum regulators, such as the FCC in the U.S., recognize that cognitive radios can be applied to dynamically reuse white spaces in licensed spectrum bands, thereby efficiently utilizing underutilized spectrum (FCC 2002). More importantly, as will be shown, cognitive radios enhance UAS flight safety. They provide more spectrum for better aircraft control. They are naturally frequency agile so that they can avoid interference. They allow spectrum to be preplanned so that UAS pilots can have some assurances that they will have the requisite spectrum before the flight starts. And, they can automatically navigate and manage complex spectrum assignments and avoid hazardous conflicts on UAS C3 channels. However, there remain many open questions about how such a cognitive radio system should be organized and operated. Therefore, this chapter seeks to answer the following questions:

- What are basic architectures for UAS communication?
- Who are the UAS spectrum stakeholders?
- What are the mechanisms for UAS spectrum awareness?
- What are the mechanisms for UAS spectrum access?

Certain questions such as cognitive radio security are not addressed here (Brown and Sethi 2008). The remainder of this chapter addresses the above four questions and then provides an example scenario to suggest how the different elements fit together. The goal is not to find the right answer to each of these questions. But, rather, to lay out the possible range of solutions that are available to guide system designers as they try to adopt cognitive radio technology for specific UAS.

35.2 Basic UAS Communication Architectures

UASs consist of a wide range of aircraft sizes, ranges, altitudes, and endurances (van Blyenburgh 2009). Unmanned aircraft sizes can be as small as a handheld aircraft or as large as a 12,000-kg Global Hawk. Ranges can be less than a kilometer to thousands of kilometers. Altitudes range up to 20,000 m. Endurances go from a few minutes to several days.

UASs operate through different phases of flight (RTCA 2011):

1. Flight Planning: flight tasking and uploading of the flight plan
2. Start and Taxi: performing checklists, rolling down taxiways, rolling down runway
3. Takeoff and Departure: leaving ground to being assigned to en route ATC authority
4. En route: traveling directly from one place to another after takeoff and departure
5. Aerial Work: loitering, searching patterns, etc. as part of operations
6. Decent and Approach: decreasing altitude to final approach point
7. Terminal and Landing: arriving at airfield, landing, and exiting the runway
8. Postlanding: taxiing and parking
9. Emergency: UAS failure that prevents normal operation

The different phases of flight are related to the bandwidth, delay, and reliability requirements (Box et al. 2009). To accommodate these requirements, various communication architecture and spectrum bands can be used to connect the unmanned aircraft and the unmanned aircraft control stations (UACS). The next section describes UAS communication architectures and then the role of cognitive radios.

35.2.1 UAS Communication

These wide-ranging specifications foster a variety of architectures to meet UAS communication needs between the UA and control station. UASs currently use several models of communication including satellite and direct ground-based line of sight. Short-range UASs as well as other UASs during takeoff and landing use direct line-of-sight communication links. These can have communication ranges beyond a 100 km for larger aircraft and more capable radio systems or less than a kilometer for simpler systems. Longer-range UASs use satellite communication links en route when beyond line of sight of the control station. Generally, these are on larger aircraft but simple satellite-phone-based links have been used on smaller aircraft.

The RTCA SC-203 has been examining the architectures and spectrum requirements for UAS operations (Heppe 2008). In addition to current approaches, outlined above, they also recognize a third approach, in which there is a network of ground-based radios similar to a cellular telephone system. Cellular telephone systems are noted for their very efficient use of the spectrum, ability to implement strong security, and the ability (when well planned) to provide seamless coverage.

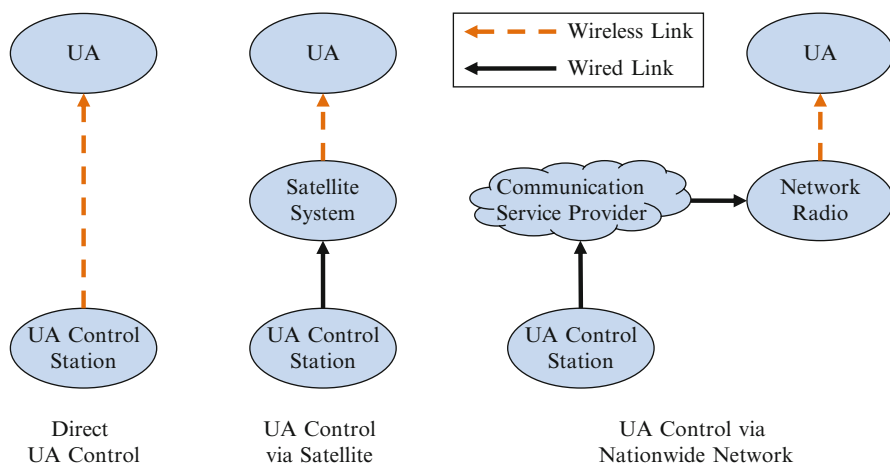


Fig. 35.3 UAS C3 architectures

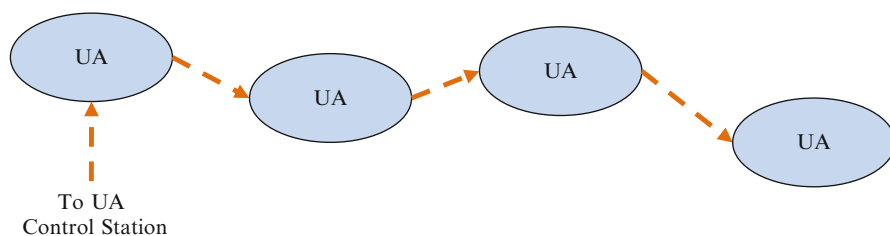


Fig. 35.4 UAS mesh

The use of a cellular-like ground-based system to provide nationwide high-bandwidth communication service has been demonstrated by a commercial vendor. AirCell of Louisville, CO (www.aircell.com), provides a nationwide multimegabit per second air to ground communication service in the U.S. Although based on commercial cellular technology and not approved for flight-safety communications, the AirCell system does demonstrate the feasibility of this technology and also is collecting valuable operational experience. The three types of command and control architectures are illustrated in Fig. 35.3. A given UAS may use one, two, or all three of these architectures depending on location and phase of flight.

In addition to these architectures, a fourth architecture builds on the previous architectures in a so-called mesh as shown in Fig. 35.4. In this architecture, one or more UA have a connection to the control station using one of the above architectures. Other UA have C3 access via relay communications back to the control station. This architecture is more useful for controlling multiple UA from the same control station. Here the focus is on controlling a single UA.

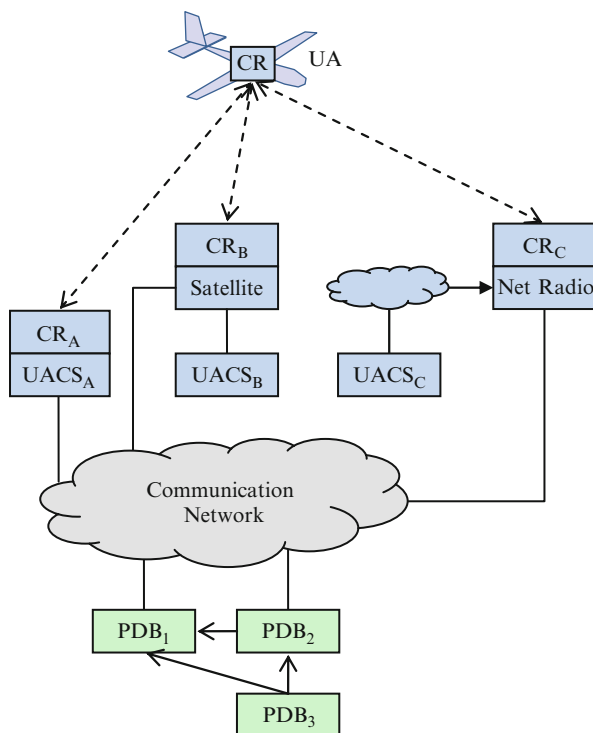


Fig. 35.5 Cognitive radio (CR) architecture for UAS

35.2.2 Cognitive Radio Architecture

The cognitive radio architecture should support the different UAS communication architectures. For each of these architectures, a cognitive radio is added on the final link as shown in Fig. 35.5. In the direct architecture, a cognitive radio is connected to the UA control station (either directly or through some intermediate network). In the satellite architecture, the cognitive radio is on the satellite. The cognitive radio would likely be distributed with the transmitter and receiver on the satellite and the other cognitive radio components located on the ground portion of the satellite system. In the nationwide network, the cognitive radio is located at the network radios.

In each of these architectures, the cognitive radios need to have access to policies. Conceptually these can be thought of as residing in policy databases (PDB) that are connected to the cognitive radios in the UAS.

Cognitive radio technology can address the UAS spectrum problem. However, what are the unique challenges for UAS and the components that make up a UAS cognitive radio architecture? The next section starts with the UAS spectrum stakeholders.

35.3 UAS Spectrum Stakeholders and Requirements

Cognitive radios can allow a UAS to share spectrum with an existing spectrum user. In this model the existing spectrum user is denoted the *incumbent*, and the UAS that is attempting to use the spectrum is denoted the *entrant*. This section describes the incumbent’s and the UAS entrant’s requirements that a spectrum-sharing approach must meet as well as the requirements of the spectrum regulator. The incumbent could be another UAS or a different type of wireless service. It is critical to understand and to address these requirements in any spectrum-sharing scheme.

35.3.1 Incumbent Requirements

Some of the requirements that incumbents would expect to be implemented through technology, spectrum-sharing criteria (i.e., regulations or standards), or negotiated sharing arrangements are shown in Table 35.1. The most obvious mandatory requirement involves assured coexistence and compatibility, addressing the core concern that any new entrant must not cause “harmful interference” to the incumbent systems. Another critical, but potentially optional, requirement is that incumbent must not only have the rights to change and upgrade its operations (e.g., waveform, locations, bandwidth), they need strong, reliable means to enforce those rights.

Table 35.1 Requirements of incumbents

| Requirement | Purpose/description |
|--|--|
| Coexistence | Entrants must not cause harmful interference to incumbents; their respective operations must be compatible with each other |
| Accommodate reasonable changes in incumbent operations or technical parameters | Incumbents should be able to change waveform types, occupied bandwidth, locations, frequencies, duty cycle, etc. without objections from entrants |
| Right to use spectrum | Entrants should accept outages or have alternative means to communicate so that incumbents can reclaim shared frequencies at times of need |
| Enforcement | There must be a method to track down and remedy interference events economically and quickly |
| Safeguards/security | Systems must be protected against unauthorized and accidental use and prevent hackers |
| Spectrum-sharing system diversity/complexity | The number and types of entrants should be manageable |
| Trust/confidence among parties | Incumbents must have assurance that sharing criteria and negotiated arrangements will not be challenged or changed except after reasonable amount of time (as mutually agreed) |
| Information security | Incumbents’ confidential information should not be revealed |

35.3.2 Entrant Requirements

Some of the initial requirements that new entrants in shared bands would expect to be implemented through technology, spectrum-sharing criteria, or negotiated arrangements are shown in Table 35.2. Again, the most obvious, mutual coexistence requirement involves some level of assured access to the shared bands and geographic areas without an unreasonable risk of “harmful interference” from the incumbent systems. A related requirement is that entrants need access to a significant amount of spectrum, especially at times of peak capacity. Another key issue for entrants is the need for standardized equipment. This may be a short-term

Table 35.2 Requirements of entrants

| Requirement | Purpose/description |
|--|--|
| Coexistence | Entrants should be able to have access to shared band without being subject to unreasonable interference protection criteria or surprises as a result of, for example, unexpected incumbent system/receiver characteristics |
| Safeguards/security | Systems must be protected against unauthorized and accidental use and prevent hackers |
| Support current architecture | Entrants should be able to use the same type of system architecture in shared bands compared to exclusive bands |
| Minimal changes and standards | Entrants need to purchase standardized, non-proprietary equipment from multiple vendors |
| Low prime power | Spectrum sharing should not significantly increase the power consumption of end user devices |
| Technical flexibility and minimal software integration and equipment upgrade costs | Entrants should be able to deploy alternative and cost-effective spectrum-sharing solutions that provide comparable interference protection to incumbents |
| Capacity and coverage assurance | Entrants’ capacity and coverage expectations in shared bands should be well understood and, subject to negotiation, binding on incumbents; a specified amount of spectrum capacity needs to be available at specified times and places |
| Spectrum and network management workload | Entrant’s spectrum and network management costs must be reasonable |
| Trust and confidence among parties | Entrants must have assurance that sharing criteria and negotiated arrangements will not be challenged or changed except after reasonable amount of time (as mutually agreed) |
| Fair use policy | In the unlicensed entrant scenario, spectrum sharing should have a policy to enable all entrants to have fair use of the shared spectrum |

matter because LTE, CDL, and other popular standards do not currently include advanced spectrum coexistence modalities, but such development is underway or expected over the next several years.

35.3.3 Spectrum Regulator Requirements

The role for regulatory officials (e.g., the FCC and National Telecommunications and Information Administration (NTIA) in the U.S.) is to maximize use of spectrum in a fair and equitable manner. Attempts to reduce spectrum sharing to a uniform, simple set of interference avoidance rules have proven to be very difficult. For example, the FCC TV white space proceeding started in 2004 and is still ongoing. Detailed technical interference avoidance rules for dynamic frequency selection to enable sharing with radar in the 5,470–5,725 MHz band had unintended consequences that made them unusable in practice. Thus, a key issue in both sets of requirements is that firm numerical values or detailed criteria should not be contained in regulations or industry standards, but could be subject to negotiation between or among the parties. There are too many potential sharing scenarios and not enough experience with more complex spectrum-sharing approaches to quantify many parameters. This can be addressed technically by developing specific system design options for policy-based spectrum-sharing systems that can be demonstrated for stakeholders for detailed feedback, which then leads to some quantification or standardization of specific requirements.

On the policy and economic side, the UAS industry needs to develop concrete recommendations to ensure that adequate spectrum-sharing rights and responsibilities can be embodied in the most appropriate and cost-effective manner. Only a few sharing criteria need to be embodied in government rules and regulations. Some may be documented in etiquettes, standards, and best practices, and they can be easily implemented in policy-based systems through machine-readable policies. The bulk of these requirements, rights, and responsibilities would be part of negotiated sharing agreements or through brokered arrangements.

If continuous regulatory involvement is necessary to manage spectrum sharing, then the regulators will impose a requirement that the management approach is required to minimize government investment. For example, a cost-effective enforcement approach will be required to track down and to mitigate any interference events.

35.4 Mechanisms for UAS Spectrum Awareness

There are multiple UAS spectrum-sharing approaches possible that have different features, capabilities, and costs. Viable approaches must be comprehensive and consider all stakeholder requirements. For example, some sharing approaches rely on geolocation (typically via GPS) and frequency assignment databases to determine

what channels an entrant can use without causing interference to incumbents. Other approaches use RF sensing to achieve that same goals. In addition to avoid causing unintended interference, the systems must have a method to enable the incumbent to enforce his spectrum rights in case the entrant system has a fault.

35.4.1 Geolocation and Database Approaches

A geolocation and database spectrum-sharing approach can consist of two types: geolocation and database and geolocation and database pooling.

35.4.1.1 Geolocation and Database Approach

The geolocation and database UAS spectrum-sharing approach uses incumbent and entrant position information, and propagation models to determine entrant operating frequencies as shown in Fig. 35.6. There could be multiple types of incumbents in several spectrum bands and multiple types of entrants sharing the spectrum. There could be multiple types of entrants system, each with different network operators. In this approach a so-called band or spectrum manager (see Sect. 35.5.2.3) collects the position and operating frequency information and estimates the propagation loss and antenna gains between the incumbents and entrants. The spectrum manager then determines the allowable entrant operating frequencies using specified interference to noise ratios (INRs) for co-channel operation and adjacent channel operation. This frequency list is then transmitted to the network operator who then decides what frequency each entrant should use. The network operator then periodically sends the frequency to entrants.

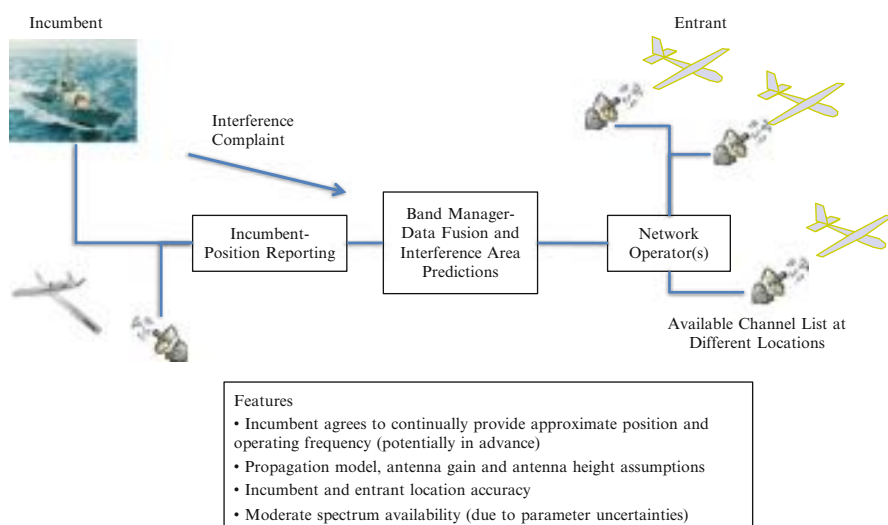


Fig. 35.6 Geolocation and database spectrum-sharing approach

This approach meets the enforcement goal because the entrants are supplying their position to the spectrum manager and they are receiving frequency lists from the spectrum manager. If an incumbent reported interference, it would be relatively straight forward to determine what entrant was causing the problem and to make adjustments.

This type of approach is being proposed for some federal bands for geographic spectrum sharing using exclusion zones or coordination zones (NTIA 2010). The FCC has issued rules for using competitive geolocation and databases to enable access to the television broadcast bands by entrant, low-power devices (FCC 2010).

The geolocation and database UAS spectrum-sharing approach has disadvantages. Agencies operating in federal frequency bands are very unlikely to provide sufficient information (e.g., location and operating frequency) on their operations, especially military uses, that is necessary to populate such databases. Obfuscation techniques such as diluted precision or added false entries may alleviate this concern. Such operations are also likely to involve shipboard, airborne, and other mobile operations that cannot be accurately captured in a database. Accordingly, a stand-alone geolocation and database approach would not likely yield significant spectrum capacity or coverage for new entrants especially if large geographic exclusion zones are the result of information gaps. This information gaps include the actual propagation losses, the current entrant and incumbent positions, the true antenna patterns as a function of angle, and other parameters. The geolocation and database must make assumptions of all of these parameters to determine the available frequencies.

The advantage of the geolocation and database UAS spectrum-sharing approach is that it may be attractive to new broadband entrants because minimal system modifications are required. For the entrant, the process is similar to the already required need to access a policy database. For less dynamic incumbent users, the incumbent reporting requirements may be minimal. For instance, a band for fixed radar would only require database changes when new radars are built.

35.4.1.2 Geolocation and Database Pooling Approach

The geolocation and database pooling UAS spectrum-sharing approach shown in Fig. 35.7 is similar to the geolocation and database approach, except in the pooling approach the spectrum sharing is among peer UAS spectrum users (there is no incumbent users). There are a single spectrum manager. The advantage of this approach is that the allowable interference levels are known and controlled by the UAS community. There are reduced issues about sharing position information since the information stays within the UAS community.

35.4.2 Sensing Approaches

A sensing spectrum-sharing approach can consist of four types: sensor network, off-board sensing, onboard sensing, and onboard sensing pooling. The goal is to identify spectrum “holes” that are not being used by incumbents (Tandra et al. 2009).

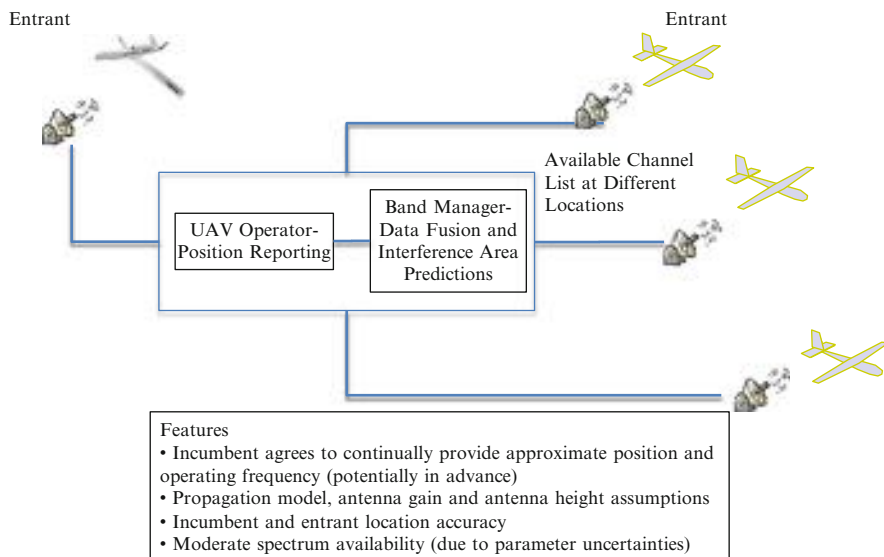


Fig. 35.7 Geolocation and database spectrum-sharing pooling approach

35.4.2.1 Sensor Network Approach

In this approach, a network of interoperable entrant UAS radios acquires context information about the regional spectrum environment by querying a stand-alone external sensor network as shown in Fig. 35.8. The sensors are located near the incumbent system locations.

This approach supports the enforcement requirement. Insufficient densities or uneven sensor distributions can result in a higher likelihood of false positive or false negative spectrum white space detection decisions. Some kind of control channel or “cognitive pilot channel” (Delaere and Ballon 2008) is therefore assumed.

One of the key advantages of this approach is to simplify the entrant radios, which would result in reductions in cost and energy consumption (Weiss et al. 2010). Another advantage is that this approach also improves the availability of spectrum white space based on superior local knowledge (the exact incumbent operating frequency and operating state is known) when compared to a stand-alone geolocation and database approach (where the incumbent operating frequency and operating state are inferred by reports and predictions).

The disadvantage of this system is the cost to install and to operate the dedicated sensor network. Another disadvantage is that the incumbent might operate in a nonstandard location (e.g., respond to hurricane in New Orleans) where there are no dedicated sensors.

35.4.2.2 Off-Board Sensing Approach

In this approach, off-board sensors are colocated with or integrated into the network base stations as shown in Fig. 35.9. Standardized or separate proprietary sensors are

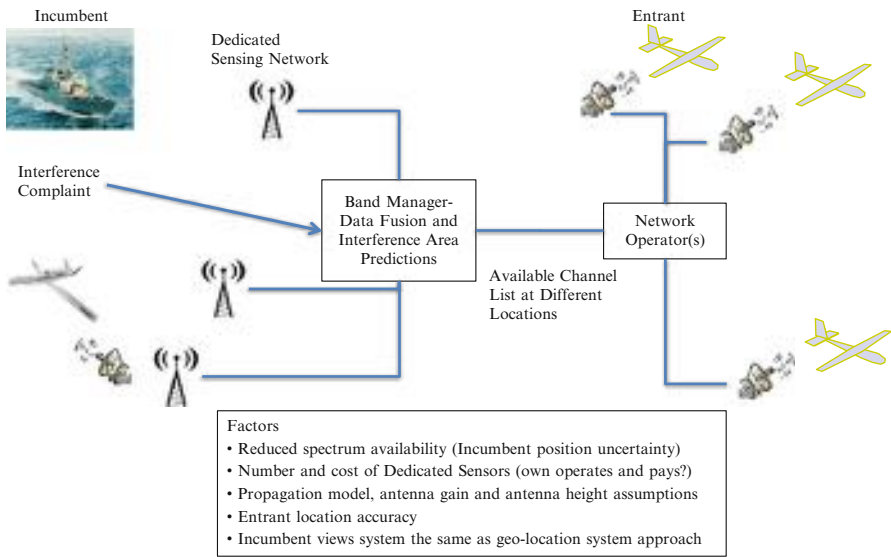


Fig. 35.8 External sensor network spectrum-sharing approach

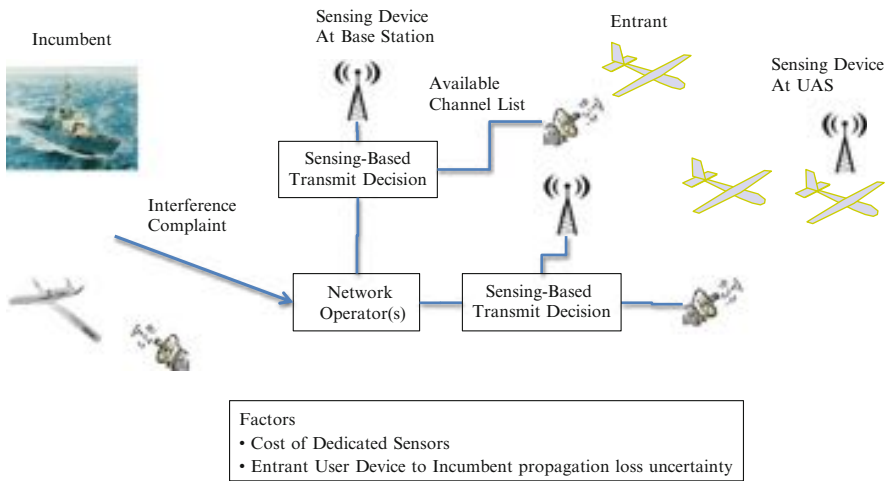


Fig. 35.9 Sensing on some entrant radio spectrum-sharing approach

used that incorporate complex technical methods to “look through” the entrant system co-channel interference to detect the weak incumbent signals. They sense the local spectrum environment with improved coverage density and distribution, thereby reducing incorrect spectrum availability decisions. Since these sensors are within close proximity to and interoperate with the end-user devices, they provide better local spectrum environment information when compared to the previous approach.

Therefore, the principal advantage of this approach is increased spectrum availability for the broadband system because of less uncertainty in determining the propagation losses between protected emitters and the broadband system base stations. However, the “look through” method might not provide enough detection sensitivity for interference free operation.

Another potential disadvantage of this approach if proprietary software solutions are used, is the nonrecurring development and testing costs required for full UAS entrant transceiver integration into the base station radio. A short-term solution involves a colocated, nonintegrated hardware/software sensor, which also adds to the incremental costs. This method could also be more difficult and costly to implement for ad hoc and low-power networks with no fixed infrastructure.

35.4.2.3 Onboard Sensing Approach

In this approach, all or a large number of interoperable UAS entrant transceivers sense the RF environment directly and make operational decisions based on those inputs as well as information gathered from neighboring, collaborative devices as shown in Fig. 35.10. The onboard sensors integrate coordinated sensing periods into the entrant waveform, measure the signal power of incumbent transmitters, and share these measurement results with other UAS entrant transceiver radios.

This approach supports the enforcement requirement because each UAS ground station is in contact with a network operator. If there is an interference event with an incumbent user, the network operator is contacted by the spectrum manager, and the offending UAS or ground station is identified and fixed. Each UAS has

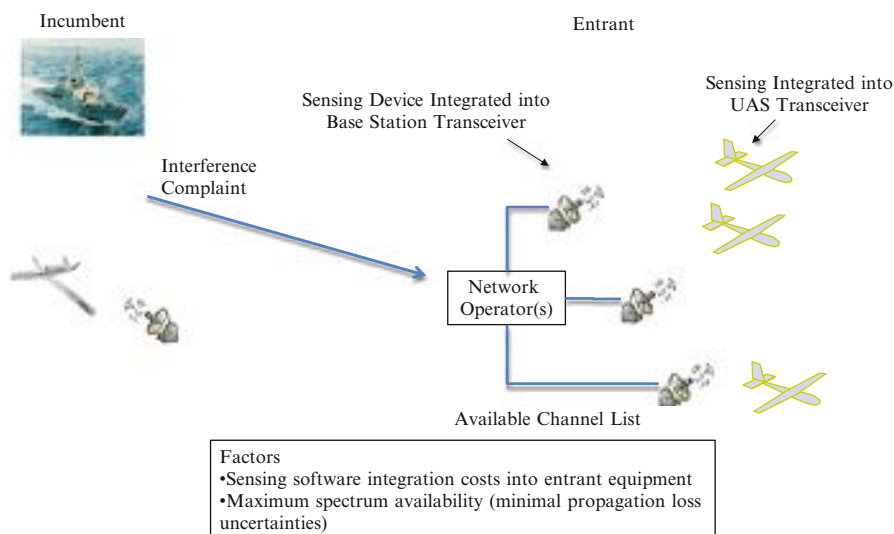


Fig. 35.10 Sensing on all entrant radio spectrum-sharing approach

a time-limited set of operating frequencies, so if it loses contact with the ground stations, it reverts to a “safe frequency” that is not likely to cause interference to an incumbent user.

The greatest advantage of this approach is that it provides the maximum spectrum availability in terms of capacity and coverage for broadband systems because of the minimal uncertainty in determining propagation losses between the incumbent systems and the device itself. Another advantage is that there are no extra hardware costs since the UAS radios will only sense in bands in which they already transmit.

A main disadvantage for short-term prospects under this approach is that non-recurring software and API development and testing are necessary to fully integrate the spectrum-sensing software in end-user devices. In addition, although some spectrum-sensing concepts (such as using the RSSI (received signal strength indicator) to select the operating frequency) are in the Long-Term Evolution (LTE) and other standards, further spectrum-sensing standards development must continue (IEEE DySPAN 2011). In light of the current lack of standards for embedding full spectrum-sensing functionalities together with the initial costs to upgrade end-user devices, this approach may be of less interest to service providers and manufacturers in the near term. Nevertheless, the preferred, long-term commercialization approach would lead to widespread development and deployment of spectrum-sensing-enabled end-user devices with onboard sensing capabilities.

Cooperative sensing information could be used to improve the incumbent radio detection probability and/or reduce the UAS sensing requirements. The problem with cooperative sensing is that if there are not enough UASs participating, then there is no sensing improvement. The requirement to operate with a single UAS sets the sensing requirements.

35.4.2.4 Onboard Sensing Pooling Approach

In this approach, all or a large number of interoperable UAS entrant transceivers sense the RF environment directly and make operational decisions based on those inputs as shown in Fig. 35.11. There is no collaboration with other UAS.

The advantage of this approach is the lowest deployment and operational costs. This approach does not require any backhaul between the ground stations, hence, is applicable to remote UAS operation. This approach is used by the military for its dynamic spectrum access tactical radios.

This approach does not directly support the enforcement requirement because the UAS positions are unknown to the incumbents and the incumbents do not have direct control of the UAS operation.

35.4.3 Beacon Approaches

A transmitter beacon spectrum-sharing approach can consist of three types: per radio, area, and entrant signaling.

Fig. 35.11 Sensing on all entrant radio spectrum-sharing pooling approach

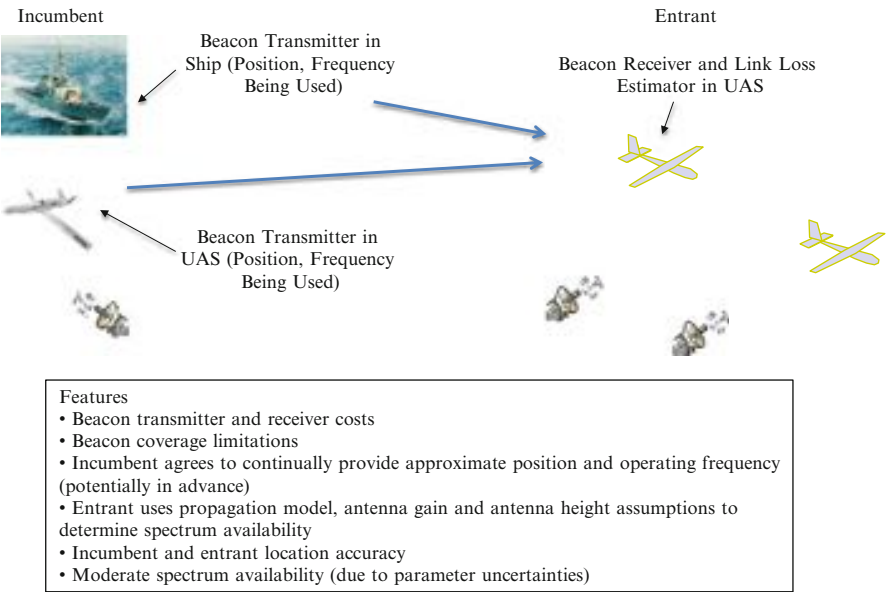
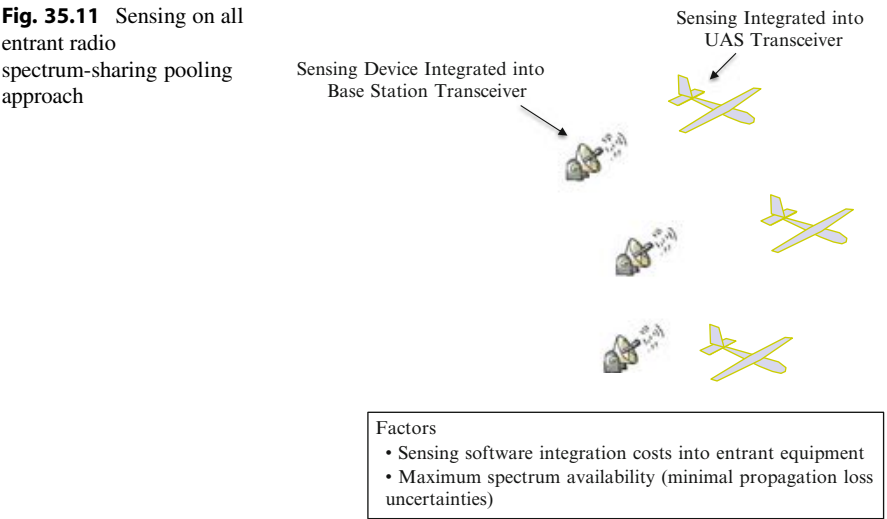


Fig. 35.12 Per radio beacon spectrum-sharing pooling approach

35.4.3.1 Per Radio Beacon Approach

A direct approach to alerting the presence of an incumbent service to entrant devices is for each incumbent transmitter or receiver to have a standardized, easy to detect, simple to receive beacon as shown in Fig. 35.12. To simplify the search for these beacons, they should be located at carrier frequencies that are standardized for

each incumbent band. The beacon could be as simple as a carrier tone. This has the potential to interfere with the incumbent signal or for spurious signals and harmonics from other bands to be detected. A more reliable approach would be to modulate the carrier with a random direct sequence spreading code. The spreading code would make it much less likely that unwanted signals could lead to false-positive detections. It could also spread the signal over a greater bandwidth so as to cause less interference to the desired incumbent signal. The beacon can encode information on the beacon signal such as the location of the transmitter, its transmit power, or a definition of its coverage area.

The advantage of this approach is more reliable channel detection while keeping full control of the beaconing with the incumbent service provider. Further, each transmitter can beacon the precise channels they are using rather than the set of channels they could be using.

The disadvantages of this approach are the beacon transceiver costs, the difficulty in integrating the beacons to existing incumbent devices, many incumbents may not want to transmit a beacon signal for security reasons, and the limited beacon link range compared to the entrant-to-incumbent interference range.

35.4.3.2 Area Beacon Approach

Instead of each transmitter having to integrate a beacon into its signal, a small amount of bandwidth could be set aside for dedicated beacons that would act as a service for all the incumbent transmitters in an area as shown in Fig. 35.13.

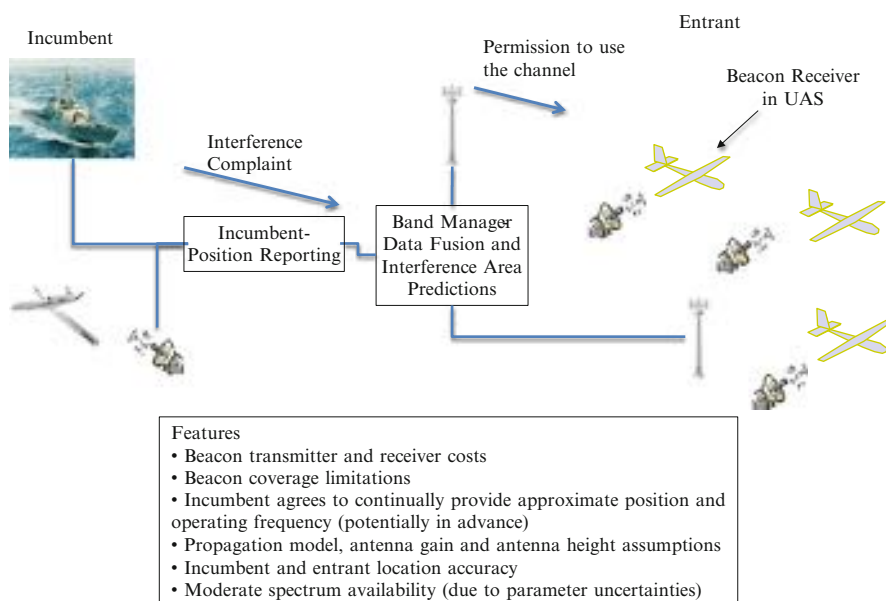


Fig. 35.13 Area beacon spectrum-sharing approach

Being in a dedicated channel, they can be higher power. Using spread spectrum techniques similar to CDMA cellular service, the same beacon channel can be used throughout the country, with different offsets of the same spreading code used for different beacons to prevent inter-beacon interference. Each beacon can provide a low-data rate service that announces the frequency band, location, and coverage area of each incumbent transmitter in the area. The required bandwidth can be small (e.g., 12.5 kHz). The power can be relatively high (relative to the bit rate) so that the coverage is large at modest power (e.g., 100 W). Approximately 1,000 area beacons could cover the continental USA.

In essence this would be a push model for the geolocation and database method with the exception that the information is limited to primary radios in the area of the beacon. The beacon information would be derived from an incumbent database. The beacons can independently choose the information they broadcast depending on their location and intended coverage area. The incumbent users can be shielded from when beacons are added, moved, or removed. This provides a level of separation between the incumbent users and the beacons. Incumbents edit the database. The beacons read the database.

This method has the advantage of providing a dedicated standardized access to the incumbent user database that avoids the need for Internet access. Since presumably there are relatively few beacons compared to the number of entrant device, the load on the central database would be minimal and therefore the database could be simpler and lower cost. Like the database method, it requires the entrant devices to have a location for themselves, although the beacons themselves provide a crude localization. For instance, the beacon might include a list of channels that can be used by any entrant device that can receive the beacon signal. In congested areas, this list may be empty, but in rural areas, this is sufficient precision to identify many unused channels. The disadvantage to this approach is the cost of the beacon network and the need for the entrants to be able to access these beacons which may be at frequencies quite different from the desired operating frequencies.

35.4.3.3 Incumbent Signaling

A more distributed option is to set aside a small frequency band for entrant devices to exchange information about incumbent transmitters. It would provide a known open channel that the entrant devices could use to signal each other about the presence of incumbent and entrant services in the area. Like the area beacons, the format could be standardized as an etiquette protocol that would enable disparate devices to communicate with each other. This would be used to augment any of the other methods in this chapter and enable new methods. For instance, entrant users might set up dedicated monitoring devices at prominent locations to identify and locate incumbent transmitters. These would in turn broadcast their findings for a user community. This could all be done on an existing entrant band, but, usable bands like the 902–928-MHz ISM band are far from the incumbent bands below 100 MHz and would add expense to the entrant radio antenna and front end if it was required to operate over this wide a range of frequencies. Alternatively, a small amount of spectrum could be set aside in or near each incumbent service frequency band.

35.4.4 Hybrid Approach

The choice of which approach to use depends on many factors. The geolocation and database approaches and the beacon approaches impose a burden on the incumbent to either maintain database information or to install beacons. The sensor approach imposes fewer burdens on the incumbent although for best detection, the sensors would need to know and track specifics of the incumbent waveform.

From the entrant perspective, each of the approaches imposes different burdens. The geolocation and sensing approaches require access to a database. The sensing imposes a burden of requiring the sensing capability as well as potentially using some spectrum resources for the sensing operation. The beacon may require the entrant to build a beacon network and require the incumbent to be able to populate the information in the beacons. In the case of the pooling approaches, the entrants need protocols and signaling channels in order to coordinate.

In terms of protection of the incumbent user, all of the approaches can be made sufficiently conservative so as to avoid harmful interference to the incumbent. Generally, as the approaches become more conservative, less bandwidth is available to the entrant. The entrant will miss opportunities to communicate if its access to the database is infrequent or its geolocation is imprecise. Sensing of weak radio signals is notoriously difficult and the number of false positives (where an entrant decides not to use a frequency band that, in fact, has no incumbent) high. Per radio beacons can be more precise but only to the extent that the beacon propagation matches the interfering signal propagation. Area beacons have the same limitations as the database approach.

Each of the approaches has advantages and disadvantages. Therefore, a combination of the above techniques may be used to address short-term cost and performance concerns. For example, onboard sensing could be supplemented with off-board or other sensing or database information.

35.5 Mechanisms to Manage UAS Spectrum Access

Spectrum access is the process by which the cognitive radios in the UAS choose which frequency bands to communicate. It consists of several dimensions related to the type of spectrum, which parties control its use, and how it interacts with UAS operations. Conceptually, spectrum access can be thought of as having a policy that permits usage of a specific band for a specified period.

As an example, consider ATC communications for a manned aircraft at a small airport. Aviation maps indicate the ATC channel for the airport. So based on location and phase of flight (i.e., in Start and Takeoff), the pilot has an existing “policy” for this situation to use the ATC band on a shared basis without asking permission. After the plane takes off and enters the en route phase of flight the access changes. Now the pilot does not have an existing “policy” for using an ATC channel. The pilot is assigned a channel by the en route controller who periodically assigns the

pilot to new channels based on plane location. In this case, the channel used is not in a set of stored policies selectable by the pilot. Instead, the policy is dynamically assigned by the ATC center. Like the airport channel, the en route channel is shared; however, access to the channel is controlled and limited. The goal of the cognitive radio is to automate these processes while retaining the level of control required by the different stakeholders.

35.5.1 Types of Spectrum Access

Spectrum access can be classified along two dimensions. In one dimension, spectrum access can be *licensed* or *unlicensed*. Licensed refers to an access where UASs have to ask for a permission before they can use spectrum. Unlicensed refers to an access where UASs can use spectrum without having to ask for any permission. But, they may still have to follow certain rules.

In another dimension, spectrum access can be *interruptible* or *uninterruptible*. Interruptible refers to when a UAS entrant accesses the idle incumbent spectrum on the condition that UAS does not interfere with the incumbent. UASs have to vacate the spectrum if the incumbent starts to transmit. In other words, incumbents have priority over UAS entrants.

Uninterruptible, in contrast with interruptible, refers to an access where UASs do not have to avoid any other incumbents but may have to avoid interfering with other UASs via the use of some etiquette protocol. There are no other spectrum users having priority over UASs.

The two dimensions combine as shown in Table 35.3 and described below.

Licensed, Uninterruptible: One version of this model resembles the traditional approach for licensed spectrum use. UASs are the incumbents and have exclusive rights to use spectrum granted by the spectrum regulator. But the spectrum assignment in this model could be more dynamic – the spectrum rights could be changed frequently and assigned quickly. In a further variation, UASs lease spectrum from other spectrum holders and temporarily have the rights to use the leased spectrum for a specified period. During the lease period, the spectrum holders cannot interrupt the UASs.

Unlicensed, Uninterruptible: UASs use designated unlicensed bands like the 2.4-GHz ISM band.

Unlicensed, Interruptible: UASs are unlicensed entrants accessing white spaces in incumbent’s spectrum. Certain bands that are normally used by incumbents are available to UASs for unlicensed operation when and where the incumbents are not using channels. UAS radios have to be capable to avoid interfering with primary

Table 35.3 Types of spectrum access and an example of each

| | Uninterruptible | Interruptible |
|------------|-----------------|----------------------------|
| Licensed | Cellular | Secondary spectrum leasing |
| Unlicensed | WiFi | TV white space |

users and vacate channels if primary users appear. This model is similar to the approach in which the FCC has allowed TV band devices accessing TV white spaces (FCC 2010).

Licensed, Interruptible: This is a secondary spectrum-leasing model, where UASs lease spectrum from other incumbents but must avoid interfering with these incumbents. This model is similar to the licensed, uninterruptible spectrum leasing, but the difference is that the spectrum holders can interrupt the UASs anytime as they wish. If the incumbents interrupt, UASs have to vacate the spectrum. This model is also similar to unlicensed, interruptible model, but the number of entrants is limited by licensing.

Each model could be further divided into fixed assignment and dynamic assignment. In fixed assignment, users take control of spectrum for a long time, e.g., 10 years, while in dynamic assignment, users use spectrum for a relatively short time, e.g., a few minutes, hours, or days.

To support more dynamic leases, policies have a notion of a *time-limited lease* (Chapin and Lehr 2007). This implies that each policy has a specified period over which it is valid. The specified period could be specific dates it is valid or short time intervals. For instance, a control channel for landing might be assigned in a policy valid for a 10-min period during final approach after which it would be released for other landing aircraft.

35.5.2 Control of Spectrum Access

Spectrum access is controlled by policies. What are the mechanisms to distribute and manage policies that underpin the control of spectrum access? To address this question, it is necessary to examine where the policies come from, where they are ultimately used, and how the process is mediated.

35.5.2.1 Where Do Policies Come From?

An important concept is the leasing and sub-leasing of spectrum through policies as illustrated in Fig. 35.14. Policies originate from base authorities. In the United States, this might be the FCC or NTIA. The NTIA through a policy grants the Federal Aviation Administration (FAA) control over a specific frequency range and set of locations (a specific 10-MHz block over the continental USA for 5 years in this example). This chain is extended further by the FAA providing policies to other entities. In the example, the FAA grants the Phoenix Sky Harbor International Airport (PHX) the rights over the 10 MHz in the vicinity of the airport (which, for simplicity, is described as Arizona). In turn, PHX can assign channels to specific aircraft.

The policies used to access spectrum must be valid in three ways. First, they must come from a source which is authorized to issue a policy. An end user needs to see not only the final policy but also the chain of policies tracing back to a trusted root source (in this case, the NTIA). Because policies might be corrupted or maliciously modified in transit, the policies need to be cryptographically protected with a digital signature so that the end user can verify that each policy originated

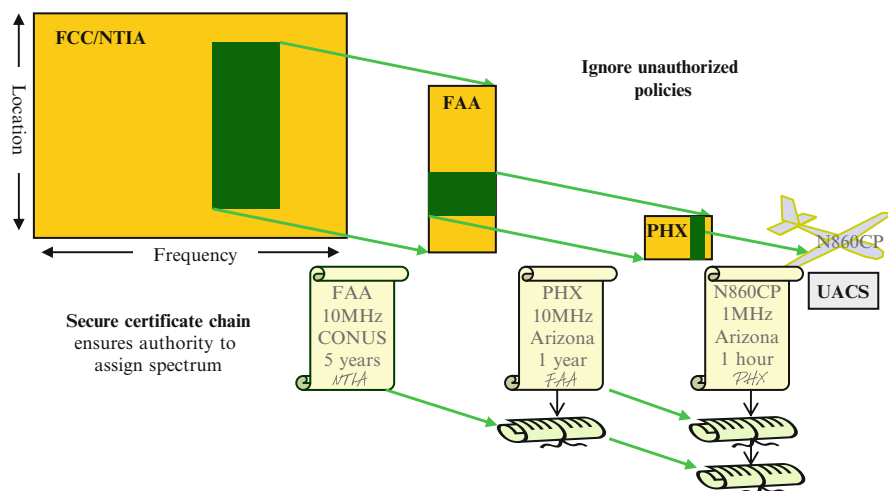


Fig. 35.14 Certificate chain

from the purported source (Stallings 2000). Second, they must be consistent and not contradict other policies. The characteristics of the policies such as the spectrum bands and time period for each subpolicy must be a subset of the higher-level policies. Further, there may be other prohibitive policies that exclude use in certain bands. The end user must validate the policy it uses against these other policies. To enable this validation and reasoning, the policies must be written in an appropriate machine-readable form (IEEE 1900.5). Third, the conditions of the policy must match the situation to which the policy will be applied. The end user must validate that it meets the location, type of communication, and other conditions of the policy before it uses the policy. In particular, the bands must be available according to the spectrum awareness function and the radio must be capable to transmit in the band.

Since the FAA/NTIA can grant such spectrum policies to multiple sub-entities which in turn can grant to multiple sub-subentities. This process produces a distributed policy tree with each node only required to (a) store the chain of policies to the root of the tree that validates its authority and (b) manage issuing policies to their direct child nodes. Policies can be issued in a chain architecture where a child policy includes the parent, grandparent, etc., policies. This simplifies the policy validation since every policy contains all the required information for the end user to validate the policy. However, in an environment where a child policy changes often, this increases the burden on the system since the entire chain has to be sent every time. In the above example, a plane that travels often to PHX would receive the entire three-policy chain every time it used the airport. An alternate approach is a web approach where the end user downloads the policies one by one. Here the challenge is that the end user needs to have access to and be able to find all the policies in the chain. This can be simplified by embedding short “resource locator” information about the needed higher-level policies. In this case, the parent policies

are stored and only the child policy changes. Note that the public keys required for the digital signatures can be distributed either embedded in the policies themselves or according to their own distribution chain.

35.5.2.2 Where Are Policies Used?

The end user of a policy could be the cognitive radio in the UA or the cognitive radio at the other side of the connection (in the control station, satellite, or network radio). In this cases the one side or the other that is receiving the policies would control the choice of frequency band used for communication. In using a policy, it is possible for the two cognitive radios to be widely separated. Thus, in applying the policy, the end user must consider the situation at both sides of the connection. In some cases, both sides of the connection may hold their own set of policies and neither is controlling the choice. Because they are widely separated, they may hold different spectrum awareness views. In the worst case, there may be no connection and the problem is for the radios to find a common spectrum band to *rendezvous* in order to start communication (Akyildiz et al. 2006).

Policies can also go to some other controlling entity. For instance, a policy might go directly to a UAS operator who performs all of the validation of a policy and then through a trusted link sends the frequency assignment to the UAS. Or, policies might go to another agent who similarly acts on behalf of the UAS.

Some policies are *open-ended*; anyone holding the policy that meets the conditions of the policy may use the policy. Other policies are assigned to use by *specific* end users.

35.5.2.3 How Are Policies Mediated?

In the above example, the policy's authority follows through specific organizations. More generally, these policies are mediated by a *spectrum manager*. The spectrum manager is responsible for gathering policies from appropriate agencies and distributing them to sublessees. A spectrum manager may be an agent within an organization or a separate designated third party.

There are several models for how spectrum policies can be distributed. In a *shared* model, policies may give access to a band of spectrum to a set of end users who must coordinate their access. At one extreme, the access is open-ended in terms of the number of users who are given access and as to the period that it is available. Like the ATC band at the airport, many users potentially have access for an extended period of time. There may be operational limits to the number of users on a band (only so many airplanes can use the airport) or other limits (a band that is busy will simply not be used). Such shared use works better when there are rules or etiquettes (e.g., listen before talk) to manage communication. To avoid channels from becoming overloaded and unusable, a *limited shared* model may be used where the set of users that have access are limited in number or time. At the other extreme is a *dedicated* model where a spectrum band may be given to only one end user at a time. The limited and dedicated models require the spectrum manager to track which users have active policies at any given time. To better control the access, the spectrum manager may flag a policy as being *terminal*, meaning that it cannot be subleased further.

Policies can be created on an *as-needed* basis. In this model, based on the operational situation, the spectrum manager issues policies to the end user as needed to fulfill specific needs. For instance, an airport may issue policies to UA on approach to provide them with a dedicated high-quality band for controlling the landing of the aircraft. Policies can be *negotiated*. In this model, the end user may make a request to a spectrum manager for a policy to fill a specific need. The spectrum manager responds with a policy that meets the request or with a message that the request cannot be met. In the latter case, the end user may make a revised request seeking to find the spectrum it needs. Policies can also be *premade* and warehoused in a database accessible to end users. Here a user visits the database, downloads suitable policies, and makes its own decision about which policy to use. The premade model works best for distributing shared policies and for storing higher-level policies that users can use to validate end-user policies.

A problem that is particular to UAS is being sure that there is spectrum available to control the aircraft throughout its flight plan. In a *real-time spectrum* model, the UAS dynamically selects policies over the course of the flight. For shorter-range and shorter-duration flights, this makes sense. Some UAs are large and costly and can fly for more than a day over distances spanning a continent. It would make sense for such a plane to have a spectrum plan to match its flight plan. In the *preplanned spectrum* approach, the UAS gathers policies not only for its current needs but also for its needs over the entire flight plan. The previous two models make the communication depend on the flight operations. A third model is a coordinated model that connects the communication and flight operations. In the *coordinated spectrum and flight* approach, the aircraft modifies its flight path as needed to ensure that spectrum is available. For instance, in the networked radio approach, if the UA flight path crosses a radio cell that is currently at capacity, the UA can modify its flight path. The flight path could be routed around the busy cell through cells that do have capacity. Or the aircraft could slow down to provide time for traffic in the cell to subside. If necessary, the plane might even loiter at the edge of the cell until the needed spectrum is available. Conversely, the spectrum manager for a cell can consider the flight plan, endurance, and other factors of incoming aircraft in prioritizing which aircraft are given a channel.

Note that spectrum access interacts with spectrum awareness. In the case of the geolocation and database approach, the spectrum availability in the database can be expressed directly in policies similar to this section. For sensing, the policies can specify how often sensing is necessary, the sensitivity levels for detection, and the consequences of positive detection. For beacons, the policies can specify how often beacons must be received and other conditions. So, in this way, policies can mediate the interaction between different spectrum awareness schemes. For instance, some bands may be allowed without any sensing. In this way entrants without sensing can still access spectrum. Other spectrum will specify sensing or beacon requirements that will allow radios with these capabilities further access to the spectrum.

This section emphasizes the flexibility with which spectrum access can be managed through the use of policies. A summary of all the choices in a UAS cognitive radio design are given in Table 35.4.

Table 35.4 UAS cognitive radio architecture choices

| Property | Design choices |
|---------------------------------------|---|
| UAS C3 architectures | Direct Satellite Ground network Mesh |
| Spectrum awareness | Geolocation Geolocation pooling External sensing network Sensing on some entrant radios Sensing on all entrant radios Sensing on all entrants pooling area beacon per radio beacon Incumbent signaling Hybrid approach |
| Spectrum access | Unlicensed Licensed |
| Entrant response to incumbent | Uninterruptible Interruptible |
| Policy authentication | Embedded-entire chain in every policy Web-chain assembled from multiple policies |
| Public key distribution | Embedded in policies Separate infrastructure |
| Frequency choice | Assigned by UA CR Assigned by other CR Assigned by third party Rendezvous |
| Policy assignment | Open-ended Specific end users |
| Spectrum exclusivity | Open shared (unlicensed) Limited shared Dedicated |
| Policy creation | As needed Negotiated Premade |
| Spectrum interaction with flight plan | Real-time spectrum Preplanned spectrum Coordinated spectrum and flight |

35.6 Example

In this section, an example is given of a cognitive radio-enabled UAS. In this fictitious example, a medium altitude long-endurance unmanned aircraft operated by the U.S. Customs and Border Protection Agency (CBP) is tasked with flying along

a coast to monitor marine traffic. The aircraft is controlled from a remote ground control station. The example follows the aircraft through different phases of flight.

Flight Planning: As the plane is prepared for its mission by the ground crew, the autopilot indicates that the plane is in the flight planning phase of flight. For this phase, any spectrum (e.g., unlicensed interruptible) can be used. To exchange flight plan data, update maintenance records, and upgrade onboard software, the plane needs high bandwidth. Delays are tolerated and higher-layer (i.e., TCP) protocols compensate for packet losses. Therefore, based on its location, phase of flight, and a long-term premade policy, the UA identifies the 2.4-GHz ISM band as acceptable. This is a shared unlicensed band. The policy chain is direct from the FCC. However, the policy was downloaded months before from an intermediate CBP policy database. The policy is validated using a well-known public key for the FCC. Also at this time, the UAS negotiates a policy for the spectrum used during the en route and aerial work phases of flight.

Start and Taxi: Control of the plane is passed to a distant UA control station. A ground-based network communicates with a transmitter located at the airport that communicates directly with the plane. At this point the UA chooses a premade policy for UA ground operations issued by the local airport and stored in a public database. This is a policy specifying a shared unlicensed uninterruptible band that is restricted to aircraft in start and taxi operations. However, the spectrum is allocated to an incumbent for a future different but, as yet, unneeded purpose. Every week, the incumbent user issues a policy good from the time of issue until a month into the future. During this month, the incumbent revokes its right to interrupt and use the spectrum. When the incumbent wishes to regain use of the spectrum, it stops issuing new policies. Since this policy is downloaded infrequently (twice a month would be sufficient), the entire policy chain is embedded in the policy along with the public key chain.

Takeoff and Departure: The pilot indicates that the plane is ready for departure. At this point the plane is assigned a policy that is valid for 15 min using a spectrum appropriate for this airport. If the plane should be delayed on the runway, additional policies would be assigned that extend the time period. The spectrum is a dedicated licensed uninterruptible spectrum allocated for this purpose, and the UA is assigned a policy for one of the channels in the allocation. Communication is direct between the UA and a radio at the airport. To simplify policy assignments, only the last link of the policy chain is sent. The plane must have stored the higher links in the policy chain. The plane separately downloads public key information.

En route: As the plane gains altitude, the autopilot indicates that the plane has entered the en route phase and the plane switches to its preplanned en route frequencies. The policy used for departure is released. While en route, the plane uses a network of ground radios as it flies from inland toward the ocean. Each ground radio uses spectrum in its area on a secondary basis from unused white space. The spectrum is a mix of interruptible and uninterruptible spectrum. There is, in fact, a surplus of spectrum for this purpose, and statistically, the plane has enough preplanned policies such that the likelihood of having its entire spectrum interrupted is negligible. Further, though spectrum is shared with other UAS, the preplanning

ensures that each ground node always has sufficient spectrum to serve all aircraft in its cell. If planning would have revealed an overload, then alternate routes or times would have been suggested to the UA at the preflight stage. Much like a cellular mobile system, the ground cells direct the plane as to which channel it should use at any time. Through a network of sensors, the ground stations identify the presence of incumbents and guide the use of spectrum. En route communication traffic demands are small and sporadic so access to spectrum is shared for efficiency but limited to ensure good performance. Because of the potentially many sources of spectrum, the full policy chain and public keys are embedded in each policy.

Aerial Work: Reaching the ocean, communication switches to a licensed satellite channel to provide UA C3. The UA would also like to send video and sensor data back to the control station. Since the channels are limited and aerial work operations are difficult to preplan, the UA switches to a coordinated spectrum and flight operations mode. For instance, upon reaching the ocean, if no satellite channel is available, the UA might loiter until a channel becomes free. Once under the control of the satellite, the plane must negotiate periodically for how much spectrum it gets at any time. During some times it can backhaul all video and sensor data. At other times only basic C3 data can be sent. To simplify negotiations, since all but the last link of the policy chain is the same in each policy, the initial links are sent once and only modifications are sent at each renegotiation.

Emergency: While over the water, the UA engine loses power and must find a place to land within a 75-mile radius. The plane has a set of preloaded policies for this situation. Initially, it uses the dedicated satellite channel. But as it approaches the airport, it needs to switch to a more responsive ground-based architecture. It has a premade policy which directs it to use one of several unlicensed frequency channels but only as allowed by an area beacon located at the airport. Though unlicensed, it is only allowed for emergency operations. The beacon specifies which band is currently reserved for emergency operations and helps facilitate rendezvous with the control station radio should the emergency be communication related. With this frequency band, the emergency landing is successful.

The specific architecture choices made at each phase of flight are summarized in Table 35.5.

35.7 Conclusion

Cognitive radio is a flexible technology that supports different communication architectures and enables access to more spectrum for UAS operations. Fallow spectrum can be used through time-limited policies so that a spectrum manager can manage when and where spectrum can be accessed. Different spectrum awareness schemes ensure that UAS are aware of available spectrum and avoid causing harmful interference to incumbents. By resolving the static allocation issues, spectrum managers would be more willing to offer up spectrum, thus making more spectrum available and less spectrum left fallow.

Table 35.5 Summary of architecture choices made during example flight

| Property | Flight planning | Start and taxi | Takeoff and departure | En route | Aerial work | Emergency |
|-------------------------------------|--------------------|--------------------------|-----------------------|--------------------------|-------------------------|--------------------|
| Policy chain | FCC→all | NTIA→primary→airport→all | NTIA→airport→end user | Various→network→end user | NTIA→satellite→end user | NTIA→airport→all |
| UAS C3 architectures | Direct | Direct | Direct | Ground network | Satellite | Direct |
| Spectrum awareness | Geolocation | Geolocation | Geolocation | External sensing | Geolocation | Area beacon |
| Spectrum access | Unlicensed | Licensed | Licensed | Unlicensed | Licensed | Unlicensed |
| Entrant resp. to incumbent | N/A | Uninterrupt. | Uninterrupt. | Interruptible | Uninterrupt. | N/A |
| Policy authentication | Embedded | Embedded | Web | Embedded | Web | Embedded |
| Public key distribution | Separate | Embedded | Separate | Embedded | Separate | Embedded |
| Frequency choice | UA CR | UA CR | Third party | Other CR | Other CR | UA CR |
| Policy assignment | Open-ended | Open-ended | Specific end user | Specific end user | Specific end user | Open-ended |
| Spectrum exclusivity | Open shared | Open shared | Dedicated | Limited shared | Dedicated | Open shared |
| Policy creation | Premade | Premade | As needed | As needed | Negotiated | Premade |
| Spectrum interaction w/ flight plan | Real-time spectrum | Real-time spectrum | Real-time spectrum | Preplanned spectrum | Coordinated | Real-time spectrum |

The cognitive radio approach can also be used to increase flight safety. Spectrum arrangements can be made which can more easily be checked for consistency since they are explicitly coded in machine-readable form. Further, the cognitive radio enables appropriate spectrum to be used in each phase of the flight allowing more bandwidth to be safely used as illustrated in the extended example. Spectrum usage can be planned before flights to assure sufficient spectrum is available and the cognitive radio can more easily avoid channels with interference and other impairments increasing communication reliability.

Cognitive radios enable new spectrum allocation and business models. For instance, a spectrum owner can rent spectrum for individual UA flights. The UA operator based on its flight plans could get leases for specific time periods covering the flight. The spectrum can be rented long in advance (e.g., a month) and a policy issued immediately specifying the frequency for the period of the planned flight. The spectrum owner is ensured that the spectrum will be used only as planned and the UA operator can better prepare its flights.

References

- I.F. Akyildiz, W.Y. Lee, M.C. Vuran, S. Mohanty, NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Comput. Netw.* **50**(13), 2127–2159 (2006)
- F. Box, L. Globus, Y.S. Hoh, I. Gheorghisor, J.L. Costa, *Throughput Requirements for Control and Non-Payload Communications of a Single Unmanned Aircraft (SC-203 Working paper)*, Washington, DC, Radio technical Committee for Aviation. Retrieved from <http://www.rtca.org>
- T.X. Brown, A. Sethi, Potential cognitive radio denial-of-service vulnerabilities and protection countermeasures: a multi-dimensional analysis and assessment. *J. Mobile Netw. Appl.* **13**(5), 516–532 (2008)
- J.M. Chapin, W.H. Lehr, Time-limited leases for innovative radios, in *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*. DySPAN 2007, 2007, pp. 606–619. doi: 10.1109/DYSPAN.2007.85
- S. Delaere, P. Ballon, Multi-level standardization and business models for cognitive radio: the case of the cognitive pilot channel, in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*. DySPAN 2008, 2008, pp. 1–18. doi: 10.1109/DYSPAN.2008.67
- Federal Aviation Administration, Order 7610.4J: special military operations, Change 2 Effective, 12 July 2001
- Federal Aviation Administration, Meeting the Challenge: Unmanned Aircraft Systems. *Fed. Aviat. Adm. R&D Rev.* **4** (2006)
- Federal Communications Commission, Spectrum policy task force report (ET Docket No. 02–135), Nov. 2002. http://hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-228542A1.pdf
- Federal Communications Commission, In the matter of Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing CognitiveRadio Technologies (ET Docket No. 03–108) and Authorization and Use of Software Defined Radios (ET Docket No. 00–47). ET Docket No 03–222 Notice of proposed rulemaking and order, 30 Dec. 2003
- Federal Communications Commission, Second Report and Order, Order on Reconsideration, and Second Further Notice of Proposed Rulemaking, In the Matter of Promoting Efficient Use of Spectrum Through Elimination of Barriers to the Development of Secondary Markets, WT Docket No. 00–230 (2004)
- Federal Communications Commission, Second Memorandum Opinion and Order, In the Matter of Unlicensed Operation in the TV Broadcast Bands, Maxwell AFB, AL, ET Docket No. 04–186 (2010)

- M.E. Griswold, Spectrum Management: Key to the Future of Unmanned Aircraft Systems? Air War College Maxwell Paper No. 44 (2008)
- S. Heppel, UAS control and communications architectures. SC203-CC005, RTCA (2008)
- IEEE 1900.5, Working group on policy language and policy architectures for managing cognitive radio for dynamic spectrum access applications. <http://grouper.ieee.org/groups/scc41/5>
- IEEE DySPAN Standards Committee, Comments in the matter of "Promoting More Efficient Use of Spectrum Through Dynamic Spectrum Use Technologies," ET Docket No. 10-237. <http://fjallfoss.fcc.gov/ecfs/document/view?id=7021034092>, Feb. 11, 2011
- International Telecommunication Union, Agenda for the 2015 World Radiocommunication Conference, ITU, Resolution 807 (WRC-12) (2012)
- J. Mitola III. *Cognitive Radio Architecture: The Engineering Foundations of Radio XML* (Wiley, Hoboken, 2006) p. 473
- M. Neale, UAS Control and Communications Link Spectrum Considerations. RTCA SC-203 Issue Paper WG2-S1-001-A-SAPP, Aug 2007
- NTIA, An assessment of the near-term viability of accommodating wireless broadband systems in the 1675-1710 MHz, 1755-1780 MHz, 3500-3650 MHz, and 4200-4220 MHz, 4380-4400 MHz bands. http://www.ntia.doc.gov/reports/2010/FastTrackEvaluation_11152010.pdf, 15 Nov 2010
- RTCA, UAS airspace integration architecture description, RTCA SC-203 (2011)
- W. Stallings, *Network Security Essential: Applications and Standards* (Prentice Hall, Upper Saddle River, 2000)
- R. Tandra, S.M. Mishra, A. Sahai, What is a spectrum hole and what does it take to recognize one? Proc. IEEE **97**(5), 824-848 (2009)
- P. van Blyenburgh, *Unmanned Aircraft Systems – The Global Perspective 2009/2010* (Blyenburgh & Co, Paris, France, 2009)
- M.B.H. Weiss, S. Delaere, W.H. Lehr, Sensing as a service: an exploration into the practical implementations of DSA. Proc. IEEE DySPAN Singapore, 6-9 Apr 2010

Section VIII

UAV Architectures

***George J. Vachtsevanos and
Kimon P. Valavanis***

Kimón P. Valavanis and George J. Vachtsevanos

Complex engineered systems, like UAVs, combine a diverse grouping of hardware and software technologies for their performance and effectiveness. They must incorporate in their design and operation sophisticated hybrid architectures that integrate in a rather robust way and meet stringiest performance requirements. UAV Architectures presents and discusses UAV system concepts migrating from the individual UAV component technologies towards a “*system of systems*” view addressing and answering the question “*how do we put it all together?*”

► **HDRC3: A Distributed Hybrid Deliberative/Reactive Architecture for Unmanned Aircraft Systems** by Doherty, Kvarnstrom, Wzorek, Rudol, Heintz, and Conte presents a distributed UAS architecture, also extendable for use with multi-platform systems, providing full integration of both low and high autonomies. It covers the full spectrum of functionalities required for operation in missions requiring high autonomy. Specific interfaces and languages are introduced, which provide seamless transition between deliberative and reactive capability and reactive and control capability. Hierarchical Concurrent State Machines are introduced as a real-time mechanism for specifying and executing low-level reactive control. Task Specification Trees are introduced as both a declarative and procedural mechanism for specification of high-level tasks. Task planners and motion planners are described which are tightly integrated into the architecture. Generic middleware capability for specifying data and knowledge flow within the architecture based

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

on a stream abstraction is also described. Emphasis is placed on the robust integration and interaction between these diverse functionalities using a principled architectural framework. The architecture has been empirically tested in several complex missions, some of which are also described in this chapter.

► **Classification of Multi-UAV Architectures** by Maza, Ollero, Casado, and Scarlatti presents a classification of different schemes for multiple UAV cooperation considering the coupling between the vehicles and the type of cooperation. It also considers UAVs that are networked with other elements in the environment to support their navigation and, in general, their operation. The chapter provides a theoretical framework but also emphasizes practical field demonstrations involving aerial vehicles.

► **Operator Interaction with Centralized Versus Decentralized UAV Architectures** by Cummings starts from the fact that there are major attempts to streamline UAV operations and reduce staffing in order to invert the current many-to-one ratio of operators to vehicles. It then introduces architectural schemes in the context of optimizing operator workload and efficiency. In centralized multiple UAV architectures, a single operator, who requires significant cognitive resources, interacts with and oversees every UAV in the network. In decentralized multiple UAV architectures an operator interacts with an automated mission and payload manager, which coordinates a set of tasks for a group of highly autonomous vehicles. While a single operator can maintain effective control of a relatively small network of centralized UAVs, decentralized architectures are more scalable, particularly in terms of operator workload, and more robust to single points of failure. However, in terms of operator workload, the ultimate success of either a centralized or decentralized UAV architecture is not how many vehicles are in the network per se but rather how many tasks the group of vehicles generates for the operator and how much autonomy is onboard these vehicles. Task-based control of UAV architectures with higher degrees of autonomy (i.e., decentralized networks) can mitigate cognitive overload and reduce workload. Mutually exclusive boundaries for humans and computers in multiple UAV systems should not be the goal of designers for either centralized or decentralized architectures, but rather more effort needs to be spent in defining mutually supportive roles such that humans and computers complement one another.

Collectively, the reader is provided with information on single and multiple UAV architectural systems, hierarchical and hybrid architectures, centralized and decentralized ones, as well as architectures that optimize operator performance.

HDRC3: A Distributed Hybrid Deliberative/Reactive Architecture for Unmanned Aircraft Systems

37

Patrick Doherty, Jonas Kvarnström, Mariusz Wzorek, Piotr Rudol,
Fredrik Heintz, and Gianpaolo Conte

Contents

| | | |
|--------|--|-----|
| 37.1 | Introduction | 850 |
| 37.1.1 | Overview of the HDRC3 Architecture | 853 |
| 37.1.2 | Roadmap to the Chapter | 857 |
| 37.2 | Unmanned Aircraft Platforms | 859 |
| 37.2.1 | The RMAX Helicopter Platform | 859 |
| 37.2.2 | Computational Aspects | 860 |
| 37.2.3 | Communications, Sensors, and Other Hardware | 862 |
| 37.3 | The Control Layer | 863 |
| 37.3.1 | Real-Time Aspects | 863 |
| 37.3.2 | The Control Kernel | 863 |
| 37.3.3 | Path-Following Control Mode | 865 |
| 37.3.4 | Vision-Based Landing Mode | 868 |
| 37.4 | The Reactive Layer | 870 |
| 37.4.1 | Hierarchical Concurrent State Machines | 871 |
| 37.4.2 | Interfacing Between the Control Kernel and Higher Layers | 884 |
| 37.4.3 | Task Specification Trees | 889 |
| 37.5 | The Navigation Subsystem | 893 |
| 37.5.1 | Path and Motion Planning | 896 |
| 37.5.2 | HCSMs for Path Execution | 900 |
| 37.6 | DyKnow: Stream-Based Reasoning Middleware | 903 |
| 37.6.1 | DyKnow | 903 |
| 37.6.2 | Streams and Policies | 905 |
| 37.6.3 | Objects, Features, and Fluent Streams | 905 |
| 37.6.4 | State Generation | 906 |
| 37.6.5 | Semantic Integration of Symbolic Reasoning | 906 |
| 37.6.6 | ROS-Based Implementation | 908 |
| 37.6.7 | A Traffic Monitoring Example | 909 |

P. Doherty (✉) • J. Kvarnström • M. Wzorek • P. Rudol • F. Heintz • G. Conte
Department of Computer and Information Science, Linköping University, Linköping, Sweden
e-mail: patrick.doherty@liu.se; jonas.kvarnstrom@liu.se; mariusz.wzorek@liu.se;
piotr.rudol@liu.se; fredrik.heintz@liu.se; gianpaolo.conte@liu.se

| | | |
|--------|---|-----|
| 37.7 | The Deliberative Layer | 910 |
| 37.7.1 | Temporal Action Logic | 911 |
| 37.7.2 | Task Planning Using TALplanner | 913 |
| 37.7.3 | Execution Monitoring | 918 |
| 37.7.4 | High-Level Mission Specification | 924 |
| 37.8 | Collaborative Systems | 930 |
| 37.9 | Mission Applications | 933 |
| 37.9.1 | Emergency Services Assistance | 933 |
| 37.9.2 | Map Building Using a Laser Range Finder | 943 |
| 37.10 | Conclusions | 946 |
| | References | 948 |

Abstract

This chapter presents a distributed architecture for unmanned aircraft systems that provides full integration of both low autonomy and high autonomy. The architecture has been instantiated and used in a rotor-based aerial vehicle, but is not limited to use in particular aircraft systems. Various generic functionalities essential to the integration of both low autonomy and high autonomy in a single system are isolated and described. The architecture has also been extended for use with multi-platform systems. The chapter covers the full spectrum of functionalities required for operation in missions requiring high autonomy. A Control Kernel is presented with diverse flight modes integrated with a navigation subsystem. Specific interfaces and languages are introduced which provide seamless transition between deliberative and reactive capability and reactive and control capability. Hierarchical Concurrent State Machines are introduced as a real-time mechanism for specifying and executing low-level reactive control. Task Specification Trees are introduced as both a declarative and procedural mechanism for specification of high-level tasks. Task planners and motion planners are described which are tightly integrated into the architecture. Generic middleware capability for specifying data and knowledge flow within the architecture based on a stream abstraction is also described. The use of temporal logic is prevalent and is used both as a specification language and as an integral part of an execution monitoring mechanism. Emphasis is placed on the robust integration and interaction between these diverse functionalities using a principled architectural framework. The architecture has been empirically tested in several complex missions, some of which are described in the chapter.

37.1 Introduction

Much of the recent research activity with unmanned aircraft systems (UASs) has focused primarily on the air vehicle (AV) itself, together with the avionics and sensor subsystems. Primary focus has been placed on the navigation subsystem together with low-level control combined with motion planners that allow a UAS to operate

with limited autonomy. The Control Kernel implements diverse control modes such as takeoff, landing, flying to waypoints, and hovering (in the case of rotor-based systems). Sensor payloads are then used to gather data after positioning the AV at salient points of interest.

Development of this type of low-level autonomy has been impressive, resulting in many AV systems that with the help of human operators can autonomously execute missions of moderate complexity. Specification of such missions is often based on the manual or semi-manual construction of a waypoint database, where waypoints may be annotated with sensor tasks to be achieved at each of these points. Such a means of specifying missions is often time consuming and also prone to error due to the low level of abstraction used and to the lack of automation in generating such plans. Additionally, one lacks techniques for the automatic verification of the correctness of a mission.

Although these capabilities provide the basic functionality for autonomous AVs, if one is interested in increasing the complexity of the missions executed and the usefulness of UASs, much more is required. The collection of functionalities and capabilities required to automate both the process of specifying and generating complex missions, instantiating their execution in the AV, monitoring the execution, and repairing mission plans when things go wrong commonly goes under the umbrella term “high autonomy.” Systems with high autonomy require additional architectural support beyond what one commonly uses to support the low-level autonomy. Furthermore, one has to ensure that each of the architectural components that support both low and high autonomy is fully integrated in the resulting system and provides the proper level of quality of service for the relevant tasks.

This chapter describes a distributed software architecture that fully integrates functionality for both low and high autonomy. The architecture has been developed for a number of years and has been both tested and deployed on a number of rotor-based AV systems. The purpose of this chapter is not only to describe the instantiation of the architecture on a particular AV system but also to isolate and describe various generic functionalities essential to the integration of low and high autonomy. These functionalities and their architectural support can be used on any robotic system, be it an aerial, ground, or underwater system. Consequently, this architectural framework should be of interest to anyone developing autonomous systems.

Let us begin by providing a motivating scenario for the use of UASs. On December 26, 2004, a devastating earthquake of high magnitude occurred off the west coast of Sumatra. This resulted in a tsunami which hit the coasts of India, Sri Lanka, Thailand, Indonesia, and many other islands. Both the earthquake and the tsunami caused great devastation. Initially there was a great deal of confusion and chaos in setting into motion rescue operations in such wide geographical areas. The problem was exacerbated by shortage of manpower, supplies, and machinery. The highest priorities in the initial stages of the disaster were searching for survivors in many isolated areas where road systems had become inaccessible and providing relief in the form of delivery of food, water, and medical supplies.

Assume that for a particular geographical area, one had a shortage of trained helicopter and fixed-wing pilots. Assume also that one did have access to a fleet of autonomous unmanned helicopter systems with ground operation facilities and the ability to both scan for salient entities such as injured humans and deliver supplies and resources such as relief packages containing food, water, and medical supplies. These systems could then provide essential assistance to emergency rescue services.

Participating UASs would require functionalities associated with both low and high autonomy operating in tight integration with each other. Additionally, this scenario would involve several UASs, so additional support would be required for collaboration between ground operation and AVs in addition to cooperation between the AVs themselves.

The complex real-world deployment outlined above would have two phases. In the first, one would require the use of as many AVs as were in the area to scan a specific region for injured and to generate a saliency map that could then be used by emergency services or other UASs to provide relief assistance in the form of medical and food supplies. In the second phase, given a particular saliency map, one would have to determine efficient ways to deliver the appropriate supplies to the different regions in an optimal manner. In essence, there are two high-level goals to be achieved:

1. Scan a specific region with available AVs to generate a saliency map consisting of geolocations of injured civilians.
 2. Put together a logistics plan based on the saliency map and available AVs to deliver medical and food supplies to appropriate locations within this region.
- Ideally, one would like to specify a mission at this level of abstraction and have the UASs generate the appropriate plan or plans for the specific AVs involved based on their capabilities, resources, and availability.

This in itself is a highly complex endeavor based on the use of sophisticated automated distributed planning techniques for teams of AVs and human resources in a mixed-initiative context. An essential aspect of the higher-level or high-autonomy functionalities is the ability to clearly and succinctly specify and generate missions to be executed and to coordinate their execution in a distributed manner.

The execution of the tasks associated with such missions in the individual AVs is just as complex. An essential aspect of the lower-level or low-autonomy functionalities involves different forms of compilation and execution mechanisms in (soft) real time that ultimately result in the coordination and use of real-time continuous control modes associated with the individual AVs. Most importantly, the tight integration of processes at these different levels of abstraction is fundamental to the robust operation of AVs in missions which require such high degrees of autonomy.

Later in the chapter, additional detail will be provided as to how the architecture supports the deployment of AVs in missions of this complexity. Before this, there will be an overview of the architecture and the functionalities it includes and a roadmap based on this overview. The remainder of the chapter will provide more detailed descriptions of these functionalities and their integration with each other.

37.1.1 Overview of the HDRC3 Architecture

In recent years, there has been a consolidation of thought regarding essential conceptual components required in software architectures for mobile robots, although there is wide variation in the way these components are implemented in specific robotic systems. One of the clearest presentations of this consolidation of thought is described in Gat (1997). In this article, he traces the history and development of robotic architectures in artificial intelligence and the transition from sense-plan-act architectures to what are now generically known as three-layered architectures. In his own words,

The three-layer architecture arises from the empirical observation that effective algorithms for controlling mobile robots tend to fall into three distinct categories: (1) reactive control algorithms which map sensors directly onto actuators with little or no internal state; (2) algorithms for governing routine sequences of activity which rely extensively on internal state but perform no search; and (3) time-consuming (relative to rate of change of the environment) search-based algorithms such as planners. (Gat 1997, p. 209)

The algorithms associated with each of these categories have distinct temporal latencies in the decision cycles: (1) millisecond range, (2) seconds range, and (3) seconds or even minutes range. Each of these computational abstractions requires interfaces to each other which in essence implement compilational transitions between the layers.

The Hybrid Deliberative/Reactive (HDRC3) architecture presented here is based conceptually on the three-layered approach. It is important to state that this is due not only to the theoretical clarity of the approach but also to empirical necessity. In the development of architectures which support high and low autonomy and their integration for use in complex missions, one naturally gravitates towards such architectures. Although conceptually layered, in practice, the HDRC3 architecture is better described as concentric (as depicted in Fig. 37.1a) in the sense that data and control flow are both vertical and horizontal within and across layers and that the processes evoked by the various algorithms are highly concurrent and distributed.

Figure 37.1b provides a conceptual and functional depiction of the HDRC3 architecture. It also provides a visual depiction of the roadmap for the chapter where a bottom-up approach will be used to describe the architectural components. The HDRC3 architecture consists of three functional layers and a number of interfaces between the layers:

Control Layer This layer provides a library of control modes based on the implementation of continuous control laws such as takeoff, trajectory following, and vision-based landing. Algorithms associated with this layer generally have little or no internal state. Additionally, it provides a real-time environment for accessing hardware and sensors, executing control modes, and switching between them.

Reactive Layer This layer coordinates the execution of high-level plans generated by the deliberative layer and implements a set of robotic behaviors such as **fly-to** and **scan-area** that may be viewed as compiled plans or as reactive procedures.

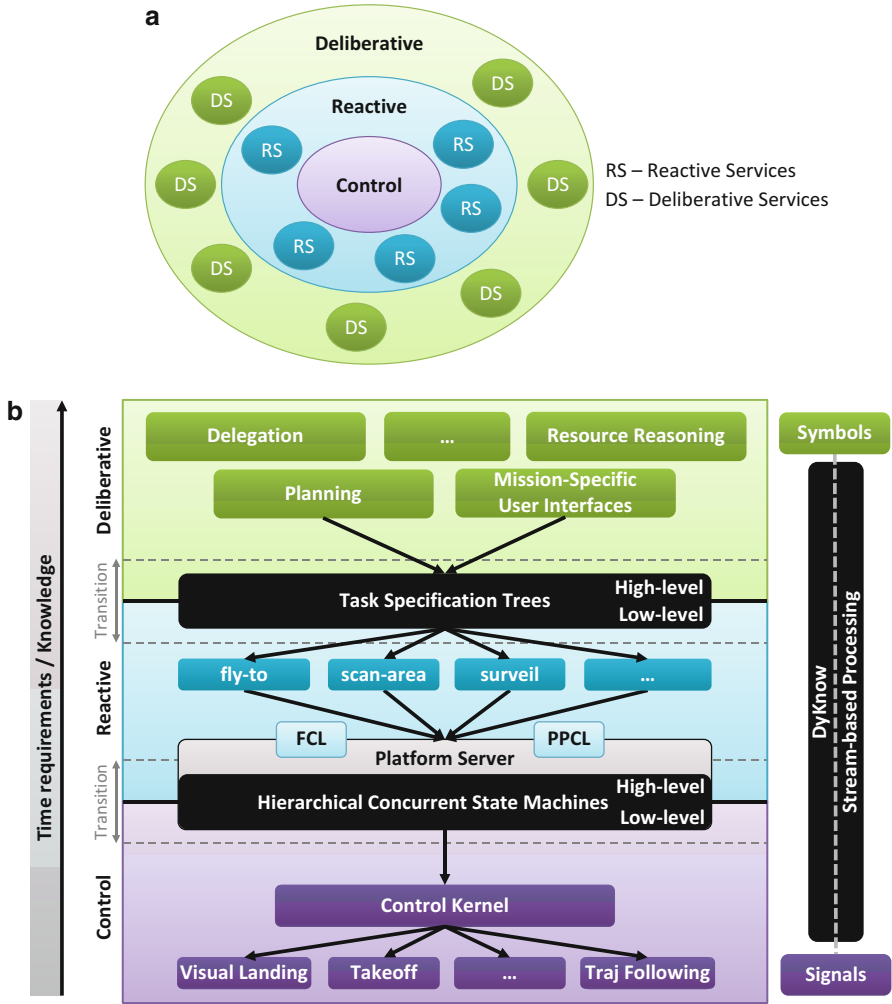


Fig. 37.1 The structure of the Hybrid Deliberative/Reactive (HDRC3) architecture. **(a)** The concentric view. **(b)** The layered view

These are specified as Task Specification Trees (TSTs). The reactive layer also includes a thread-based executor for TSTs. Algorithms associated with this layer do have internal state, but of a postdictive nature generally in terms of previous states of actions.

Deliberative Layer This layer provides a rich set of algorithms commonly associated with deliberation such as automated task planners, motion planners, reasoning and diagnosis mechanisms, and execution monitoring mechanisms.

Algorithms associated with this layer generally have rich internal state of both post- and predictive nature and often include search through large state spaces.

The interfaces used to transition between these layers are central to the success of such architectures. The HDRC3 architecture provides a rich set of interfacing mechanisms:

Interfaces Between the Control Layer and the Reactive Layer A specialized language and implementation is used for interfacing between the control layer and the reactive layer. Hierarchical Concurrent State Machines are used to specify and implement mode switching behaviors at the control layer. Additionally they are used to implement low-level reactive behaviors normally associated with flight control or perception control.

Platform Server Interface Helicopter pilots and ground control personnel often think of controlling unmanned aircraft in terms of a Flight Control Language (FCL) representing various flight modes and a Payload and Perception Control Language (PPCL) representing various modes of perception and sensor control. Because this abstraction is so powerful, an independent server exists which implements this abstraction using two well-defined languages for flight control and perception control, respectively. Any functionality in the system can access the server through these abstract languages.

Interfaces Between the Reactive and Deliberative Layers The notion of robot task specifications and the processes they invoke are central to the achievement of goal-directed behavior. Tasks can be specified in many different ways. HCSMs are one way to specify low-level tasks. At the other end of the spectrum are highly complex tasks which use low-level tasks as primitives. Task Specification Trees (TSTs) are used as a means of not only specifying such high-level tasks declaratively, but also providing procedural correlates executable in the system. TSTs therefore provide both a declarative and procedural means of transitioning between the deliberative and reactive layers. It is often the case, for instance, that TSTs are used to sequentialize commands associated with the FCL and PPCL. Additionally, the output of an automated planner may be viewed as a TST.

37.1.1.1 Middleware Infrastructure

The functionality in the reactive and deliberative layers of the HDRC3 architecture should be viewed as sets of loosely coupled distributed processes that are highly concurrent and often require asynchronous communication with each other. These processes run on multiple onboard (or ground-based) computers, communicating through service calls and transmitting information through a set of distributed communication channels. This requires a rich middleware infrastructure to provide the communication channels between these processes and the ability of each to use other processes when required. An example would be interaction between task planners, path planners, and execution monitors.

In early iterations of the HDRC3 architecture, the underlying middleware infrastructure was based on the use of CORBA (Common Object Request Broker

Architecture). The architecture has recently been transitioned to using ROS, the Robot Operating System. This is a convenient open-source framework for robot software development that provides client libraries and tools for several programming languages (Quigley et al. 2009).

Software written for ROS is organized into *packages* which contain nodes, libraries, and configurations. *Nodes* represent computational processes in the system and are written using the client libraries. For example, many capabilities such as task planning and motion planning are realized as separate ROS nodes. These nodes communicate by passing structured messages on *topics* which can be seen as named buses to which nodes can subscribe and by using request/reply communication through *services*.

Although emphasis in this chapter is placed on a single platform using the HDRC3 architecture, it is also set up for collaborative missions with multiple AV systems. All agents in such a collaborative system currently use a common ROS Master, a standard ROS functionality providing registration and lookup services. For disconnected operation, a federated approach to the ROS Master is used.

Figure 37.2 gives an overview of some of the essential processes and ROS topics that are present in the HDRC3 architecture. These include functionality for use of individual platforms in collaborative scenarios. The functionality associated with the control layer is encapsulated in the Platform Server and accessed by the two languages FCL and PPCL mentioned earlier.

Black arrows indicate calls between processes. Each gray rectangle contains processes and topics that are explicitly associated with a particular agent. Inside this, each rounded rectangle represents a distinct functionality that is currently implemented as one or more related ROS services provided by one or more ROS nodes. Given the fundamental structure of ROS, functionality can easily be moved between nodes without affecting the remainder of the system. Therefore, an exact specification at the node level is not relevant at this level of abstraction.

There is one special functionality associated with the HDRC3 architecture which can be viewed as a shared middleware and interfacing mechanism supporting the creation and management of data flow in the architecture. This middleware is called DyKnow and it is based on the abstraction of streams of data.

One of the highly problematic and open research issues in AI and robotics is the large discrepancy between the quantitative and large amount of raw sensor data input to a robotic system and the requirement for sparse, well-structured, and grounded qualitative data used by the deliberative functionalities in robotic architectures. This is often called the sense-reasoning gap.

DyKnow is a middleware functionality that has been developed in the HDRC3 architecture which provides a principled means of closing this gap both theoretically and pragmatically. DyKnow provides abstractions for the construction, reasoning, processing, and abstraction of data streams within the architecture itself. This middleware is leveraged throughout the architecture and used by many functionalities to great benefit. It is used in a variety of ways at any layer in the HDRC3 architecture and is integrated with ROS.

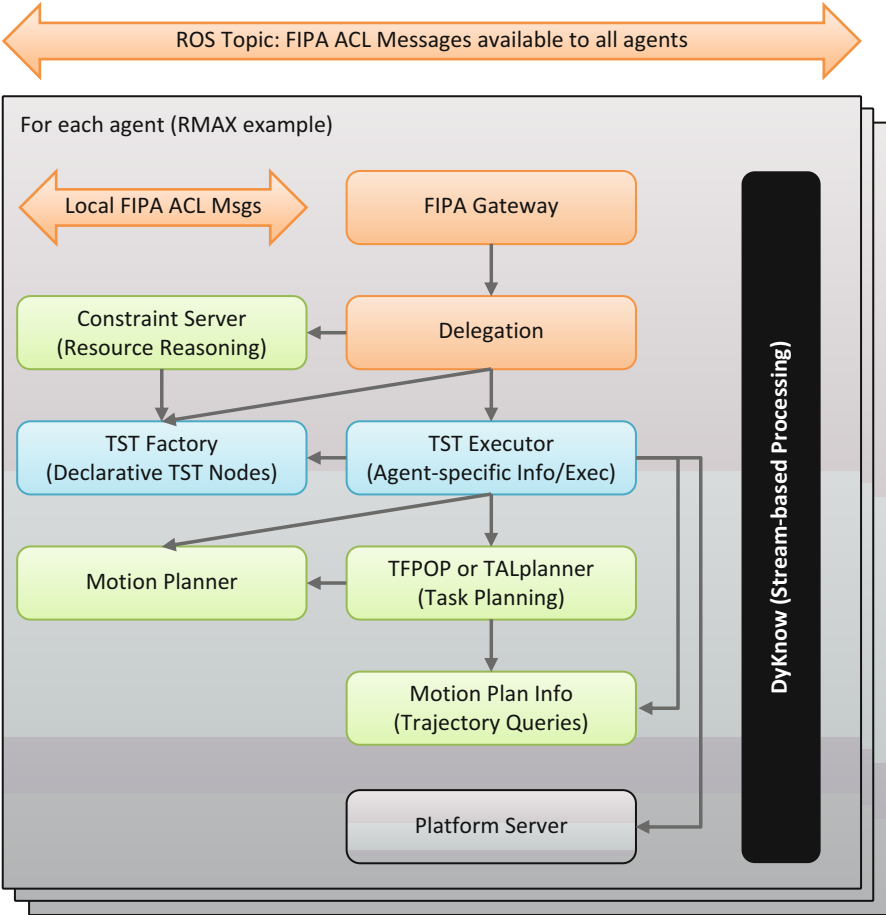


Fig. 37.2 Overview of the ROS-based implementation

37.1.2 Roadmap to the Chapter

Both Figs. 37.1 and 37.2 provide concise visual depictions of the HDRC3 architecture and its essential functionalities. This framework will be used to structure and provide a roadmap for the remainder of the chapter.

Section 37.2 describes the UAS Tech Lab (UASTL) RMAX helicopter system which includes a distributed hardware architecture and a suite of sensors integrated with the base Yamaha RMAX helicopter system.

Section 37.3 describes the Control Layer. This includes a description of the Control Kernel itself with its primary flight modes (Sect. 37.3.2), in addition to two complex compound flight modes: a path-following control mode (Sect. 37.3.3) and a vision based landing mode (Sect. 37.3.4).

Section 37.4 describes the reactive layer. This includes a description of Hierarchical Concurrent State Machines and a real-time HCSM interpreter (Sect. 37.4.1). This is a generic capability that requires instantiation and use in terms of specific HCSMs. Section 37.4.1.1 describes these specific HCSMs used in the UASTL RMAX system. The higher-level ROS-based system shown in Fig. 37.2 interfaces to the real-time control functionalities (Sect. 37.3.2) through the *Platform Server*. This is described in Sect. 37.4.2. Complex tasks and missions are concretely represented as Task Specification Trees using elementary actions such as flying together with a number of task composition constructs such as sequences, concurrent execution, and loops (Sect. 37.4.3). Nodes in such trees, which are general and declarative specifications of tasks to perform, are both created by and stored in a *TST Factory*. The *TST executor* functionality is responsible for generating platform-specific procedural *executors* for any node type that a particular agent supports. Platform-specific constraints on how a certain task can be performed, such as those related to the flight envelope of the RMAX, are also specified here.

Section 37.5 describes the navigation subsystem. This subsystem includes the path and motion planning algorithms (Sect. 37.5.1) in addition to the specific HCSMs used to coordinate these processes (Sect. 37.5.2).

Section 37.6 describes DyKnow, the stream-based processing middleware framework used for managing data and knowledge flow within the HDRC3 architecture. This functionality is used by many other functionalities within the architecture. Connections to DyKnow are omitted in Fig. 37.2 in order to avoid clutter, as information processed by DyKnow can be used by any deliberative or reactive functionality. A traffic monitoring example, where DyKnow plays an essential role, is described in Sect. 37.6.7.

Section 37.7 describes many of the high-level deliberative functionalities in the HDRC3 architecture that can be used by other functionalities in the architecture. One open research issue of great importance is the requirement of formally verifying the behavior of complex deliberative functions which are often dependent on rich world models and also highly nondeterministic in nature. The HDRC3 architecture uses temporal logics not only for specification but for online monitoring purposes. Section 37.7.1 describes Temporal Action Logic (TAL) which is used for formally specifying task plans and Task Specification Trees. Section 37.7.2 describes a *task planner* called TALplanner that is used in the HDRC3 architecture and is highly integrated with motion planners and execution monitoring mechanisms. One of the responsibilities of the task planner is to expand *goal nodes* in a Task Specification Tree into detailed task specifications, which is why it can be called from the TST executor shown in Fig. 37.2. The task planner can also call the motion planner and Motion Plan Info functionalities in order to estimate whether a given flight action is feasible as part of a plan being generated, and if so, at what cost. Section 37.7.3 describes a very sophisticated and efficient execution monitoring functionality based on the use of model-checking temporal logical formulas in real time through use of a progression algorithm. Section 37.7.4 describes an extension to Temporal

Action Logic that allows one to formally specify complex missions. These missions can be compiled into Task Specification Trees and are also useful in specifying collaborative missions.

Section 37.8 briefly discusses collaborative systems consisting of multiple platforms. In a collaborative setting, *delegation* is the key to assigning the responsibility for specific aspects of a mission to specific agent-based unmanned systems. Delegation requests are specified as speech acts in the FIPA Agent Communication Language (ACL) (Foundation for Intelligent Physical Agents 2002) and are sent between agents through a set of communication channels, currently implemented using ROS topics. A *gateway* functionality acts as a clearinghouse for all interagent communication based on such speech acts. Determining whether an agent can accept a delegation request also requires reasoning about whether and how the agent can satisfy the temporal and resource-related constraints associated with a task. This is implemented through a resource reasoning capability that includes a *constraint server*. These particular functionalities are shown in Fig. 37.2.

Section 37.9 provides examples of several highly complex mission scenarios that have been used to empirically test the HDRC3 architecture, its various functionalities, and the tight integration required among the functionalities. UASTL RMAXs have been deployed and used in these scenarios and variations of them on numerous occasions to not only refine the HDRC3 architecture but also to ensure repeatability of missions and robustness of the architecture in real-world environments.

Section 37.10 summarizes and discusses the generic nature of the architecture and many of its functionalities.

37.2 Unmanned Aircraft Platforms

The HDRC3 architecture and variations of it have been used on a number of different research platforms, in particular in the context of research with heterogeneous collaborative systems (Doherty et al. 2013; <http://www.ida.liu.se/divisions/aiics/aiicssite/>). For example, a lightweight version of the architecture has been ported to the LinkQuad, a micro-aerial vehicle platform developed within the UAS Tech Lab (<http://www.uastech.com>). However, this chapter focuses solely on the use of the HDRC3 architecture on a Yamaha RMAX system which has been extended for autonomous use. This platform is called the UAS Tech Lab (UASTL) RMAX (Doherty et al. 2004; Wzorek et al. 2006a; Rudol and Doherty 2008; Conte and Doherty 2009).

37.2.1 The RMAX Helicopter Platform

The UASTL RMAX platform is a modified Yamaha RMAX helicopter (Fig. 37.3). The Yamaha RMAX helicopter is a radio-operated commercial system marketed in



Fig. 37.3 The UASTL RMAX helicopter

Japan by Yamaha Motor Co. Ltd. Two of these were acquired and used as a basis for developing fully autonomous systems. The HDRC3 architecture is integrated on both systems.

The RMAX is approximately 2.7×0.7 m with a total length of 3.6 m including the rotor. It uses a 21 HP two-stroke engine and has an empty weight of 61 kg and a maximum takeoff weight of 95 kg.

37.2.2 Computational Aspects

The computer systems used onboard the UASTL RMAX need to work reliably in harsh environmental conditions characterized by high vibrations, limited space, and a high range of operating temperatures. The PC104 is an industrial-grade embedded computer standard that is especially suited for such applications. The onboard system developed for the UASTL RMAX contains three such PC104 embedded computers. These embedded computers physically reside in a specially designed box which is integrated with the RMAX and is shown in Fig. 37.3. Figure 37.4 provides a hardware schematic of the integrated system.

- The Primary Flight Computer (PFC, 1.06 GHz Intel Celeron M) handles core flight functionalities and control modes. This computer is directly connected to a GPS receiver and several additional sensors including a barometric

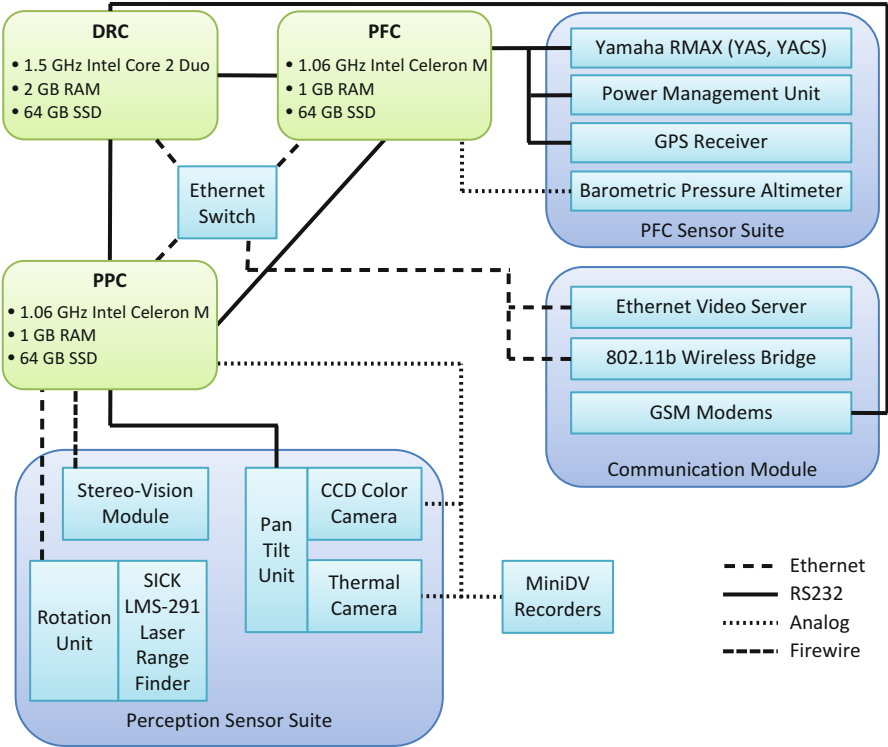


Fig. 37.4 Onboard hardware schematic

altitude sensor. The PFC is also connected to the UASTL RMAX helicopter through the standard Yamaha Attitude Sensor (YAS) and Yamaha Attitude Control System (YACS) interfaces.

- The Deliberative/Reactive Computer (DRC, 1.5 GHz Intel Core 2 Duo) executes all high-level autonomous functionalities such as path planning (Sect. 37.5.1), stream-based reasoning (Sect. 37.6), and mission planning (Sect. 37.7.2).
- The Primary Perception Computer (PPC, 1.06 GHz Intel Celeron M) runs software related to the use of cameras, range finders, and similar equipment. This includes image processing and sensor fusion.

Due to the high degree of vibration onboard the UASTL RMAX, in its current configuration it is not possible to use standard hard drives for storage. Instead all computers use Solid State Disk (SSD) technology.

Network communication between the onboard computers is realized with serial lines (RS232C) as well as Ethernet. The serial lines are used to implement a robust real-time communication channel between each of the onboard computers.

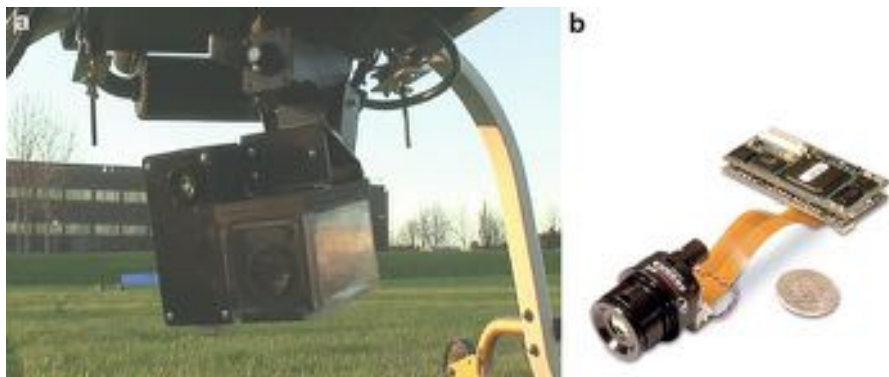


Fig. 37.5 Cameras used on the UASTL RMAX. (a) CCD Block Camera on PTU. (b) Infrared Camera

37.2.3 Communications, Sensors, and Other Hardware

A wireless Ethernet bridge as well as a GSM modem is available for communication with ground control stations. GSM provides a mature, commercially available communication infrastructure that permits the operation of AVs at large distances, out of sight from the ground operator. It also provides a redundant alternative in the case where other communication frequencies are jammed. In order to test the feasibility of GSM technology for interfacing purposes, a multimodal graphical user interface has been designed, constructed, and implemented on a mobile telephone (Wzorek et al. 2006b). This has been used as a mobile ground operator control system where mission plans can be specified, uploaded, and monitored through the mobile telephone.

A camera platform is suspended under the AV fuselage, vibration-isolated by a system of springs. The platform consists of a Sony FCB-780P CCD block camera (Fig. 37.5a) and a ThermalEye-3600AS miniature infrared camera (Fig. 37.5b) mounted rigidly on a pan-tilt unit (PTU). Section 37.9.1.1 presents a method for using both cameras in an algorithm for finding human bodies and building saliency maps. The video footage from both cameras is recorded at full frame rate by two miniDV recorders to allow processing after a mission.

A modified SICK LMS-291 laser range finder has also been integrated with the UASTL RMAX using a rotation mechanism developed in-house. It is mounted on the front as shown in Fig. 37.3. This system provides the ability to generate high-accuracy 3D models of the environment in which the aircraft is operating, without the need for reflectors, markers or scene illumination (Sect. 37.9.2).

A software-controlled power management unit provides information about voltage levels and current consumption. It also allows for remote power control where all devices can be switched on or off through software. The unit controls all three PC104 computers, all sensors, and the wireless Ethernet bridge.

The HDRC3 architecture is distributed on this hardware platform. The control layer of this architecture, which in many ways operates close to the hardware, will now be described.

37.3 The Control Layer

The control layer implements basic continuous control modes used by the UASTL RMAX. These control modes are implemented using continuous control laws as a basis. Already at this level, the algorithms used are quite complex. For instance, two essential modes described in this section, path-following (Sect. 37.3.3) and vision-based landing (Sect. 37.3.4), require tight integration with the architecture and involve discrete event control capability (Sect. 37.4.1). This section describes both these modes in addition to the Control Kernel (Sect. 37.3.2) and its real-time demands (Sect. 37.3.1).

37.3.1 Real-Time Aspects

Certain functionality, such as control laws in the Control Kernel, requires support for real-time processing with latency guarantees. Since the current implementation of the HDRC3 architecture runs on a Linux-based operating system, the Real-Time Application Interface (RTAI, Mantegazza et al. 2000) is used for this purpose.

RTAI is a hard real-time extension to the standard Linux kernel and provides industrial-grade real-time operating system functionality. RTAI allows for the creation of a kernel module that takes full control over the CPU and can suspend itself in order to let user-space applications run. Such a module is used to implement real-time functionalities in the HDRC3 architecture. The standard Linux distribution running on the same CPU is viewed as a separate task with lower (non-real-time) priority, running preemptively so that it can be interrupted by real-time processes at any time. Real-time and non-real-time processes can then communicate through shared memory.

All computers in the UASTL RMAX platform (Fig. 37.4) have both hard and soft real-time components, but the processor time is assigned to them in different proportions. On one extreme, the PFC runs mostly hard real-time tasks with only a minimal number of non-real-time user-space applications (e.g., an SSH daemon for remote login). On the other extreme, the DRC uses the real-time part only for device drivers and real-time communication, while the majority of its time is spent on running deliberative services.

37.3.2 The Control Kernel

The Control Kernel (CK) encapsulates all core control functionalities, including a set of continuous control laws implementing the basic flight and payload control modes as well as control over the perception capabilities that can be used by the

reactive and deliberative layers. It is also responsible for switching between control modes through the use of Hierarchical Concurrent State Machines (Sect. 37.4.1).

The kernel is distributed on two onboard computers (PFC and PPC) and also coordinates real-time communication between these computers as well as between CKs of other robotic systems when collaborative missions are involved. However, the kernel itself is self-contained, and only the part running on the PFC computer is necessary for maintaining flight capabilities. This enhances the safety of the operation of the UASTL RMAX platform.

The following flight control modes have been implemented on the UASTL RMAX platform:

- *Hovering*: Keep the helicopter in the desired position, heading, and altitude given current sensor readings. The mode is implemented using a set of PID loops.
- *Takeoff*: Bring the helicopter from the ground to a specified altitude.
- *Path-following*: Fly along a specified segmented path (Sect. 37.3.3).
- *Vision-based landing*: Land the AV on an artificial pattern. This mode does not require a GPS signal, instead it relies on a more accurate relative state estimation performed onboard (Sect. 37.3.4).
- *Reactive car following*: Maintain a specified distance to a moving car. The car position estimation is performed onboard the AV and is based on color tracking using camera images.

Additionally, a set of control modes for payloads has been developed:

- *Camera pan-tilt absolute/relative control*: Maintain the absolute/relative angles of the pan-tilt mechanism.
- *Camera pan-tilt visual servoing control*: Keep a tracked object in the center of the camera image. The position of the object is provided by image processing functionalities such as vision-based color object tracking.
- *Camera pan-tilt look at GPS coordinate control*: Keep a geographical location in the center of the camera image.
- *Camera (color, IR) parameter control*: Actively control camera parameters such as shutter, zoom, and iris. Among other things, this allows for adjusting to the current lighting conditions.
- *Laser range finder rotational mechanism control*: Actively control parameters of the rotational mechanism, such as rotational speed. Among other things, this allows for adjusting the trade-off between speed and accuracy when building a 3D map.
- *Laser range finder parameter control*: Actively control parameters of the laser range finder, such as its measurement distance resolution and angular resolution. Among other things, this allows for adjusting the trade-off between speed and accuracy when building a 3D map.

Perception functionalities accessible in the Control Kernel include:

- *Vision-based state estimation*: Provides a very accurate pose estimate in relation to a specifically designed pattern. This functionality is used for the autonomous vision-based landing mode.
- *Vision-based object tracking*: Provides an estimate of the relative position to a moving object based on the color and/or thermal image streams. This is used in the reactive car-following control mode.

- *Vision-based human body identification*: Identifies human body positions in a stream of color and thermal images (Sect. 37.9.1.1).
 - *Laser range finder 3D map building*: Provides a 3D elevation map of the environment based on laser range finder data. The map can be used for various applications such as navigation where it is used as input to a path planner (Sect. 37.9.2).
 - *Localization based on reference images*: Provides an absolute position estimate in GPS-denied environments based on geo-referenced imagery (Conte 2009).
- Two complex flight control modes will now be considered in more detail.

37.3.3 Path-Following Control Mode

One of the control modes executed and coordinated by the Control Kernel is the path-following control mode (PFCM (Conte et al. 2004; Conte 2009), bottom part of Fig. 37.17 on p. 33) which executes paths consisting of a set of segments. Figure 37.7 presents an example path consisting of three segments.

Given a specific robotic platform, a classical problem in control theory is to find a trajectory compatible with its kinematic and dynamic constraints. A trajectory is the evolution of the state of a robot. The robot state is a vector composed of a set of time-dependent variables describing its kinematic and dynamic status. In this specific case, the most relevant components of the state vector are the three-dimensional position, velocity and acceleration, the attitude angles (pitch, roll, yaw), and the attitude rates. In general, the problem of finding a trajectory compatible with the platform dynamic constraints is a very difficult problem. In the context of a robotic helicopter flying in a cluttered environment, the problem becomes even harder since the platform dynamics is unstable. In addition, the flight path must not collide with the obstacles in the environment. The methodology used here to deal with this complexity is to decompose the problem as follows:

- *Decoupling the path-planning problem from the platform dynamics*, by first searching for a collisionfree path and then adapting this path to the dynamics of the AV platform. This yields a dramatic dimensionality reduction in the search space. In the HDRC3 architecture, the task of finding a collision-free path is solved by a path planner.
- *Dividing the control problem into fast and slow dynamics*. This is usually achieved using an *inner* feedback loop which has the task of stabilizing the attitude dynamics. The inner loop is usually implemented at a relatively high frequency (around 200 Hz). An *outer* feedback loop is then used to stabilize the platform position and velocity at a lower frequency (around 20 Hz). Dividing the problem in this way facilitates analyzing and solving the platform control task. In other words, the inner loop takes care of stabilizing and controlling the rotational dynamics, while the outer loop takes care of stabilizing and controlling the translational dynamics. They are implemented at different frequencies because the platform rotational and translational dynamics have different time constants. The rotational dynamics is usually faster than the translational dynamics.

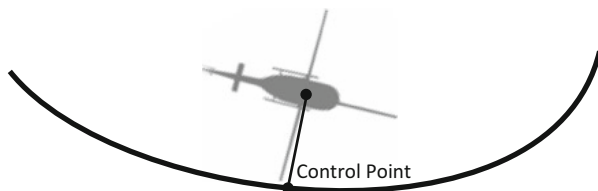


Fig. 37.6 Control point on the reference path

This section deals with the problem of finding a *guidance* algorithm which enables the helicopter to follow a geometric path generated by a path planner. The distinction between path and trajectory must be emphasized here. As previously mentioned, a trajectory is the evolution of the state of the vehicle in the state space. The trajectory is usually time dependent. The path is a geometric description of the flight course and it is time independent. The problem addressed here is referred to in the literature as the *path-following* problem. The approach is suitable when a geometrical description of the path is provided in a parameterized polynomial form. In this case the so-called virtual leader approach (Egerstedt et al. 2001) can be applied.

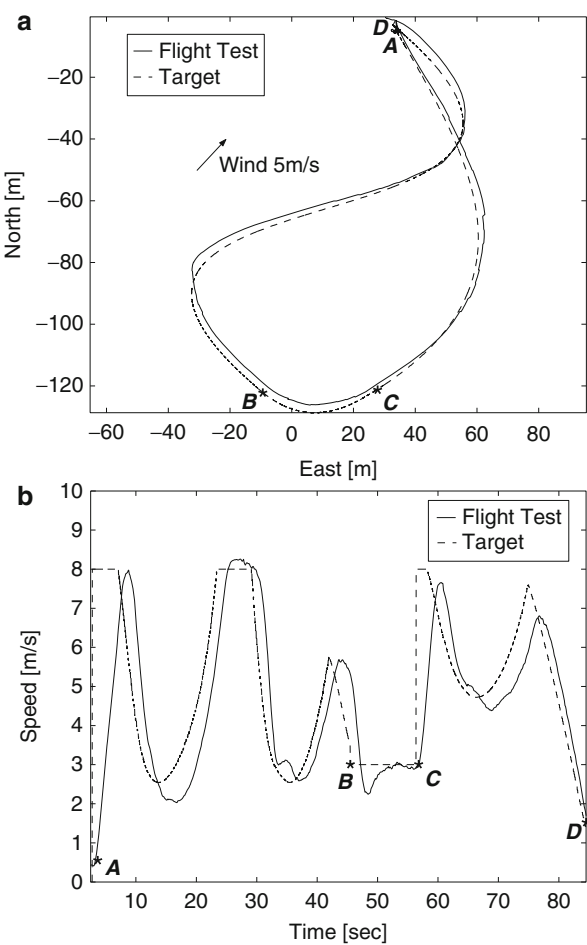
By using the path-following method, the helicopter is forced to fly close to the geometric path with a specified forward speed. In other words, following the path is always prioritized over following other trajectory parameters as, for instance, the desired speed. This is a requirement for robots that, for example, have to follow roads and avoid collisions with buildings. The PFCM method developed is weakly model dependent and computationally efficient.

The approach adopted here uses a guidance algorithm similar to the one described in Egerstedt et al. (2001) which is also known as virtual leader. In this case the motion of the control point on the desired path is governed by a differential equation containing error feedback which gives great robustness to the guidance method. The control point (Fig. 37.6) is basically the closest point on the path relative to the current platform position.

A path is composed of several consecutive segments. Each segment is mathematically described by the equation $\vec{p}(s) = \mathbf{A}s^3 + \mathbf{B}s^2 + \mathbf{C}s + \mathbf{D}$, where \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} are 3D vectors calculated from the boundary conditions of the segment. s is the segment parameter and assumes values between 0 (start of the path) and 1 (end of the path). The control point is found using a feedback method. Details of the algorithm can be found in Conte (2009) and Conte et al. (2004). Once the control point is found, from the path equation it is possible to compute all the geometric parameters which are then used for control purposes. The parameters are the 3D position, 3D path tangent, and 3D path curvature.

Path segments are generated in the deliberative layer of the HDRC3 architecture by the path-planner functionality (Sect. 37.5). The segments in a path are passed sequentially to the PFCM for execution. Together with the path parameters, the path planner provides a desired target velocity to the PFCM. During execution,

Fig. 37.7 Flight test of a multi-segment path. (a) Target and executed path. (b) Velocity profile



the PFCM performs a compatibility check between the desired target velocity and the maximum allowable velocity at that precise point on the path. The maximum allowable velocity profile for a path is calculated using the path curvature R and the maximum helicopter roll angle ϕ_{\max} : $V = \sqrt{|\phi_{\max} g R|}$, where g is the gravity constant. In addition, the path curvature is used as a feed-forward term in the outer control loop in order to reduce the tracking error. Besides the basic control task, the PFCM returns a set of status flags which are used to coordinate the path segment switching mechanism.

A safety braking procedure is activated in case the next segment is not provided by the navigation subsystem (e.g., due to communication failure) before a specific point in time. This timepoint is calculated using the minimum distance necessary to safely stop the helicopter at the end of the current segment.

Figure 37.7 shows a multi-segment path and the relative velocity profile during a real flight test using the UASTL RMAX helicopter. The segments AB, BC, and CD have the desired target velocity set at 8, 3, and 8 m/s, respectively. One can observe how the PFCM adjusts the target speed, slowing down the helicopter in order to meet the compatibility requirements discussed above.

37.3.4 Vision-Based Landing Mode

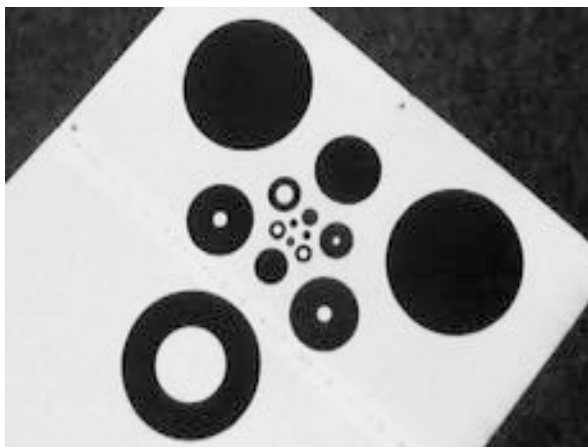
Many autonomous landing systems for AVs are based on GPS and a dedicated close range sensor for accurate altitude measurement (radar altimeter, sonar, infrared, or theodolites). However, in urban environments, buildings and other obstacles disturb the GPS signal and can even cause loss of signal (multipath effects, EM noise due to active emitters). Once the GPS signal is lost, the dead reckoning capability of affordable onboard inertial navigation systems does not allow precision navigation for more than a few seconds.

In contrast, the vision-based landing mode discussed here is self-contained and not jammable (Merz et al. 2004). In the current implementation, it provides a position measurement one order of magnitude more accurate than standard GPS (cm accuracy or better). Additionally, it does not require a complex infrastructure, only a pattern placed on the designated landing area (Fig. 37.8). To improve robustness, readings from an inertial measurement unit are fused with the position sensed by the vision system.

The landing system includes three main components: (a) a vision system that allows robust pose estimation from a suitable distance at a sufficient rate with low latency, using a landing pad of minimal size; (b) an algorithm to fuse vision data with inertial measurements; and (c) a matching control strategy.

Vision System. The vision system consists of a monocular camera mounted on a pan/tilt unit (PTU) and a landing pad (a foldable rectangular plate) with a reference pattern on its surface (Fig. 37.8). The reference pattern is designed specifically for fast recognition, accurate pose estimation for close and distant range, minimum size, and minimal asymmetry. The choice of black circles on a white background allows for fast detection and provides accurate image features. From the projection of three circles lying on the corner points of an equilateral triangle, the pose of an object is uniquely determined assuming all intrinsic camera parameters are known. Circles are projected as ellipses, described by the center point, the semimajor axis, the semiminor axis, and the semimajor axis angle. The pose of the landing pad with respect to the camera coordinate system is estimated by minimizing the reprojection error of the extracted center points and semi-axes of the three ellipses. Five circle triplets of different size are used (radius, 2–32 cm; distance, 8–128 cm) with a common center point to achieve a wide range of possible camera positions. Each triplet is uniquely determined by a combination of differently sized inner circles.

Fig. 37.8 Landing pad with reference pattern seen from the onboard camera



The output of the image processing unit is the camera pose relative to the pattern. The pose parameters are converted to helicopter position and attitude using angles from the PTU and known frame offsets and rotations.

Sensor Fusion. The position and attitude estimates provided by the vision system cannot be fed directly into the controller due to their intrinsic lack of robustness: The field of view can be temporarily occluded (e.g., by the landing gear), and the illumination conditions can change dramatically just by moving a few meters (sun reflections, shades, etc.). On the other hand, vision readings are very accurate, when available.

Hence, a navigation filter based on a Kalman filter (KF) has been developed, fusing highly accurate 3D position estimates from the vision system with inertial data provided by the onboard accelerometers and angular rate gyros. Besides filtering out a large part of the noise and outliers, the filter provides an adequate dead reckoning capability, sufficient to complete the landing even when the vision system is “blind” due to occlusions between the camera and the pattern (landing gears) or due to sun reflections.

Control Strategy. The requirements set on the flight control system during landing are the following:

1. It should be possible to engage the landing mode from any point where the landing pad is visible, meaning approximately within a 20 m radius hemisphere centered on the pattern.
2. Once the landing mode is engaged, the helicopter state should be compatible with the proper functionality of the vision system until touchdown. Thus, during the approach phase the following should be considered: (a) The helicopter's position and attitude should not cause physical occlusion of the visual field; (b) the regions where the accuracy of the vision system is worst should be avoided, if possible; (c) the helicopter velocity and angular rates should not saturate the

pan/tilt unit's capability for compensation – too high angular rates of the visual beam may result in blurred images; and (d) the position of the dominant light source (sun) should be considered, to avoid full reflections.

3. The wind direction has to be taken into account: Tailwind landings should be avoided.
4. The control system should be dimensioned for wind levels up to 10 m/s.
5. The engine should be shut down autonomously once touchdown is detected. Detection should be timely: Early detections cause high touchdown loads and late detections can cause ground resonance.
6. The vertical velocity at touchdown should be of the same order of magnitude as for a manual landing.

Experimental Results. Numerous autonomous landings were conducted from different relative positions to the landing pad within the specified envelope, on grass and snow fields, with different wind and illumination conditions. The vertical velocity at touchdown ranged between 18 and 35 cm/s, corresponding to load factors of about 1.4 g on grass fields. The horizontal velocity was in the order of magnitude of 15 cm/s. The average touchdown point precision was about 42 cm (13 % of the rotor diameter).

This section has isolated and described the essential aspects of the Control Kernel and two complex control modes. In order to put these to use, though, they have to be integrated and interfaced with many different parts of the HDRC3 architecture. These details are described in Sects. 37.4 and 37.5.

37.4 The Reactive Layer

The functionality associated with the reactive layer is an essential part of the HDRC3 architecture since it provides transitional capability between functionality in the deliberative layer and functionality in the control layer. The tasks associated with this layer range from very low-level state machine-based processes which provide discrete control capability for the control layer to much higher-level procedural mechanisms which provide discrete control capability for complex functionalities in the deliberative layer. In fact, the abstractions and language interfaces used for these transitions are what make such architectures so unique and useful. Specification and design of these mechanisms involve more than just engineering skills.

There is an independent generic flavor to the mechanisms required for combining both discrete and continuous control capabilities and managing the processes evoked by task descriptions at many different levels of abstraction with required temporal latencies to guarantee quality of service and robustness.

This section will describe three generic interfacing capabilities associated with the HDRC3 architecture in addition to several task execution mechanisms for tasks defined at different levels of abstraction. Section 37.4.1 considers the specification of complex combinations of state machines using Hierarchical Concurrent State Machines as a specification language. Additionally, an algorithm for executing

HCSMs in real time using an online interpreter is described. Section 37.4.2 describes the two abstract languages FCL and PPCL and a Platform Server which relates these languages to combinations of low-level control modes. This mechanism provides a high-level abstraction for any functionality in the system to access the control layer and combinations of control modes in an efficient and declarative manner. Section 37.4.3 describes Task Specification Trees. TSTs provide both a declarative means of specifying complex tasks and an execution mechanism which allows for distributed execution of TSTs.

37.4.1 Hierarchical Concurrent State Machines

Missions generally involve multiple control modes used in sequence or in some cases in parallel. For example, in building surveillance one would use takeoff, hovering, path-following, camera control, landing, and possibly other control modes. Even if only a sequence of control modes is needed, execution is non-trivial as it requires a smooth handover between the control mode that is terminated and the one that is started, leading to what is often called the *mode switching problem*.

The conditions for being able to switch to a new control mode vary depending on the particular mode transition. For example, in the case of switching between hovering and path-following modes, one must verify that the current helicopter heading is aligned with the path to be flown. If the heading difference is too large, an additional yaw maneuver is necessary before the transition to the path-following mode. Otherwise, the execution of the path could result in a maneuver that potentially leads to a crash.

Solutions to the mode switching problem are often based on Finite State Machines (FSMs) (Harel 1987; Koo et al. 1998), a mathematical abstraction used to create a behavior model of a system. FSMs are composed of a finite number of states, state transitions that can be guarded by conditions, and actions. For example, a condition can depend on the evaluation of a sensor value. The input to a state machine is a sequence of symbols, events, or commands. Standard FSMs such as Moore (1956) and Mealy (1955) machines have been successfully used in the past for modeling systems for various purposes. However, several extensions are necessary to make them useful for modeling complex real-time systems.

One major problem related to FSMs is their flat single-level structure, which provides no easy way to partition a problem into subproblems that can be dealt with separately. Each state has to be modeled explicitly, leading in the worst case to a combinatorial explosion in the number of states. This results in large and unmanageable models even for moderately complex systems.

It would be much easier to define a system gradually with different levels of granularity or with the help of various levels of abstraction. An obvious additional advantage would be the reusability of existing modeled parts of the system in different contexts. A novel modeling and development framework for hybrid control systems called *Hierarchical Concurrent State Machines* (HCSMs, Merz 2004; Wzorek 2011) has been developed. HCSMs are based on a combination of Moore

and Mealy machines with several extensions which include support for hierarchy and concurrency. This allows all low-level components to be efficiently modeled and contributes to a solution to the mode switching problem in the HDRC3 architecture. It permits all *functional units* of the control system to be coordinated ranging from the lowest level (such as device drivers), through the use of control laws (such as hovering and dynamic path-following) and communication, to high-level deliberative components. The system has proven to be robust, reliable, and easy to extend and has been used in a number of autonomous missions during a period of several years.

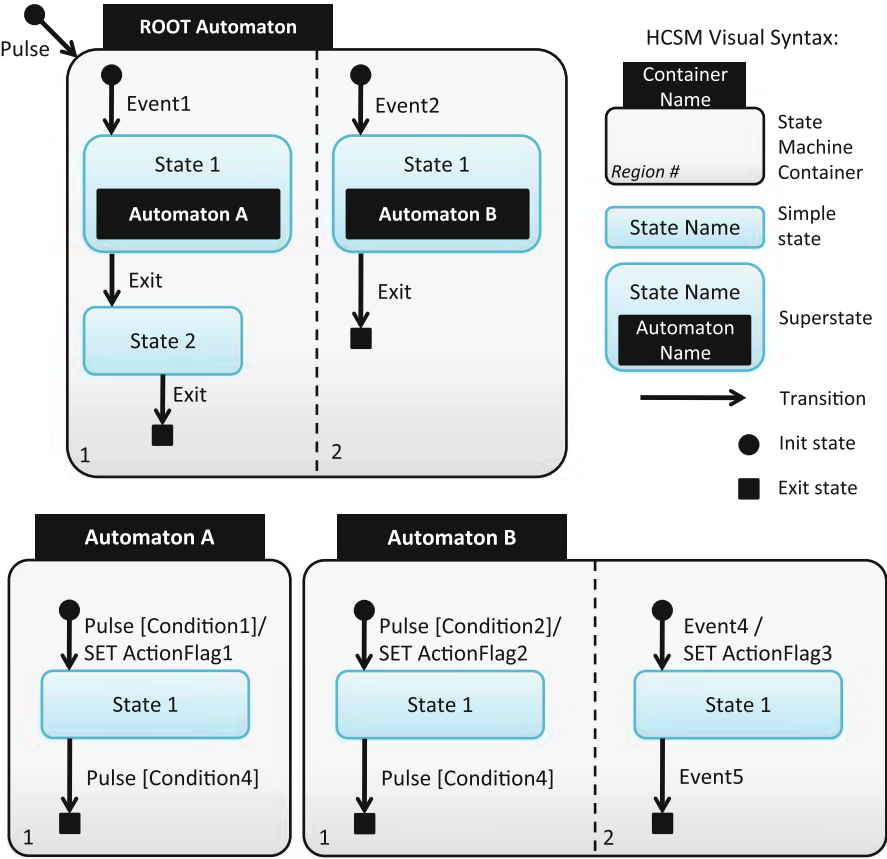
The HCSM specification language and computation model are to an extent influenced by the Statecharts formalism (Harel 1987) which has certain similarities to MATLAB Stateflow. However, the HCSM language and model have several differences intended to support a clean visual syntax, a clear semantics, and a highly efficient implementation providing strong real-time guarantees. HCSMs can also be interpreted as opposed to being compiled. This facilitates reconfiguring the system at runtime.

State Machines, Hierarchy, and Concurrency. A *state machine* consists of states and transitions. A state represents any activity of a system at any level of abstraction. Two main types of states are defined: *simple states* and *superstates*. A superstate represents a nested state machine, allowing a hierarchy of state machines to be created. Two special types of simple states are defined: *init* (the starting state) and *exit* (which terminates execution). Figure 37.9 presents a visual syntax for HCSMs and provides an example of a simple state and a superstate: *State 2* and *State 1* of the root automaton, respectively. In the remainder of this chapter, the terms *state machine* and *automaton* will be used interchangeably and will refer to HCSMs.

A *state machine container* is a collection of one or more concurrent (child) state machines. Each of these machines is contained in a *region*. Regions are ordered by a consecutive number (RegionNumber) and are separated by a dashed line in the visual syntax. For example, *Automaton B* contains two regions in Fig. 37.9.

Hierarchies provide a powerful mechanism for encapsulating specific behaviors. The framework permits reuse of state machine containers, providing all the necessary means to execute multiple instances of a state machine. Such encapsulation also provides a practical way of aborting a particular behavior. *Concurrency* prevents combinatorial explosion of states, which would occur in FSMs, and permits easy handling of asynchronous state machines. In the HCSM visual language, a *hierarchy* is modeled by vertical state machine decomposition and *concurrency* by horizontal decomposition (Fig. 37.9).

An example where one can take advantage of concurrency and hierarchy is when modeling complex behaviors such as vision-based landing (Sect. 37.3.4). The landing mode consists of several lower-level behaviors controlled by the main superstate of the landing mode. For instance, it includes control laws steering the helicopter, and it coordinates the camera system and image processing functionalities. When the landing behavior is activated, several state machines modeling the necessary activities are executed. These include searching for a predefined pattern with the



Syntax of a transition in EBNF notation:
Transition = EventID ["[" Guard "]"] ["/" ActionList]
ActionList = Action { "," Action }
Action = SET FlagID | UNSET FlagID | GET MemorySlotID | PUT MemorySlotID |
SEND EventID MemorySlotID DestinationID
Guard = ["-"] FlagID | Guard Op Guard | "(" Guard ")"
Op = "&" | "V"
(* all ID symbols are ANSI C id-tokens *)
(* all Name symbols are sequences of ASCII characters *)

Fig. 37.9 HCSM visual syntax with an example of three state machines

camera system and inputting the image processing results to a Kalman filter which fuses them with inertial measurements. Once the pattern is found, another state machine controls the camera in order to keep the pattern in the center of the image. This increases the robustness of image processing when the helicopter is close to the ground or in the presence of strong wind gusts.

State Transitions, Events, and Guards. A transition between two states is triggered by an *event*. Events in the HCSM framework can be generated internally by the state machine itself or externally. Both asynchronous (or sporadic) and periodic events are supported.

There are two types of special events. A *pulse* event is a periodic event generated before each iteration of the HCSM algorithm, similar to a clock pulse (discussed later). For example, this can be used to trigger a transition without a specific asynchronous event. An *exit* event is created when a state machine is in its exit state and it is only sent to the parent superstate informing it that a child automaton has finished its execution.

State transitions can optionally be guarded by *conditions* in the form of Boolean expressions. If an event for a particular state transition has been generated and the condition guarding the transition is TRUE, then the state transition takes place.

Activities vs. Actions. As in Harel (1987), a distinction is made between actions and activities. Actions have no duration (zero-time assumption) and are executed in transitions (as in a Mealy machine), while activities take time and are associated with states (as in a Moore machine).

During each transition, a possibly empty set of *actions* can be executed. Supported actions include setting binary flags (SET *flag-name*), sending events (SEND *event-name data target-computer*), retrieving (GET *source*), and storing (PUT *dest*) data. Data is transmitted using a predefined memory bank with labeled slots. Data from received events is automatically stored in the *cache* memory slot. A GET action copies data from a source slot to the cache slot. A PUT action copies data from the cache slot to a destination slot.

Activities are defined in terms of regularly executed functions. They are coordinated by a set of binary flags which are changed by actions. Functions are executed outside the state machine algorithm, and their execution is discussed in the next section.

HCSM Design and Execution. As illustrated in Fig. 37.10, the design of a system in the HCSM framework starts either by generating a visual description of state machines using a graphical tool or (in simpler cases) by directly describing state machines in a text-based language. In either case, tables describing transitions can be derived and passed to the system and are then interpreted by HCSM Interpreters at runtime on the robotic platform. Due to its compact and efficient implementation, the interpreter can run in the real-time part of the system as a periodic task with high execution rate.

The HCSM supports AND/OR/NOT operators in condition statements directly in the visual description and HCSM text files. However, the atomic conditions and the activities used are implemented procedurally. A library of such conditions and activities is generally written in an early development phase, after which the modeling of the system behavior itself can take full advantage of the flexibility of the interpreted state machine language with no need for recompilation.

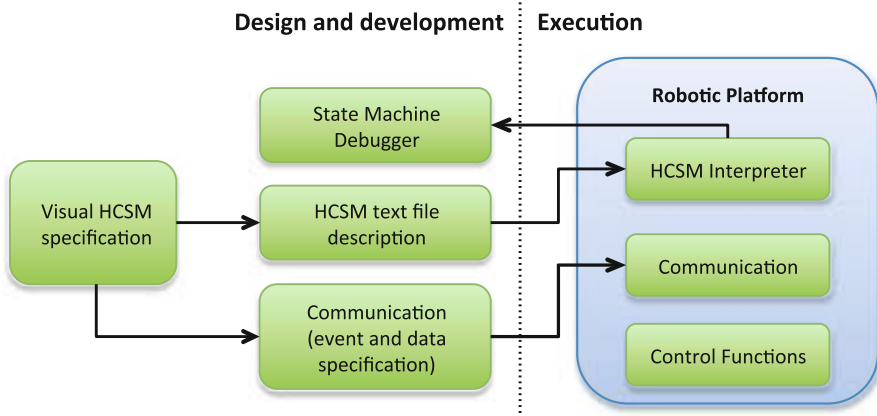


Fig. 37.10 Overview of the HCSM design process and execution

Three forms of communication are supported: (1) between states of the HCSM language processed by the same interpreter, (2) between computer systems of the same robot, and (3) between different robots or robots and operators. The first form is realized by an *internal event queue*, while the remaining two are realized by transmitting external events in packets with predefined sizes. Received external events are put in an *external event queue*. The framework directly supports real-time transmissions with built-in integrity checking given that the network satisfies real-time properties, and it automatically generates the required communication code during the design process (Fig. 37.10).

Abstractly, HCSM execution proceeds as shown in Algorithm 1 (see Sect. 37.4.1.1 for concrete examples). The automatically generated communication functions are called in the *Communicate* function. Events received by the system are parsed, checked for integrity and put in the external event queue. The HCSM is then executed (*ExecuteStateMachine*). Binary flags set by the state machine transitions (using actions) coordinate the execution of the control functions (*RunControlFunctions*) which are defined by the user.

Details are shown in Algorithm 2. Transitions in HCSMs are divided into *macro steps* and *micro steps*, where external events are considered in the macro step and internal events in the micro step. A macro step (*ExecuteStateMachine* lines 5–9) begins by processing the first event from an external event queue. Events are processed in the order they were received. An event can trigger one or more transitions. These transitions may generate both internal and external events which in turn trigger more transitions. The macro step is finished when the external event queue is empty and no more transitions are made. Micro steps are steps within a macro step (*ExecuteStateMachine* lines 7–8).

In case the external event queue is empty, only the micro steps are executed, which results in all of the internal events being processed (at the beginning one pulse event is in the internal event queue).

Algorithm 1 Skeleton of a procedure for execution of a HCSM-based system on a single computer

– *Main Execution Loop*

```

1: while system is running do
2:   ...
3:   Communicate();           {Send and receive data packets containing debug information
                             and external events. Events are put in the external event queue.}
4:   ExecuteStateMachine();   {Execute Algorithm 2}
5:   RunControlFunctions();    {Run the appropriate control functions based on the binary
                             flags set by actions in the HCSM.}
6:   ...
7: end while

```

Algorithm 2 The HCSM algorithm

– *ExecuteStateMachine*

```

1: lock memory
2: create empty internal event queue
3: append pulse event to internal event queue
4: repeat
5:   remove first event from external event queue and append to internal event queue
6:   while internal event queue not empty do
7:     remove first event e from internal event queue
8:     call MakeTransitions(1, e)
9:   end while
10: until external event queue empty
11: unlock memory

```

– *MakeTransitions(MachineLevel, Event)*

```

1: for all concurrent state machines M at MachineLevel do
2:   if Event is received by transition of current state in M and Guards are TRUE then
3:     call ExecuteActions(actions associated with the transition)
4:     make transition
5:   else if current state is superstate then
6:     call MakeTransitions(MachineLevel+1, Event)
7:   end if
8: end for

```

– *ExecuteActions(ActionList)*

```

1: for each action Action in ordered ActionList do
2:   if Action is send event action then
3:     if destinationID is external computer then
4:       append Event to external communication queue
5:     else
6:       append Event to internal event queue
7:     end if
8:   else
9:     execute action
10:  end if
11: end for

```

The system assumes the “synchrony hypothesis”: During a macro step, inputs do not change and external events are not received. In practice, external events are not processed as soon as they arrive, but they are buffered until the state machine interpreter is called.

On the UASTL RMAX, the state machine interpreter is called periodically every 20 ms in the real-time environment. The duration of one *macro step* is set to 200 μ s which corresponds to the worst-case execution time. This time depends on the complexity of the state machine being executed, which is expressed in terms of the number of regions and the maximum number of *micro steps* (generated events). The periodic update time and the duration are user configurable and were empirically chosen for the UASTL RMAX system.

Related Work. Many specification languages, software frameworks, and design tools are used in control or embedded system design, including Ptolemy II (Eker et al. 2003), Esterel (Berry 1992), Statecharts (Harel 1987), Stateflow (<http://en.wikipedia.org/wiki/Stateflow>), Simulink (<http://en.wikipedia.org/wiki/Simulink>), and UML 2 (Booch et al. 2005), but none of these are optimized for building control systems for autonomous robots.

HCSMs are primarily influenced by Harel’s Statecharts formalism. State machine-based approaches have already been used successfully in many robotic systems. Brooks (1989), for instance, uses state machines to build reactive systems, and Kleinhagenbrock et al. (2004) include them in a deliberative/reactive system. Albus and Proctor (1996) propose an architecture for intelligent hybrid control systems which has some similarities with the HCSM framework. It also includes state machines, defines a hierarchy of functional modules, and includes a communication system, but it lacks some of the features mentioned above. The HCSM framework supports the component-based design methodology. In Brooks et al. (2005), a component-based framework is proposed which aims for similar goals, but it also does not provide some of the features mentioned above (such as real-time aspects). Cremer et al. (1995) propose a framework for modeling reactive system behavior in virtual environment applications. Although some similarities in the modeling formalism exist, it lacks some of the fundamental features required for use in robotics applications such as strong real-time execution guarantees and a clear definition of the visual syntax necessary in the design of complex control systems.

37.4.1.1 Using HCSMs on the UAS Tech Lab RMAX

HCSMs are used on the UASTL RMAX platform as a low-level real-time mechanism for modeling system behavior and are executed on all three of its onboard computers. This is illustrated in Fig. 37.11 which also shows the Platform Server that will be discussed in Sect. 37.4.2.1. HCSMs are used on the PFC computer for modeling and executing the control system. HCSMs control a set of sensors and available perception functionalities on the PPC computer. On the DRC computer, HCSMs provide an interface to flight control modes, payload, and perception functionalities in the Control Kernel (Sect. 37.3) accessible to the high-level deliberative and reactive algorithms. Because the HCSMs running on different

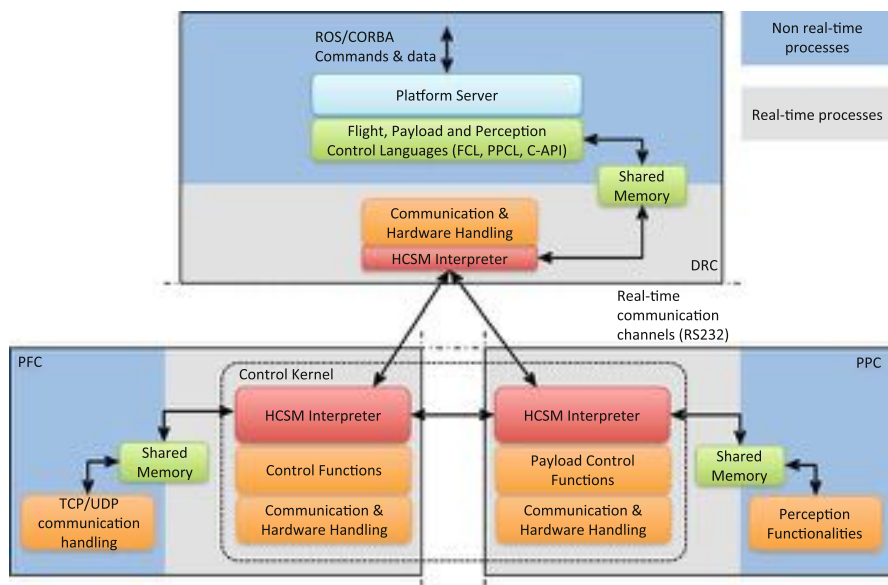


Fig. 37.11 HCSMs and the control kernel

computers communicate with each other, all of the individual events used in the system have globally unique IDs.

Several non-real-time processes also run on each of the computers and communicate with the real-time HCSMs through shared memory. Specifically:

- **PFC:** The TCP/UCP communication handling necessary for sending HCSMs status data used for state machine debugging interfaces (see Fig. 37.10)
- **DRC:** The *Platform Server*, which provides the deliberative and reactive services with an interface to the Control Kernel through the FCL and PPCL languages (Sects. 37.4.2.1–37.4.2.3)
- **PPC:** The perception functionalities available in the Control Kernel

The remainder of the section presents an example using HCSMs running on the PFC computer. An overview of all HCSMs running on the PFC system is given, followed by a description of two state machines involved in control mode switching. A more detailed example of the use of HCSMs in the context of the path-following control mode (PFCM) is presented in Sect. 37.5.2.

Overview of the PFC HCSMs. Figure 37.12 presents a hierarchical view of all of the 15 automata that run on the PFC computer. The whole UASTL RMAX system uses 207 events in total. Note that the HCSMs presented in this section are examples from the deployed system. Various extensions and different ways of modeling to achieve the same functionality of the system are of course possible.

The software-controlled power system (Sect. 37.2.3) is managed by the PFC computer. This allows all devices except the PFC itself to be switched on and off

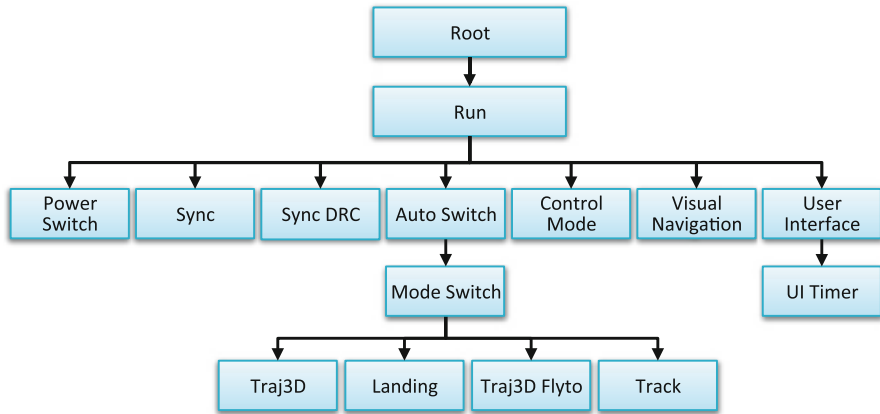


Fig. 37.12 A hierarchical view of the HCSM automata running on the PFC computer

through software coordinated by the *Power Switch* automaton. At system start-up, the *Root* automaton makes sure that at least the DRC computer is switched on (in case it has not been manually switched on by the user). The *Root* automaton also contains a superstate for the *Run* automaton, which in turn contains only one superstate for the following automata: *Power Switch*, *Sync*, *Sync DRC*, *Auto Switch*, *Control Mode*, *Visual Navigation*, and *User Interface*.

The *Sync* and *Sync DRC* automata are responsible for achieving a common state among the different computers through synchronization. For example, the common state includes the local computer time and the ground position and altitude reference. The *Sync DRC* automaton handles the synchronization between the PPC and DRC computers. The *Sync* automaton is responsible for sending the common state from the PFC to the PPC and DRC computers.

The *User Interface* automaton handles commands received from a ground control user interface (UI), while the *UI Timer* automaton implements timeouts for accepting commands by the *User Interface* state machine when using a miniaturized external keyboard.

The *Visual Navigation* automaton coordinates the execution of a localization algorithm that runs on the PPC and is based on reference images (Conte 2009). The *Auto Switch* automaton handles helicopter platform initialization and operational modes, including the following aspects:

- Initializing the UASTL RMAX helicopter system before it is ready to accept commands. This includes monitoring the engine-on status and the RC radio transmitter status.
- Monitoring sensor data and initializing a Kalman filter used for the state estimation of the helicopter.
- Handling the simulation mode of a hardware-in-the-loop simulator that uses a dynamic helicopter model (Duranti and Conte 2007; Conte 2007) and runs all of

the software components used during a real flight onboard the AV. This allows newly developed functionalities to be tested before a real flight is performed.

- Handling the RC radio transmitter switch that selects manual or autonomous flight mode.

The *Auto Switch* automaton starts up the execution of the *Mode Switch* state machine after a successful initialization of the system when the autonomous flight mode switch is selected. The *Mode Switch* and *Control Mode* automata in turn handle the switching between flight control modes. This includes initializing, running, and handling error conditions for particular control functions.

Certain control modes are also associated with their own mode-specific state machines: *Traj3D* for the path-following control mode (Conte et al. 2004), *Landing* for the vision-based landing control mode (Merz et al. 2004), *Traj3D Flyto* for a simplified path-following mode which only uses straight line paths, and *Track* for a reactive car tracking mode. Other control modes such as takeoff and hovering do not require multiple states and are therefore modeled directly by the *Mode Switch* and *Control Mode* automata without any additional state machines.

Periodic Execution. The *Main Execution Loop* in Algorithm 1 executes periodically at a rate of 50 Hz. In each iteration data packets can be sent and received, and HCSMs are executed (Algorithm 2) and can update a set of action flags using SET. The active control mode is then updated based on the current values of these flags, after which the flags are automatically unset. Thus, action flags should essentially be seen as triggers that switch control modes. Finally, the currently selected control mode is executed.

Examples. Figure 37.13 presents two state machines that handle the execution of control modes and implement a mode switching mechanism for the UASTL RMAX. For clarity the following notation is used:

- For fast identification, each automaton has a letter label and all state transitions are numbered.
- Names of states end with “-ST,” while names of events end with “-EV.”
- Names of conditions end with “-CO.”
- Names of SET action flags end with “-AC.”
- Names of labeled memory slots (data offsets) end with “-DO.”

Each control function follows a predefined template with at least three internal states (*initializing*, *running*, and *stopping*) and two internal flags: an input flag used to switch between the internal states and a status flag used for error handling (e.g., when the initialization of the function fails). Both flags are used by the *Control Mode* automaton which models the internal states of the function (Ctrl-Init-ST, Ctrl-Run-ST, and Ctrl-Stop-ST). The additional states present in the automaton (Ctrl-Off-ST and Ctrl-Error-ST) are used to make sure the previous function has been properly stopped and no other functions are initialized.

The *Control Mode* automaton (Fig. 37.13) sets the appropriate input flag and keeps track of the control function status. Because each control function follows the

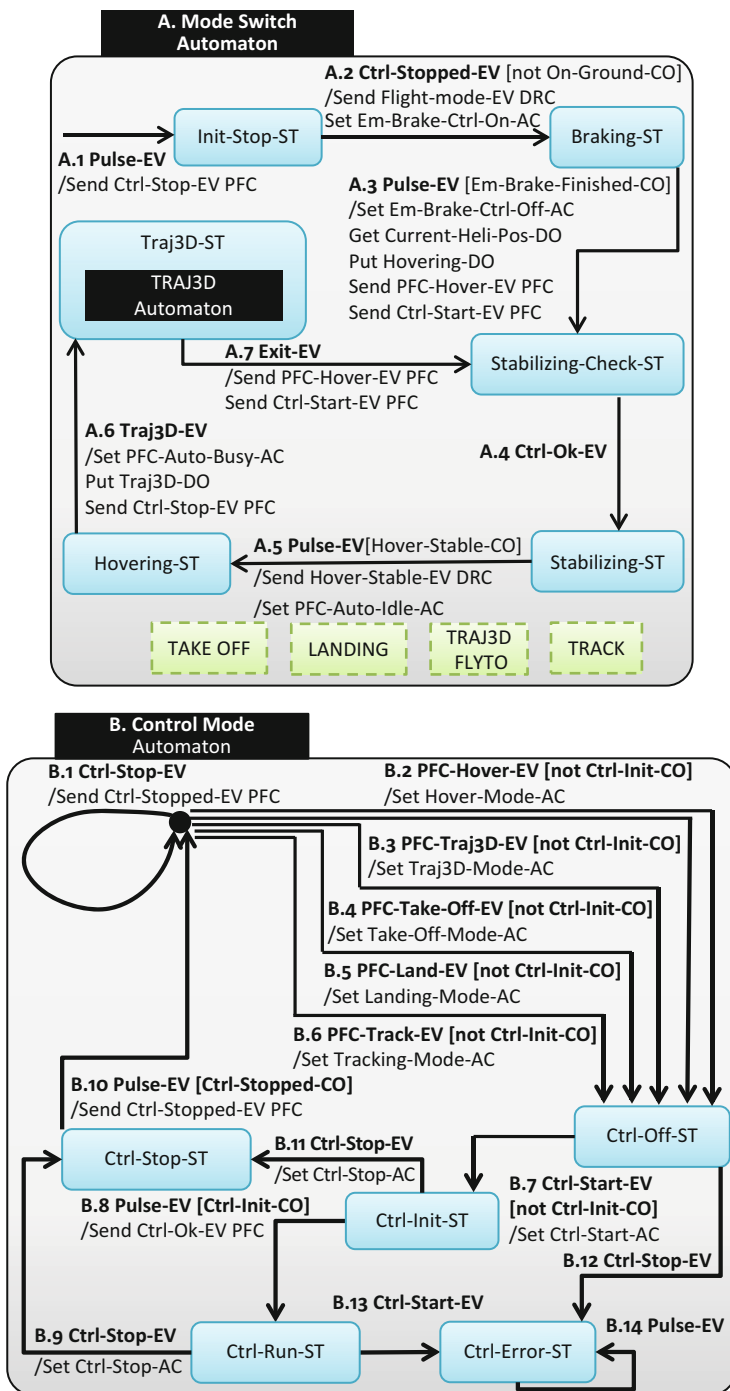


Fig. 37.13 The Mode Switch and Control Mode automata

predefined template, the *Control Mode* state machine handles all modes transparently, and there is no mode-specific state. When a particular control function should be executed, a state machine (such as *Mode Switch*) sends a mode-specific start event (such as PFC-Hover-EV) to the *Control Mode* automaton. This triggers one of the transitions B.2, B.3, B.4, B.5, or B.6 during which appropriate internal flags are set by executing SET actions (e.g., SET Hover-Mode-AC). In the next iteration of the *Main Execution Loop*, the control function starts its execution passing through the initialization state (Ctrl-Init-ST). If the initialization is successful, the *Control Mode* automaton switches its state to Ctrl-Run-ST, and the control function is executed periodically in each iteration.

Control Mode mainly interacts with the *Mode Switch* state machine which implements flight mode switching, including sequentialization and coordination of control function execution. For example, it ensures that after the execution of the path-following control mode, a default hovering mode is switched on.

The *Mode Switch* state machine, additionally, generates events that are sent to the high-level system (DRC), for example, in the A.2 transition. It also reacts to events sent from the DRC, such as A.6.

Use Case: Example of Control Mode Switching. The following example relates to a simple use case where a backup pilot performs a manual takeoff procedure. When the maneuver is finished, the AV control is switched to autonomous mode by using the auto switch button on the RC radio transmitter. A default hovering function should then be engaged as described below.

Figure 37.14 shows the interaction between the *Control Mode* and *Mode Switch* automata for this example. The time line shows state transitions and events exchanged between the two state machines. The execution starts with an exchange of two events, Ctrl-Stop-EV (A.1) and Ctrl-Stopped-EV (B.1), to make sure no other control function is active. The *Mode Switch* automaton executes a braking procedure (Braking-ST). This procedure, also called emergency braking, is designed to stop the helicopter before the default hovering mode is engaged. The procedure uses the path-following control mode with a predefined straight line path and a zero target velocity value as an input.

When braking has finished, the current helicopter position is saved to be used for the hovering function (in the Hovering-DO memory slot). Two events are also sent to the *Control Mode* automaton. The first event (PFC-Hover-EV, A.3) makes *Control Mode* select the hovering function for execution (SET Hover-Mode-AC). The second one (Ctrl-Start-EV, A.3) starts the initialization of the hovering control function.

After a successful initialization, Ctrl-Ok-EV (B.8) is generated and *Mode Switch* changes its state to Stabilizing-ST. The automaton remains in this state until a Hover-Stable-CO condition is satisfied. The condition checks if the position, altitude, heading, and velocity are within hovering tolerance bounds. The tolerances used in the UASTL RMAX system are set to 5 m for position, 2 m for altitude, 5° for heading, and 1 m/s for vertical and horizontal velocities.

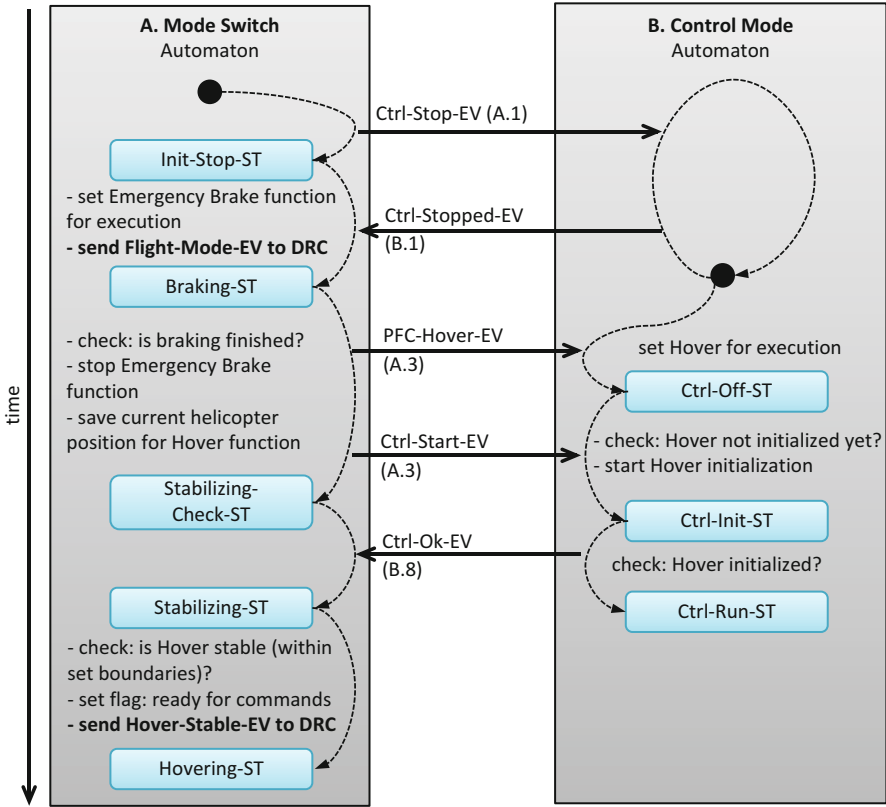


Fig. 37.14 Execution trace: The interaction between the Control Mode and Mode Switch automata

When the Hover-Stable-CO condition is satisfied, a Hover-Stable-EV event is sent to the DRC computer, and an internal system flag is set to indicate that the system is in the autonomous mode and ready to accept new commands (SET PFC-Auto-Idle-AC). The Mode Switch automaton changes its state to Hovering-ST.

Extended State Machines. The HCSM framework has been further developed, resulting in Extended State Machines (ESMs) (Merz et al. 2006). The new framework adds several useful modeling features. The main changes include explicit modeling of task states, data flow, control and system flags, an event filtering mechanism, and no explicit external events. It also includes a visual tool for designing and debugging state machines, facilitating the development of new and existing systems.

ESMs use three types of states: *simple states*, *superstates* (as in the HCSMs), and *task states*. Control and other functions are modeled explicitly in the task states. A schedule for function execution is provided by a scheduler included in

the framework. The data used as input/output to the task state functions (data flow) is also explicitly modeled in the ESM formalism by *Data paths*. Asynchronous external events are modeled by a combination of *pulse event* and guard conditions. Thus, only one internal event queue is used in the ESM. Additionally the ESM introduces an event filtering mechanism which limits the scope of internal events. The full details are available in Merz et al. (2006).

37.4.2 Interfacing Between the Control Kernel and Higher Layers

37.4.2.1 The Platform Server

The *Platform Server* (Fig. 37.11 on p. 22) encapsulates the functionality of the Control Kernel to provide a higher-level control and sensing interface for use by any of the functionalities in the reactive and deliberative layers. These functionalities interface to the Platform Server through two languages, the Flight Control Language (FCL) and Payload and Perception Control Language (PPCL), described in the following subsections. The former provides high-level parameterized commands which actuate the helicopter itself, while the latter provides commands which actuate sensor payloads and call perception-based functionality. Many of these commands provide feedback to their callers in the form of events which are communicated through a shared memory resource. Each command is essentially associated with one or more HCSMs.

From one perspective, the commands associated with FCL and PPCL provide the primitive or basic actions provided by the UASTL RMAX system. Specification of more complex tasks used by different functionality in the reactive and deliberative layers uses these basic actions as part of their definition. The intent is that most if not all use of the functionality in the control layer is accessed through the Platform Server interface.

This is a design decision that encapsulates the use of the control and sensor system in the UASTL RMAX through one common language interface. This makes the Control Kernel extensible in a straightforward manner. Any new control modes or addition of sensors is added to the system by implementing appropriate HCSMs to deal with real-time properties and temporal latencies near the hardware. The FCL and/or PPCL is then extended with an appropriate command interface.

Since the functionalities in the reactive and deliberative layers are only committed to soft real-time behavior while the HCSMs encapsulate real-time constraints, the Platform Server serves as an intermediary between these different temporal latencies. The distributed nature of the HCSM framework allows different functionalities to run on different computers. Each of the onboard computers therefore executes its own HCSM interpreter, and peer-to-peer RS232 serial lines are used to implement real-time communication channels between these. This allows the Platform Server to run on the DRC computer despite the fact that it must communicate with HCSMs associated with flight control modes and payload control which rely on timely execution and run in the real-time part of the system on the PFC and PPC computers.

Calling the Platform Server. The Platform Server runs on the DRC computer and provides a ROS-based interface to the commands available in the FCL and PPCL languages. When the server is asked to execute a specific command, it internally forwards the command through a lower-level programming interface based on the FCL and PPCL (“C-API” in Fig. 37.11). This API in turn uses shared memory to forward the command to an HCSM executing in the real-time part of the DRC.

The HCSM on the DRC generates one or more events that are sent through the real-time communication channel to an HCSM on either the PFC or the PPC, depending on the command. The events generated on those computers, as well as any associated sensor data generated in the CK, are passed back to the Platform Server in the same manner.

FCL and PPCL commands are executed sequentially without an internal command queue for buffering commands. If a command has been accepted for execution, a success flag (FCL_OK or PPCL_OK) is returned. In case the system was already executing another command, a busy flag (FCL_BUSY) is returned. In case a new command has to be started immediately, the caller can first abort any currently running activity by sending a Cancel command.

Data Interface. An additional task of the Platform Server is to provide a Data Interface (DI) to the state of the UASTL RMAX system. The data periodically published through this interface contains the AV’s state information (position, attitude, velocities), status information (such as the current engine RPM), payload information (such as the current position of the pan-tilt unit), and the results of perception-based functionalities (such as the color tracker). The Data Interface provides a common interface accessible by all deliberative and reactive functionalities. For example, the information obtained through the DI is used by the DyKnow system (Sect. 37.6) and by execution monitoring (Sect. 37.7.3).

37.4.2.2 FCL: Flight Control Language

The Flight Control Language (FCL) consists of a set of commands that are based on the control modes available in the Control Kernel and their parameterization. The reactive and deliberative services use the FCL commands for mission execution as it provides the interface to the available control functionalities in the CK.

The following are the most important commands currently available on the UASTL RMAX platform. The list is not exhaustive: Certain housekeeping commands, parameters, and return values are omitted here.

- *Takeoff*: `fcl.takeoff()` takes off to a predefined altitude (using the takeoff control mode). Returns:
 - FCL_BUSY: Another command is being executed.
 - FCL_NOT_ACCEPTED: Takeoff can only be commanded when the AV is on the ground.
 - FCL_FINISHED.
- *Land*: `fcl.land(heading)` performs a vision-based landing (using the landing control mode). The parameter specifies the desired heading for landing. This heading

should take into account the current position of the sun in order to avoid having the shadow of the AV fall on the landing pattern. Returns:

- FCL.BUSY: Another command is being executed.
- FCL.NOT_ACCEPTED: This command can only be issued if the AV is hovering.
- FCL.FINISHED.
- *Yaw*: `fcl.yaw(heading)` changes the heading to a specific value (using the hovering control mode with a specified heading).
Returns: The same values as for the `fcl.land()` command above.
- *Climb*: `fcl.climb(altitude)` climbs or descends to the given altitude (using the path-following control mode with a predefined vertical path).
Returns: The same values as for the `fcl.land()` command above.
- *Traj3d fly-to*: `fcl.traj3d.flyto(longitude, latitude, altitude, velocity)` flies to a position in a straight line (using the path-following control mode with a single straight line segment). The parameters specify the world position to fly to and the desired cruising velocity.
Returns: The same values as for the `fcl.land()` command above.
- *Traj3d*: `fcl.traj3d(spline_description, velocity, end.velocity)` flies between two way-points following a spline specification (using the path-following control mode). The parameters specify the beginning, ending, and direction vectors of the spline path and the cruise/final segment velocities (Sect. 37.3.3).
Returns: The same values as for the `fcl.land()` command above.
- *Track*: `fcl.track()` engages the object-following control mode (uses the path-following control mode, camera pan-tilt visual servoing control and vision-based color object tracking).
Returns: The same values as for the `fcl.land()` command above.
- *Emergency brake*: `fcl.emergency.brake()` engages the emergency brake mode (path-following and hovering control modes). This immediately aborts all flight control modes and engages the hovering control mode.
Returns: FCL_OK.
- *Cancel*: `fcl.cancel()` cancels the execution of the current FCL command.
Returns: FCL_OK.

For example, to scan an area in a given environment where the AV starts on the ground, the FCL commands used would include a *takeoff*, a sequence of *fly-to* commands, and a *land* command.

37.4.2.3 PPCL: Payload and Perception Control Language

Similarly to the FCL, the Payload and Perception Control Language (PPCL) consists of a set of commands that are based on the modes available in the CK and their parameterization. The commands in this group relate to the use of payload control modes and perception functionalities.

The following are the most important PPCL commands currently available on the UASTL RMAX platform. The list is not exhaustive: Certain housekeeping commands, parameters, and return values are omitted here.

- *Request PTU, camera, laser range finder, and perception functionality control*: Requests exclusive control and ownership for these functionalities. For example,

the control over the pan-tilt unit and camera parameters cannot be granted during a vision-based landing. The landing control mode handles these parameters.

- `ppcl.request_ptu_control()`: request control over the pan-tilt unit.
- `ppcl.request_cam_control()`: request control over the camera parameters.
- `ppcl.request_ip_control()`: request control over the image processing algorithms.
- `ppcl.request_laser_control()`: request control over the laser range finder.

Returns:

- `PPCL_CONTROL_DENIED`
- `PPCL_OK`

- *Release PTU, camera, laser range finder, and perception functionality control:*

Releases the control over these functionalities.

- `ppcl.release_ptu_control()`: release control over the pan-tilt unit.
- `ppcl.release_cam_control()`: release control over the camera parameters.
- `ppcl.release_ip_control()`: release control over the image processing algorithms.
- `ppcl.release_laser_control()`: release control over the laser range finder.

Returns:

- `PPCL_OK`

- *Pan-tilt unit control:* engages one of the pan-tilt control modes.

- `ppcl.look_at(longitude, latitude, altitude)`: world coordinates of a position at which the camera should be pointed.
- `ppcl.turn_to(pan_angle, tilt_angle)`: values of the pan-tilt unit axis to set.
- `ppcl.ptu_abort()`: abort the current control mode.

- *Perception control:* engages one of the perception functionalities.

- `ppcl.ip_color_tracker(x, y)`: coordinates of a center of a object to track.
- `ppcl.ip_state_estimation()`: engages the vision-based state estimation algorithm.
- `ppcl.ip_abort()`: abort the current command.

- *Camera parameters control:* sets the required camera parameters.

- `ppcl.zoom(value)`: the value of the camera zoom from 0 (full zoom out) to 1 (full zoom in).
- `ppcl.exposure(auto)`: specifies whether automatic or manual exposure should be used.
- `ppcl.iris(value)`: the value of the camera iris to set.
- `ppcl.shutter(value)`: the value of the camera shutter to set.

- *Laser parameters control:* sets the required laser range finder parameters.

- `ppcl.laser_angular_resolution(value)`: the angular resolution of the laser sensor (1, 0.5, 0.25°).
- `ppcl.laser_distance_resolution(value)`: the distance resolution of the laser sensor (1 mm, 1 cm).
- `ppcl.laser_rotation_speed(value)`: the speed and direction of the laser rotation mechanism.
- `ppcl.laser_rotation_angle(value)`: the angle of the laser rotation mechanism to set.

- *Laser control*: engages one of the laser perception functionalities.
 - `ppcl.laser_start_data_collection()`: starts collection of the laser range finder data.
 - `ppcl.laser_stop_data_collection()`: stops collection of the laser range finder data.
- *Pan-tilt activity info*: returns which pan-tilt control mode is currently active.
 - `ppcl.ptu_activity_info()`.
 Returns:
 - `PPCL_LOOK_AT_POINT`
 - `PPCL_TURNTO`
 - `PPCL_IDLE`
- *Camera parameters*: returns parameters of a specific camera.
 - `ppcl.get_camera_info(index)`: index of the camera (color or thermal)
 Returns: Intrinsic camera parameters.
- *Pan-tilt parameters*: returns parameters of the pan-tilt mechanism.
 - `ppcl.min_pan_value()`, `ppcl.max_pan_value()`, `ppcl.min_tilt_value()`, `ppcl.max_tilt_value()`:
 Returns: Ranges of pan-tilt unit angles which can be set.

In the case of scanning an area in a given environment, the required PPCL command sequence would include *request control: pan-tilt*, *request control: camera*, *pan-tilt control: tilt 90°*, *camera control: set specific zoom*, *release control: pan-tilt*, and *release control: camera*.

37.4.2.4 DI: Data Interface

Along with the FCL and PPCL languages, the Control Kernel also provides information about the state of the UASTL RMAX platform. This data is made available to the higher layers periodically at the rate of 25 Hz. It is currently distributed in the system in the form of ROS topics. The two most commonly used data collections provided by this interface are:

- *Helicopter state*:
 - altitude, latitude, longitude: current position.
 - velocity_north, velocity_east, velocity_up: current velocity in the given direction.
 - pitch, roll, heading: current pitch angle, roll angle, and heading.
 - rpm: engine revolutions per minute.
 - on_ground_flag: true if the AV is on the ground.
 - auto_flag: true if the AV is in autonomous flight mode.
 - drc_minus_pfc_time, ppc_minus_pfc_time: time differences between computers.
 - power_status: indicates which onboard devices are switched on.
 - pfc_time: data timestamp.
- *Camera state*:
 - id: id of the camera (color or thermal).
 - pan, tilt: current position of the PTU's pan and tilt axis, respectively.
 - zoom: current zoom factor of the camera.
 - ipc_time: timestamp.

37.4.3 Task Specification Trees

The concept of a *task* has many different interpretations in the literature and has been instantiated in many different ways in architectures proposed for mobile robots. One way to specify a particular type of task that has strict real-time requirements is using HCSMs. This type of task is often called a *behavior* in the robotics literature (Arkin 1998; Konolige et al. 1997), and different ways are proposed for combining such behaviors. At the other end of the spectrum, a task is often interpreted as an action or combination of actions (composite action) that are combined using standard control structures such as sequentialization, concurrency, and conditional branching. In fact, there have been several proposals to simply implement robot behaviors and tasks using a subset of a conventional programming language such as C (Konolige 1997). Firby's Reactive Action Package (Firby 1987) is one example where tasks are specified using a specialized language which combine actions with control. RAPs are often interpreted as *compiled plans*. Automated planners themselves combine actions specified as plan operators. The output of any planner can be interpreted as a composite action or task. The transition from plans to executable tasks is an essential part of any intelligent system.

The choice and construction of a task specification language which covers much of the spectrum of interpretations above has to be chosen with generality in mind in order to be used as a common interface to many diverse functionalities, but at the same time, the actual tasks have to be efficiently executable. The choice is further complicated by the fact that when specifying collaborative mission scenarios, tasks are often shared and must be distributable. An additional factor is that in the context of heterogeneous robotic systems, the specification of tasks should include context. For example, the **fly-to** action of a fixed-wing platform and the **fly-to** action of a rotor-based platform have much in common but also much that is different.

Task Specification Trees (TSTs) have been proposed as a general task specification language which meets many of the requirements stated above. Each node in such a tree is a specification of an elementary action such as a ROS call to an FCL command; a control structure such as sequential, concurrent, branching, or loop execution; or a goal for which a plan must be generated and executed (Sect. 37.7.2). In the latter case, the tree is *dynamically expandable* as a plan is generated. Goal nodes are therefore allowed to add children to themselves, after which they act as sequence or concurrency nodes. TSTs also allow for the statement of context in the form of constraints which can then be checked for consistency with a constraint solver. Additionally, these constraints are often temporal and spatial in nature and implicitly combine both a reasoning and procedural execution mechanism for tasks.

A TST is in itself purely declarative, defining what should be achieved and providing parameters and constraints for tasks. For example, a sequence node declaratively specifies that its children should be sequentially executed while a **fly-to** node would specify that an aircraft should fly to a specific location, usually with associated parameters such as the intended speed and altitude, possibly constrained by context.

At the same time there is a close integration with execution aspects of actions and action composition constructs through the coupling of *executors* to specific types of TST nodes. Each executor is an executable procedure specifying *how* a task of the given type should be performed. In many cases, the interface provided by the Platform Server is used to issue commands in the Flight Command Language and Payload and Perception Control Language (Sect. 37.4.2). The implementation of the executor can be platform specific, allowing the exact mode of execution for a particular task to be platform dependent as is often necessary when heterogeneous platforms are used. Thus, the use of executors provides a clear separation between task specifications and platform-specific execution details.

In the case of the UASTL RMAX system, nodes for actions such as takeoff, vision-based landing, hovering, or path-following are generally associated with executors calling the Platform Server to initiate the corresponding autonomous flight modes. Through this interface TST nodes can also control the payload and sensors of the UAS platform and receive associated information including image processing results. Note that executors are required even for *control nodes* or *structural nodes*, though such executors may be identical across platforms. For example, a sequential node requires an executor that procedurally ensures its children are executed in sequential order, possibly with additional timing constraints.

TSTs have been applied and tested successfully in a number of deployed UAS systems (Doherty et al. 2010, 2013; Landén et al. 2010).

Example 37.1. Consider a small scenario similar to the first goal in the example described in the introduction. The mission is that two AVs should concurrently scan the areas Area_A and Area_B , after which the first AV should fly to Dest_4 (Fig. 37.15a). The corresponding TST (Fig. 37.15b) uses three *elementary action nodes* (marked E), corresponding to two elementary actions of type **scan-area** and one of type **fly-to**. Furthermore, it requires a *concurrent node* (marked C) specifying that the **scan-area** actions can be performed concurrently, as well as a *sequential node* (marked S). Further explanations will be given below.

37.4.3.1 Task Specification Tree Structure

Task Specification Trees are implemented as a distributed data structure where nodes can reside on any platform, linked to their parents and children on the same or another platform. Each platform involved in a mission has a TST manager accessible through ROS-based service calls, allowing communication and synchronization between nodes when necessary.

As illustrated in Fig. 37.15b, each *type* of node is associated with a set of *node parameters*. Together, these parameters are called the *node interface*. The interface always contains a platform (agent) assignment parameter, usually denoted by P_i , which identifies the agent responsible for executing the task. There are also always two parameters for the start and end times of the task, usually denoted by T_{Si} and T_{Ei} , respectively. Tasks often have additional type-specific parameters, such as a speed parameter and an area parameter for a **scan-area** action node.

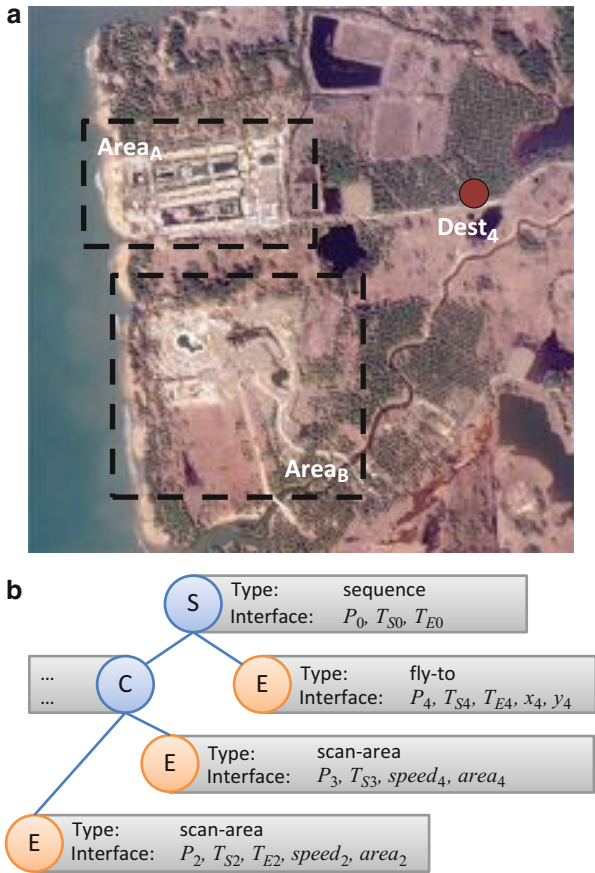


Fig. 37.15 Example mission. (a) Map of the mission area. (b) Example Task Specification Tree

When a concrete TST is specified, some node parameters may be given actual values through arguments. In the example, one may specify that the areas to scan are Area_A and Area_B (not shown in the figure). Other parameters may be left open. Concrete values may then be chosen by the executing agent, subject to a set of *constraints* associated with each node. These constraints can be used to constrain permitted values for the parameters of the same node as well as all parameters of ancestor nodes in the tree. By constraining temporal parameters, one can also express precedence relations and organizational relations between the nodes in the TST that are not implicitly captured by the use of specific control nodes. Together the constraints form a constraint network where the node parameters function as constraint variables.

Note that constraining node parameters implicitly constrains the degree of autonomy of an agent, as it reduces the space of possibilities that the agent can choose from. Also, both human and robotic agents may take the initiative and set

```

TST      ::= NAME '(' VARS ') ' '=' (with VARS)? TASK (where CONS)?
TSTS     ::= TST | TST ';' TSTS
TASK     ::= ACTION | GOAL | call NAME '(' ARGS ') ' | sequence TSTS | concurrent TSTS |
           if [VAR] COND then TST else TST | while [VAR] COND TST |
           foreach VARS where [VAR] COND do conc TST
VAR      ::= <variable name>
VARS     ::= VAR | VAR ',' VARS
ARG      ::= VAR | VALUE
ARGS     ::= ARG | ARG ',' ARGS
CONS     ::= <constraint> | <constraint> and CONS
VALUE    ::= <value>
NAME     ::= <node name>
COND     ::= <FIPA ACL query message requesting the value of a boolean expression>
GOAL     ::= <goal statement name( $\bar{x}$ )>
ACTION   ::= <elementary action call name( $\bar{x}$ )>

```

Fig. 37.16 Task Specification Tree language

the values of the parameters in a TST before or during execution. This provides support for one form of mixed-initiative interaction.

37.4.3.2 Task Specification Tree Language

For convenience, a text-based language for representing Task Specification Trees has also been defined (Fig. 37.16). The TST construct corresponds to a specific parameterized node and introduces the main recursive pattern. All such nodes must be explicitly named in order to allow name-based references. The parameters provided after the name specify the node interface which can be accessed from the outside. These can be constrained relative to each other using constraints in a **where** clause.

The formal semantics of a TST is specified through a translation from this language into composite actions in Temporal Action Logic (Doherty and Kvarnström 2008; Doherty et al. 2012). This provides a means of formally verifying the behavior of tasks and is also useful as a debugging tool.

Example 37.2. Consider again the TST depicted in Fig. 37.15b. This TST contains two composite actions, *sequence* (here named τ_0) and *concurrent* (τ_1), and two elementary actions, *scan* (τ_2 , τ_3) and *flyto* (τ_4).

Nodes in this TST have the task names τ_0 – τ_4 associated with them.

```

 $\tau_0(T_{S_0}, T_{E_0}) = \text{with } T_{S_1}, T_{E_1}, T_{S_4}, T_{E_4} \text{ sequence }$ 
 $\tau_1(T_{S_1}, T_{E_1}) = \text{with } T_{S_2}, T_{E_2}, T_{S_3}, T_{E_3} \text{ concurrent }$ 
 $\tau_2(T_{S_2}, T_{E_2}) = \text{scan-area}(T_{S_2}, T_{E_2}, \text{Speed}_2, \text{Area}_A);$ 
 $\tau_3(T_{S_3}, T_{E_3}) = \text{scan-area}(T_{S_3}, T_{E_3}, \text{Speed}_3, \text{Area}_B)$ 
where  $\text{cons}_{\tau_1};$ 
 $\tau_4(T_{S_4}, T_{E_4}) = \text{fly-to}(T_{S_4}, T_{E_4}, \text{Speed}_4, \text{Dest}_4)$ 
where  $\text{cons}_{\tau_0}$ 

```

$$\begin{aligned} cons_{\tau_0} &= T_{S_0} \leq T_{S_1} \wedge T_{S_1} < T_{E_1} \wedge T_{E_1} \leq T_{S_4} \wedge T_{S_4} < T_{E_4} \wedge T_{E_4} \leq T_{E_0} \\ cons_{\tau_1} &= T_{S_1} \leq T_{S_2} \wedge T_{S_2} < T_{E_2} \wedge T_{E_2} \leq T_{E_1} \wedge T_{S_1} \leq T_{S_3} \wedge T_{S_3} < T_{E_3} \wedge T_{E_3} \leq T_{E_1} \end{aligned}$$

The use of TSTs which call path planners is described in the next section. Additional features of TSTs and their relation to Temporal Action Logic and high-level mission specifications are described in Sect. 37.7.4.

37.5 The Navigation Subsystem

Many of the missions where AVs are deployed require sophisticated path-planning capability. One might need to fly directly to a particular waypoint, or in more complex scenarios, one might require the generation of a segmented path which is guaranteed to be collision-free. Many of the TSTs used in actual missions are required to call a motion planner which then outputs such segmented paths. The TST is then expanded with this output and executed. The navigation subsystem is responsible for this complex combination of processes which begin with a call to a motion planner at the deliberative layer. The output of the motion planner and its execution involve numerous calls to the Platform Server through use of FCL commands which in turn initiate execution of appropriate HCSMs. These in turn use the continuous control modes implemented in the Control Kernel.

This section provides a detailed description of the processes involved in the generation and execution of motion plans on the UASTL RMAX system. Additionally, it describes two path-planning algorithms integrated in the HDRC3 architecture based on extensions (Pettersson 2006; Pettersson and Doherty 2006) to two sample-based planning methods: Probabilistic RoadMaps (PRM, Kavraki et al. 1996) and Rapidly Exploring Random Trees (RRT, Kuffner and LaValle 2000).

Navigation Scheme. The standard navigation scheme in the HDRC3 architecture, assuming static operational environments, is depicted in Fig. 37.17. The task of flying is represented through an action node in a TST. For example, the task of flying to a specific waypoint can be represented declaratively as an elementary action node of type **fly-to**. The executor for nodes of this type calls a path planner (step 1 in the figure) that takes a map of static and dynamic obstacles together with the initial position, goal position, desired velocity, and possibly a set of additional constraints. The path planner then generates a segmented path (see Sect. 37.5.1) which is represented as a sequence of cubic polynomial curves. The generated path is collision free relative to the world model provided by a *geographical information system* (GIS Service). Each segment of the path is defined by start and end points, start and end directions, target velocity, and end velocity. If successful, this segmented path is returned to the TST node executor (step 2).

Once a segmented path is returned, the TST node executor requires a suitable interface to the low-level path-following control mode (PFCM). The PFCM implements continuous control laws which place certain real-time requirements on the way they are used, and the mode execution is therefore coordinated by HCSM state

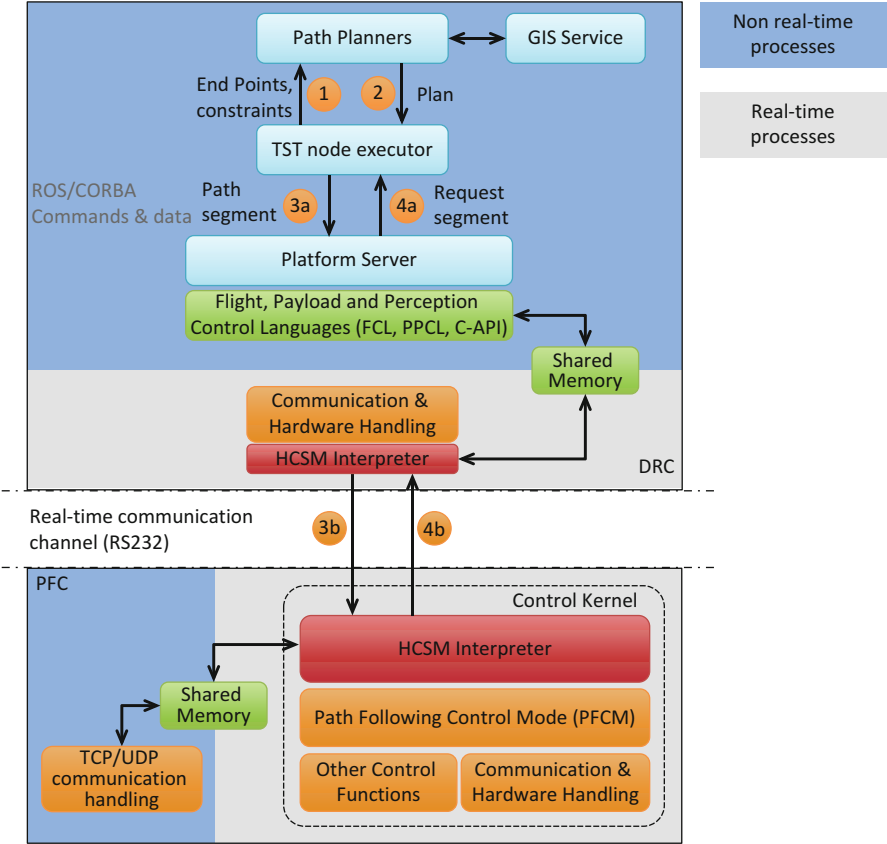


Fig. 37.17 Navigation subsystem and main software components

machines. The TST node executor therefore sends the first segment of the path via the Platform Server (step 3) and waits for a *Request Segment* event to be returned. This event is generated by the HCSM responsible for the path execution as soon as the PFCM controller receives a path segment as input. Details will be discussed in Sect. 37.5.2.

When a *Request Segment* event arrives (step 4), the TST node executor sends the description of the next segment to the HCSM that coordinates path execution at the control level. This procedure is repeated (steps 3–4) until the last segment is executed. However, because the high-level system is not implemented in hard real time, it may happen that the next segment does not arrive at the Control Kernel on time. In this case, the controller has a timeout limit after which it goes into safety braking mode in order to stop and hover at the end of the current segment. The timeout is determined by a velocity profile dynamically generated for the path segment together with the current position and current velocity.

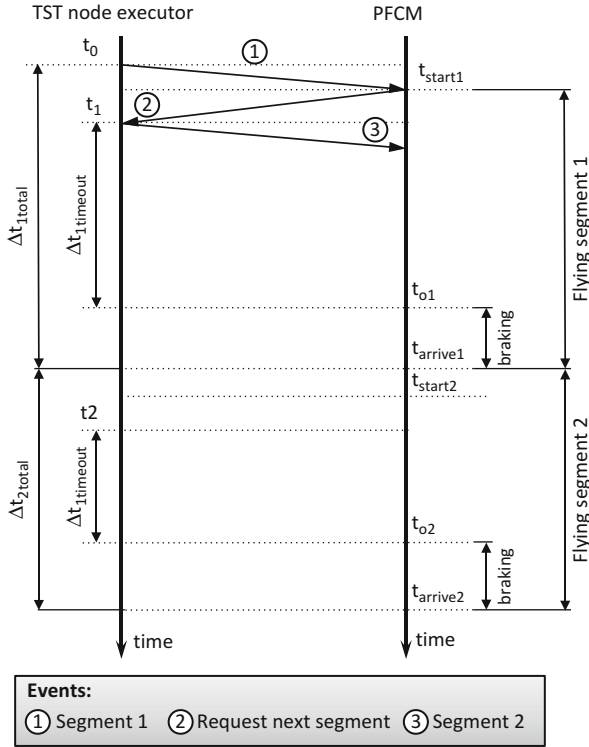


Fig. 37.18 Execution time line for a path consisting of two segments

Figure 37.18 depicts a timeline plot of the execution of a 2-segment trajectory. At time t_0 , a TST node executor sends the first segment of the path to the PFCM controller and waits for a *Request Segment* event which arrives immediately (t_1) after the helicopter starts to fly (t_{start1}). Typical time values for receiving a *Request Segment* event ($t_1 - t_0$) are well below 200 ms. Time t_{o1} is the timeout for the first segment which means that the TST node executor has a $\Delta t_{1timeout}$ time window to send the next segment to the PFCM controller before it initiates a safety braking procedure. If the segment is sent after t_{o1} , the helicopter will start braking. In practice, the $\Delta t_{1timeout}$ time window is large enough to replan the path using the standard path planner (Sect. 37.5.1). The updated segments are then sent to the PFCM controller transparently.

The path execution mechanism described here allows for dynamic replacement of path segments if necessary. This is an important part of the architecture due to the fact that it is assumed that higher-level deliberative components are continually planning, executing, and monitoring missions which are likely to change due to contingencies. This architectural solution supports a certain type of any-time behavior in the system which takes into account resource constraints when reconfiguring plans (Wzorek 2011).

37.5.1 Path and Motion Planning

Path-planning and motion planning algorithms deal with the problem of generating collision-free paths for a robot in order to navigate or move in an physical space, called the *workspace* \mathcal{W} . The workspace is most often modeled as \mathcal{R}^3 but can be restricted to \mathcal{R}^2 for robots navigating in a single plane. This type of representation is particularly well suited for collision checking since the robot and the obstacles are represented in the same space. However, in many practical applications, the workspace is not sufficient to describe the planning problem, and a more expressive representation is required.

The *configuration space* (C or C -space) is defined as a vector space or manifold of configurations q , where a configuration is a set of parameters that uniquely defines the location of all points of the robot in the workspace \mathcal{W} . For a rigid-body robot such as an AV platform, this would include not only its position but also its orientation. Additionally, not all robot configurations are attainable due to obstacle constraints. The *free space* denoted by C_{free} is a subset of the C -space that is free from collisions with obstacles.

When dealing with robotic systems in motion, the configuration of the robot is insufficient to describe the problem: The dynamic state of a robot (its velocity) also has to be accounted for. The *state-space* representation extends the configuration space by adding first-order derivatives \dot{q} of the robot configuration q . Thus, for a robot configuration $q = (q_0, \dots, q_n)$, the state x is defined by $x = \langle q, \dot{q} \rangle$ where $\dot{q} = (\dot{q}_0, \dots, \dot{q}_n)^T$.

Constraints. In addition to the requirement of avoiding collisions, plans must also satisfy *kinematic* and *dynamic* constraints. Kinematic constraints include only first-order derivatives of the configuration parameters, while second-order derivatives such as acceleration are allowed in the dynamic constraints. The algorithms presented below handle the kinematic and dynamic constraints of the UASTL RMAX platforms.

Both of these types of constraints belong to a class of nonholonomic constraints (also called motion constraints) and are common for many types of robots. A robot is nonholonomic if it has fewer controllable degrees of freedom than total degrees of freedom. A car is nonholonomic since it can only drive forward or backward, not sideways. So is a helicopter: Though it can move freely in any direction, its freedom of movement depends on its speed. When a helicopter is hovering or flying slowly, it could be considered to be holonomic, but this would constrain its usage.

Path Planning. The path-planning problem is defined as finding a path in C_{free} that connects the start (q_0) and the goal (q_g) configuration. For high-dimensional configuration spaces, this problem is intractable in general which has led to the development of sample-based methods such as PRMs (Kavraki et al. 1996) and RRTs (Kuffner and LaValle 2000). These use an approximation of the C_{free} continuous space (in configuration space or state space) in order to deal with

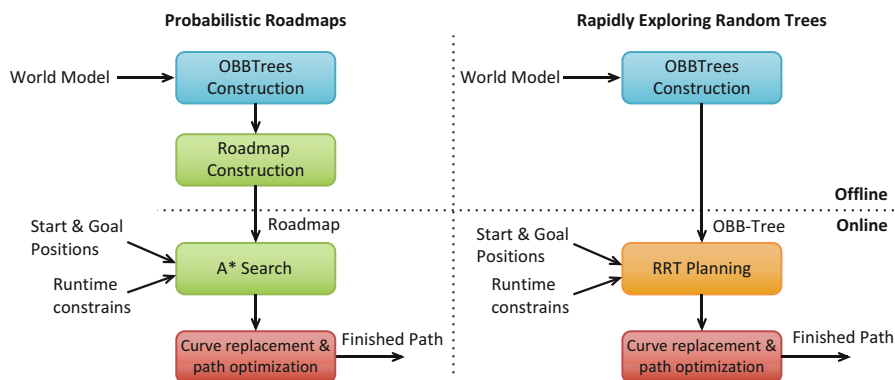


Fig. 37.19 PRM and RRT path-plan generation

the complexity of high-dimensional problems. The discrete representation of the original continuous space (typically represented in the form of a graph) sacrifices strict completeness for a weaker definition such as *resolution completeness* or *probabilistic completeness* (LaValle 2004).

37.5.1.1 Probabilistic Roadmaps and Rapidly Exploring Random Trees

The standard probabilistic roadmap (PRM) algorithm (Kavraki et al. 1996) works in two phases, one offline and the other online. In the offline phase, a discrete roadmap representing the free configuration space is generated using a 3D world model. First, it randomly generates a number of configurations and checks for collisions with the world model. A local path planner is then used to connect collision-free configurations taking into account kinematic and dynamic constraints of the helicopter. Paths between two configurations are also checked for collisions. This results in a roadmap approximating the configuration free space. In the online or querying phase, start and goal configurations are provided, and an attempt is made to connect each configuration to the previously generated roadmap using a local path planner. Finally, a graph search algorithm is used to find a path from the start configuration to the goal configuration in the augmented roadmap.

Figure 37.19 provides a schema of the PRM path planner that is used. The planner uses an Oriented Bounding Box Trees (OBBTrees) algorithm (Gottschalk et al. 1996) for collision checking and an A* algorithm for graph search. Here one can optimize for various criteria such as shortest path and minimal fuel usage.

The mean planning time in the current implementation for a selected set of flight test environments is below 1,000 ms, and the use of runtime constraints (discussed below) does not noticeably influence the mean. See Pettersson (2006) and Pettersson and Doherty (2006) for a detailed description of the modified PRM planner.

Rapidly exploring random trees (RRT, Kuffner and LaValle 2000) provide an efficient motion planning algorithm that constructs a roadmap online rather than offline (Fig. 37.19). The RRT algorithm generates two trees rooted in the start and goal configurations by exploring the configuration space randomly in both directions. While the trees are being generated, an attempt is made at specific intervals to connect them to create one roadmap. After the roadmap is created, the remaining steps in the algorithm are the same as with PRMs. The mean planning time with RRT is also below 1,000 ms, but the success rate is considerably lower, and the generated plans may sometimes cause anomalous detours (Pettersson 2006). The UASTL RMAX system uses both the PRM and RRT planners individually and in combination.

37.5.1.2 Path-Planner Extensions

The standard PRM and RRT algorithms are formulated for fully controllable systems only. This assumption is true for a helicopter flying at low speed with the capability to stop and hover at each waypoint. However, when the speed is increased, the helicopter is no longer able to negotiate turns of a smaller radius. This in turn imposes demands on the planner similar to nonholonomic constraints for car-like robots.

The most straightforward way of handling dynamic constraints in the PRM and RRT algorithms is to complement the configurations with their derivatives and record the complete state at each node in the graph. This enables the local path planner to adapt the path between two nodes to their associated derivatives, which is necessary to respect the dynamic constraints at boundary points between adjacent edges in the solution path. The drawback of this approach is that the dimensionality of the space in which the roadmap/tree is situated is doubled thus increasing the complexity of the problem. An alternative approach to nonholonomic planning is to postpone the nonholonomic constraints to the runtime phase. The following extensions have therefore been made with respect to the standard version of the PRM and RRT algorithms.

Multilevel Roadmap/Tree Planning. Inspired by a multilevel planner proposed by Sekhavat et al. (1996), new planners for AV applications have been developed (Pettersson 2006; Pettersson and Doherty 2006). In this approach, linear paths are first used to connect configurations in the graph/tree, and at a later stage, these are replaced with cubic curves when possible (Fig. 37.20). These are required for smooth high-speed flight. If it is not possible to replace a linear path segment with a cubic curve, then the helicopter has to slow down and switch to hovering mode at the connecting waypoint before continuing. This rarely happens in practice.

The random sampling nature of these planners and the limited density of the sampled nodes make the PRM and RRT algorithms produce paths that are often jagged and irregular with occasional detours. In order to improve the quality of the paths, a smoothing step is usually added. For the implemented path planners, the following smoothing steps are performed:

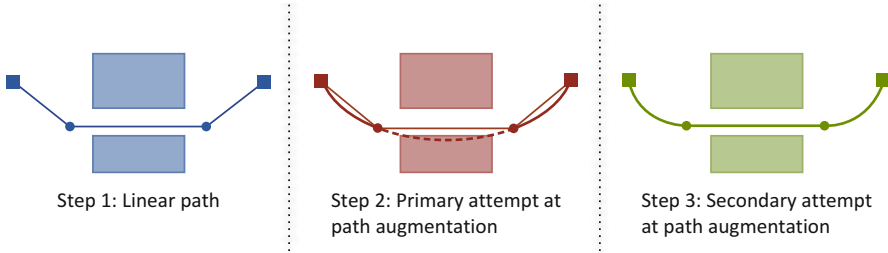


Fig. 37.20 Transformation from linear to cubic path segments for smooth flight

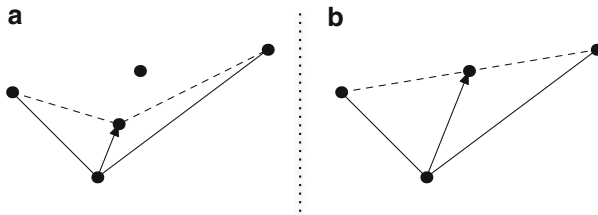


Fig. 37.21 Alignment of nodes for improved path quality. (a) Alignment halfway. (b) Full alignment

- **Node Alignment:** For each node n along the path, two attempts are made to move it to a point that straightens out the path (Fig. 37.21). The point m in the middle between the two neighbors is located. First, an attempt is made to move n halfway to m (Fig. 37.21a), if this is possible given known obstacles. Then an attempt is made to move it all the way to m (Fig. 37.21b).
- **Node Elimination:** For each node along the path, an attempt is made to eliminate it by connecting the two adjacent nodes directly. If the connection satisfies all constraints, the middle node is eliminated.

The curve replacement step described above is performed between the alignment and the elimination step.

Runtime Constraint Handling. Some constraints are not available during roadmap or tree construction but are added at runtime, when a final plan is requested. The motion planner currently handles the following types of runtime constraints during the A* search phase:

- **Maximum and minimum altitude** – verified through an intersection test between the configuration or curve and a horizontal plane.
- **Forbidden regions (no-fly zones)** – regions created by a set of horizontal polygons covering an area which the AV must not enter. The satisfaction test for configurations involves checking if the AV's position, projected in the X/Y plane, is within the polygon. The test for curves includes checking if the curve intersects any of the planes bounding the polygon in the 3D-space.
- **Limits on the ascent and descent rate.**

Adding constraints to a path-planning problem may break the connectivity of the PRM roadmap. In this case a combination of the PRM and RRT algorithms is used to mitigate the problem. An optional reconnection attempt using the RRT planner is made during the query phase of the PRM planner in order to reconnect the broken roadmap. Both planners (PRM and RRT) and their combination with the presented extensions are accessible to the deliberative and reactive services in the HDRC3 architecture.

37.5.2 HCSMs for Path Execution

The use case described in this section assumes the UASTL RMAX system is already in autonomous flight mode and the default hovering function is active (e.g., after executing the use case presented in Sect. 37.4.1). The description focuses on the execution of the path at the lowest control level running on the PFC computer after a TST node executor has received a path plan (a segmented cubic polynomial curve) from the path planner as described at the beginning of previous section. For better understanding of the interactions between the HCSM automata, the focus will be on the execution of one path segment.

As previously stated, the TST node executor sends the first segment (step 3 in Fig. 37.17, Traj3D-EV) of the trajectory via the *Platform Server* and waits for a Request-Segment-EV event that is generated by the controller. At the control level, the path is executed using the path-following control mode (PFCM, described in Sect. 37.3.2). When a Request-Segment-EV event arrives (step 4), the TST node executor sends the next segment. This procedure (steps 3–4) is repeated until the last segment is sent. However, because the high-level system is not implemented in hard real time, it may happen that the next segment does not arrive at the Control Kernel on time. In this case, the controller has a timeout limit after which it goes into safety braking mode in order to stop and hover at the end of the current segment.

As described in Sect. 37.4.1, the implementation of the PFCM function follows a predefined design template. The function itself is executed by setting an appropriate internal flag using a SET action (SET Traj3D-Mode-AC, B.3) by the *Control Mode* automaton. *Mode Switch* makes sure the default hovering function is properly terminated before the PFCM function can be activated. Additionally, when path execution is finished, it engages the default hovering function, in a manner similar to the example shown previously.

The role of the *Traj3D* automaton (Fig. 37.22) is to coordinate an exchange of events with other services (such as TST node executors) and to ensure that the appropriate segment data is available to the PFCM control function when needed. An example time line for a path execution showing the interaction between the three automata is presented in Fig. 37.23. The example focuses on the execution of a single segment.

The *Mode Switch* automaton starts in the Hovering-ST state. After receiving a Traj3D-EV event from the DRC computer, the data describing a segment for

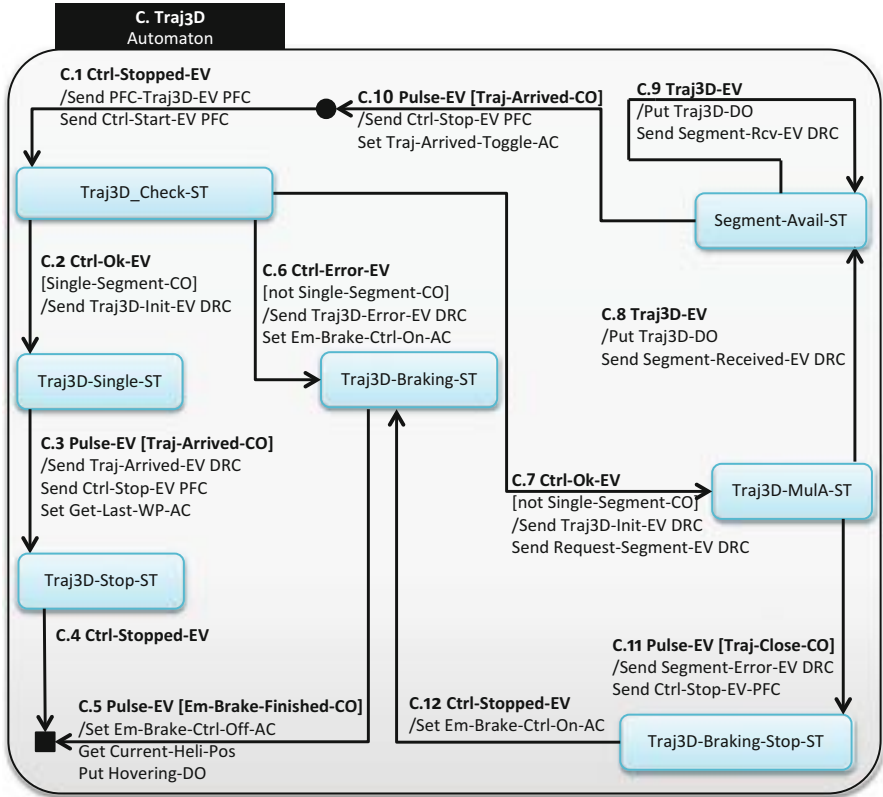


Fig. 37.22 The *Traj3D* automaton

execution is saved in the memory slot used by the PFCM function (Traj3D-DO). Additionally, the Ctrl-Stop-EV event (A.6) is sent to the *Control Mode* automaton in order to stop the execution of the default hovering function. At this time, *Mode Switch* transitions to the Traj3D-ST state (Fig. 37.22), thereby starting the *Traj3D* automaton in its *init* state. *Mode Switch* remains in Traj3D-ST until the PFCM execution is finished and then switches to the default hovering mode.

When *Traj3D* starts, it waits for confirmation from *Control Mode* that the hovering function has been terminated (Ctrl-Stopped-EV, B.10) and sends two events back to initialize and start the execution of the PFCM control function (PFC-Traj3D-EV (C.1) and Ctrl-Start-EV (C.1)). It transitions to the Traj3D-Check-ST state waiting for confirmation that the PFCM function has been initialized (Ctrl-Ok-EV, B.8). When the event arrives, the HCSM checks whether a single segment has been received (Single-Segment-CO, C.3). A single segment is defined by the path parameters with the end velocity for the segment set to zero. If the condition is satisfied, the event informing the DRC computer that the segment has been accepted for execution is sent.

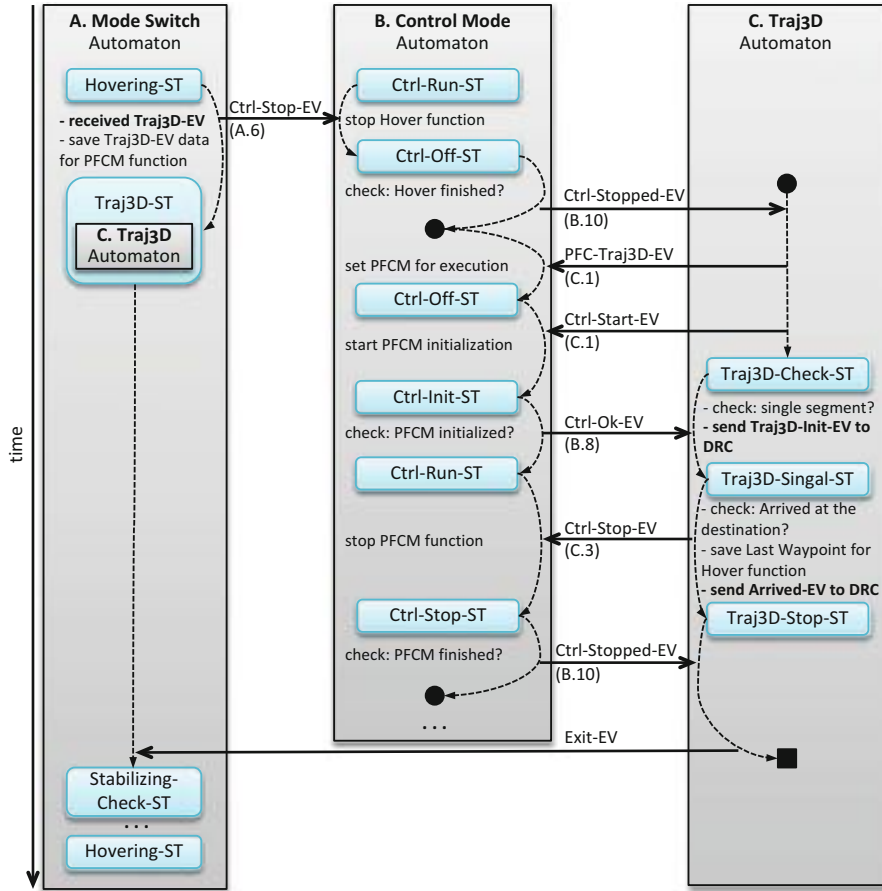


Fig. 37.23 Execution trace: The interaction between *Control Mode*, *Mode Switch*, and *Traj3D*

At this time the PFCM starts the execution of the segment. The *Traj3D* automaton is in the *Traj3D-Single-ST* state and remains there until the *Traj-Arrived-CO* condition is not satisfied. On the UASTL RMAX, the segment has been successfully executed when the distance to the final waypoint is less than 3 m. At that point, the last waypoint of the path is saved for the hovering function, and the *Traj-Arrived-EV* event is sent to the DRC computer informing it that the path execution is finished. Additionally, the *Traj3D* automaton stops the execution of the PFCM function by sending the *Ctrl-Stop-EV* event (C.3) to the *Control Mode* state machine. When the termination of the execution is confirmed by receiving the *Ctrl-Stopped-EV* event (B.10), the *Traj3D* automaton transitions to its exit state, and the *Mode Switch* state machine takes care of engaging the default hover mode, in a manner similar to the example described previously (Sect. 37.4.1.1).

37.6 DyKnow: Stream-Based Reasoning Middleware

For autonomous unmanned aerial systems to successfully perform complex missions, a great deal of embedded reasoning is required. For this reasoning to be grounded in the environment, it must be firmly based on information gathered through available sensors. However, there is a wide gap in abstraction levels between the noisy numerical data directly generated by most sensors and the crisp symbolic information that many reasoning functionalities assume to be available. This is commonly called the *sense-reasoning gap*.

Bridging this gap is a prerequisite for deliberative reasoning functionalities such as planning, execution monitoring, and diagnosis to be able to reason about the current development of dynamic and incompletely known environments using representations grounded through sensing. For example, when monitoring the execution of a plan, it is necessary to continually collect information from the environment to reason about whether the plan has the intended effects as specified in a symbolic high-level description.

Creating a suitable bridge is a challenging problem. It requires constructing representations of information incrementally extracted from the environment. This information must continuously be processed to generate information at increasing levels of abstraction while maintaining the necessary correlation between the generated information and the environment itself. The construction typically requires a combination of a wide variety of methods, including standard functionalities such as signal and image processing, state estimation, and information fusion.

These and other forms of reasoning about information and knowledge have traditionally taken place in tightly coupled architectures on single computers. The current trend towards more heterogeneous, loosely coupled, and distributed systems necessitates new methods for connecting sensors, databases, components responsible for fusing and refining information, and components that reason about the system and the environment. This trend makes it less practical to statically predefine exactly how the information processing should be configured. Instead it is necessary to configure the way in which information and knowledge is processed and reasoned about in a context-dependent manner relative to high-level goals while globally optimizing the use of resources and the quality of the results.

To address these issues, the stream-based reasoning middleware DyKnow (Heintz and Doherty 2004; Heintz 2009; Heintz et al. 2010) has been developed. This is a central part of the HDRC3 architecture as shown in Fig. 37.1.

37.6.1 DyKnow

The main purpose of DyKnow is to provide generic and well-structured software support for the processes involved in generating state, object, and event abstractions about the environments of complex systems. The generation is done

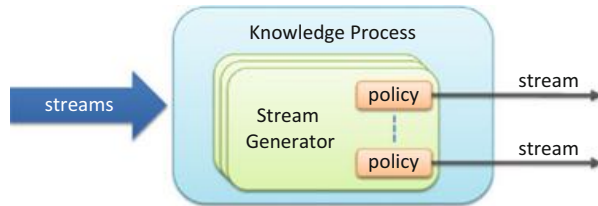
at many levels of abstraction beginning with low-level quantitative sensor data and resulting in qualitative data structures which are grounded in the world and can be interpreted as knowledge by the system. To produce these structures, DyKnow supports operations on streams at many different levels of abstraction. For the result to be useful, the processing must be done in a timely manner so that a UAS can react in time to changes in the environment. The resulting structures are used by various functionalities in the HDRC3 architecture for situation awareness and assessment (Heintz et al. 2007), planning to achieve mission goals (Sect. 37.7.2), and monitoring (Sect. 37.7.3). DyKnow provides a declarative language for specifying the structures needed by the different subsystems. Based on this specification, it creates representations of the external world and the internal state of an AV based on observations and a priori knowledge, such as facts stored in databases.

DyKnow helps organize the many levels of information and knowledge processing in a distributed robotic system as a coherent network of processes connected by streams. The streams contain time-stamped information and may be viewed as representations of time-series data which may start as continuous streams from sensors or sequences of queries to databases. Eventually, they will contribute to more refined, composite, knowledge structures. Knowledge producing processes combine streams by applying functions, synchronization, filtering, aggregation, and approximation as they move to higher levels of abstraction. In this sense, DyKnow supports conventional data fusion processes, but also less conventional qualitative processing techniques common in the area of artificial intelligence. In Heintz and Doherty (2006), it is argued that DyKnow supports all the functional abstraction levels in the JDL Data Fusion Model (White 1988).

A knowledge process has different quality of service properties such as maximum delay, trade-off between quality and delay, and how to calculate missing values, which together define the semantics of the knowledge derived by the process. It is important to realize that knowledge is not static, but is a continually evolving collection of structures which are updated as new information becomes available from sensors and other sources. Therefore, the emphasis is on the continuous and ongoing knowledge derivation process, which can be monitored and influenced at runtime. The same streams of data may be processed differently by different parts of the architecture by tailoring the knowledge processes relative to the needs and constraints associated with the tasks at hand. This allows DyKnow to support easy integration of existing sensors, databases, reasoning engines, and other knowledge producing services.

A knowledge processing application in DyKnow consists of a set of *knowledge processes* (Fig. 37.24) connected by *streams* satisfying *policies*. A policy is a declarative specification of the desired properties of a stream. Each knowledge process is an instantiation of a *source* or a *computational unit* providing *stream generators* that produce streams. A source makes external information, such as the data provided by the Data Interface in the Platform Server (Sect. 37.4.2.1), available in the form of streams while a computational unit refines and processes streams.

Fig. 37.24 A prototypical knowledge process



37.6.2 Streams and Policies

Knowledge processing for a physical agent is fundamentally incremental in nature. Each part and functionality in the system, from sensing to deliberation, needs to receive relevant information about the environment with minimal delay and send processed information to interested parties as quickly as possible. Rather than using polling, explicit requests, or similar techniques, the strategy is to model and implement the required flow of data, information, and knowledge in terms of *streams* while computations are modeled as active and sustained *knowledge processes* ranging in complexity from simple adaptation of raw sensor data to complex reactive and deliberative processes.

Streams lend themselves easily to a *publish/subscribe* architecture. Information generated by a knowledge process is published using one or more *stream generators* each of which has a (possibly structured) *label* serving as an identifier within a knowledge processing application. Knowledge processes interested in a particular stream of information can subscribe to it using the label of the associated stream generator which creates a new stream without the need for explicit knowledge of which process hosts the generator. Information produced by a process is immediately provided to the stream generator, which asynchronously delivers it to all subscribers, leaving the knowledge process free to continue its work. Using an asynchronous publish/subscribe pattern of communication decouples knowledge processes in time, space, and synchronization, providing a solid foundation for distributed knowledge processing applications.

Each stream is associated with a declarative *policy*, a set of requirements on its contents. Such requirements may include the fact that elements must arrive ordered by valid time, that each value must constitute a significant change relative to the previous value, that updates should be sent with a specific sample frequency, or that there is a maximum permitted delay. Policies can also give advice on how these requirements should be satisfied, for example, by indicating how to handle missing or excessively delayed values.

37.6.3 Objects, Features, and Fluent Streams

For modeling purposes, the environment of an AV is viewed as consisting of physical and nonphysical *objects*, *properties* associated with these objects, and

relations between these objects. The properties and relations associated with objects are called *features*. Due to the potentially dynamic nature of a feature, that is, its ability to change values through time, a total function from time to value called a *fluent* is associated with each feature. It is this fluent, representing the value over time of a feature, which is being modeled. Example objects are *the AV*, *car37*, and *the entity observed by the camera*. Some examples of features are the *velocity* of an object, the *road segment* of a vehicle, and the *distance between* two car objects.

A *fluent stream* is a partial representation of a fluent, where a stream of *samples* of the value of the feature at specific timepoints is seen as an approximation of the fluent. Due to inherent limitations in sensing and processing, an agent cannot always expect access to the actual value of a feature over time. A sample can either come from an observation of the feature or a computation which results in an estimation of the value at the particular timepoint, called the *valid time*. The timepoint when a sample is made available or added to a fluent stream is called the *available time*. A fluent stream has certain properties such as start and end time, sample period, and maximum delay. These properties are specified by a declarative policy which describes constraints on the fluent stream.

For example, the position of a car can be modeled as a feature. The true position of the car at each timepoint during its existence would be its fluent and a particular sequence of observations or estimations of its position would be a fluent stream. There can be many fluent streams all approximating the same fluent.

37.6.4 State Generation

One important functionality in DyKnow is state generation. Many functionalities require access to a consistent “state of the world,” but sensor readings take time to propagate through a distributed system which may consist of multiple AVs together with ground stations and associated hardware and software. DyKnow therefore provides services for data synchronization, generating a best approximation of the state at a given point in time using the information that has propagated through the distributed system so far.

For example, if a car has both a speed and a position, then there are two features: “speed of car” and “position of car.” But this could also be represented by a single fluent stream containing *tuples* (in this case pairs) of values, called *states*, containing both the speed and the position.

37.6.5 Semantic Integration of Symbolic Reasoning

One important use of DyKnow is to support the integration of symbolic reasoning in a UAS. To do symbolic reasoning, it is necessary to map symbols to streams available in a UAS, which provides them with the intended meaning for the particular UAS.

For example, a temporal logic formula consists of symbols representing variables, sorts, objects, features, and predicates besides the symbols which are part of the logic. Consider $\forall x \in AV : x \neq av1 \rightarrow \Box XYDist[x, av1] > 10$, which has the intended meaning that all AVs, except *av1*, should always be more than 10m away from *av1*. This formula contains the variable *x*, the sort AV, the object *av1*, the feature XYDist, the predicates \neq and $>$, and the constant value 10, besides the logical symbols. To evaluate such a formula (see Sect. 37.7.3 for details), an interpretation of its symbols must be given. Normally, their meanings are predefined. However, in the case of reasoning over streams, the meaning of features cannot be predefined since information about them becomes incrementally available. Instead their meaning has to be determined at runtime. To evaluate the truth value of a formula, it is therefore necessary to map feature symbols to streams, synchronize these streams, and extract a state sequence where each state assigns a value to each feature.

In a system consisting of streams, a natural approach is to syntactically map each feature to a single stream. This is called *syntactic integration*. This works well when there is a stream for each feature and the person writing the formula is aware of the meaning of each stream in the system. However, when systems become more complex and when the set of streams or their meaning changes over time, it is much harder for a designer to explicitly state and maintain this mapping. Therefore, automatic support for mapping features in a formula to streams in a system based on their semantics is needed. This is called *semantic integration*. The purpose of this matching is for each feature to find one or more streams whose content matches the intended meaning of the feature. This is a form of semantic matching between features and contents of streams. The process of matching features to streams in a system requires that the meaning of the content of the streams is represented and that this representation can be used for matching the intended meaning of features with the actual content of streams.

The same approach can be used for symbols referring to objects and sorts. It is important to note that the semantics of the logic requires the set of objects to be fixed. This means that the meaning of an object or a sort must be determined for a formula before it is evaluated and then may not change. It is still possible to have different instances of the same formula with different interpretations of the sorts and objects.

The goal is to automate the process of matching the intended meaning of features, objects, and sorts to content of streams in a system. Therefore, the representation of the semantics of streams needs to be machine readable. This allows the system to reason about which stream content corresponds to which symbol in a logical formula. The knowledge about the meaning of the content of streams needs to be specified by a user, even though it could be possible to automatically determine this in the future. By assigning meaning to stream content, the streams do not have to use predetermined names, hard-coded in the system. This also makes the system domain independent, which implies that it could be used to solve different problems in a variety of domains without reprogramming.

The approach to semantic integration in DyKnow uses semantic web technologies to define and reason about ontologies. Ontologies provide suitable support for creating machine readable domain models (Horrocks 2008). Ontologies also provide reasoning support and support for semantic mapping which is necessary for the integration of streams from multiple UASs.

The Web Ontology Language (OWL) (Smith et al. 2004) is used to represent ontologies. Features, objects, and sorts are represented in an ontology with two different class hierarchies: one for objects and one for features.

To represent the semantic content of streams in terms of features, objects, and sorts, a semantic specification language called SSL_T has been defined (Heintz and Dragisic 2012). This is used to annotate the semantic content of streams.

Finally, a semantic matching algorithm has been developed which finds all streams which contain information relevant to a concept from the ontology, such as a feature. This makes it possible to automatically find all the streams that are relevant for evaluating a temporal logical formula. These streams can then be collected, fused, and synchronized into a single stream of states over which the truth value of the formula is incrementally evaluated. By introducing semantic mapping between ontologies from different UASs and reasoning over multiple related ontologies, it is even possible to find relevant streams distributed among multiple UASs (Heintz and Dragisic 2012).

37.6.6 ROS-Based Implementation

The ROS-based implementation of DyKnow consists of three main parts: a stream processing part, a stream reasoning part, and a semantic integration part. Each part consists of a set of components. There are three types of components: engines, managers, and coordinators. An *engine* takes a specification and carries out the processing as specified. A *manager* keeps track of related items and provides an interface to these. A *coordinator* provides a high-level functionality by coordinating or orchestrating other functionalities. An overview of the parts and the components is shown in Fig. 37.25. This diagram corresponds to the DyKnow component in Fig. 37.2. The design is very modular as almost every component can be used independently.

The *stream processing* part is responsible for generating streams by, for example, importing, merging, and transforming streams. The *Stream Manager* keeps track of all the streams in the system. Streams can either be generated by a stream processing engine or by some external program. A *stream processing engine* takes a stream specification and generates one or more streams according to the specification.

The *semantic integration* part is responsible for finding streams based on their semantics relative to a common ontology. The *Stream Semantics Manager* keeps track of semantically annotated streams, where an annotation describes the semantic content of a stream. The *Ontology Manager* keeps track of the ontology which provides a common vocabulary. The *Semantic Matching Engine* finds all streams whose semantic annotation matches a particular ontological concept. The semantic

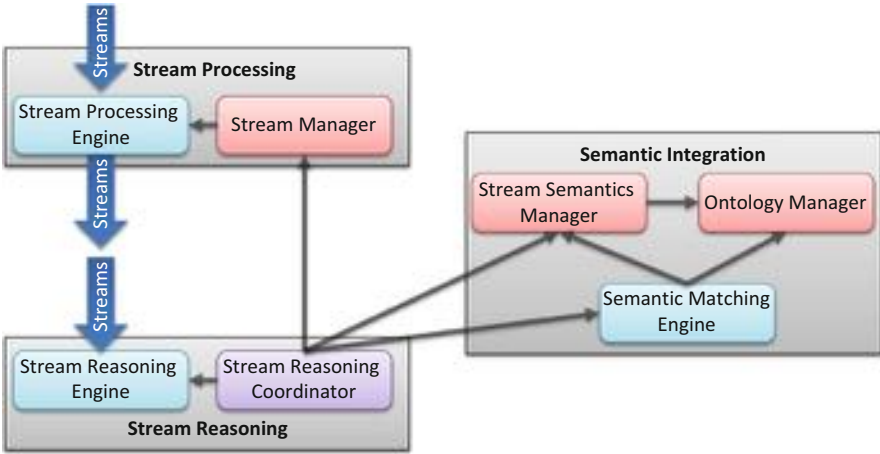


Fig. 37.25 The components of DyKnow

integration part is used by the stream reasoning part to find the relevant streams in order to evaluate a logical formula.

The *stream reasoning* part is responsible for evaluating temporal logical formulas over streams as described in Sect. 37.7.3. A *stream reasoning engine* takes a logical formula and a stream of states and evaluates the formula over this stream. A *Stream Reasoning Coordinator* takes a logical formula, finds all the relevant streams needed to evaluate the formula, creates a stream specification for generating a single stream of states from all the relevant streams, and instructs the stream reasoning engine to evaluate the formula over the stream as it is generated by a stream processing engine.

37.6.7 A Traffic Monitoring Example

As a concrete example of the use of DyKnow, Fig. 37.26 provides an overview of how part of the incremental processing required for a traffic surveillance task can be organized as a set of DyKnow knowledge processes.

At the lowest level, a *helicopter state estimation component* uses data from an *inertial measurement unit* (IMU) and a *global positioning system* (GPS) to determine the current position and attitude of the UASTL RMAX. A *camera state estimation component* uses this information, together with the current state of the *pan-tilt unit* on which the cameras are mounted, to generate information about the current camera state. The *image processing component* uses the camera state to determine where the camera is currently pointing. Video streams from the *color* and *thermal cameras* can then be analyzed in order to generate *vision percepts* representing hypotheses about moving and stationary physical entities, including their approximate positions and velocities. The data originating from the PFC and the PPC is provided by the Data Interface through the Platform Server (Sect. 37.4.2.1).

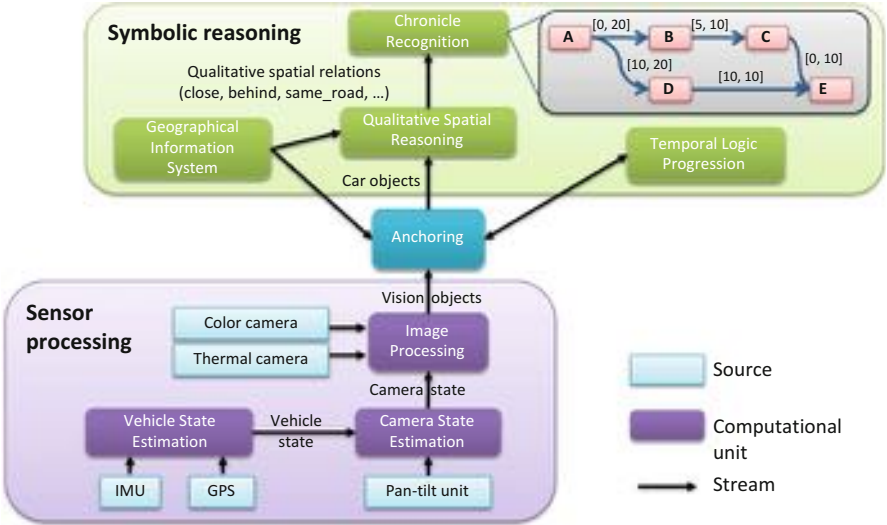


Fig. 37.26 Potential organization of the incremental processing required for a traffic surveillance task

Symbolic formalisms such as chronicle recognition (Ghallab 1996) require a consistent assignment of symbols, or identities, to the physical objects being reasoned about and the sensor data received about those objects. Image analysis may provide a partial solution, with vision percepts having symbolic identities that persist over short intervals of time. However, changing visual conditions or objects temporarily being out of view lead to problems that image analysis cannot (and should not) handle. This is the task of the *anchoring* system, which uses *progression* of formulas in a metric temporal logic to incrementally evaluate potential hypotheses about the observed objects (Heintz et al. 2013). The anchoring system also assists in object classification and in the extraction of higher-level attributes of an object. For example, a *geographical information system* can be used to determine whether an object is currently on a road or in a crossing. Such attributes can in turn be used to derive relations *between* objects, including *qualitative spatial relations* such as *beside*(*car*₁, *car*₂) and *close*(*car*₁, *car*₂). Concrete events corresponding to changes in such attributes and predicates finally provide sufficient information for the *chronicle recognition system* to determine when higher-level events such as reckless overtakes occur.

37.7 The Deliberative Layer

Conceptually, the deliberative layer includes all high-autonomy functionalities normally associated with rich world models and deep reasoning capabilities. Functionalities commonly associated with the deliberative layer are automated task

planners, motion planners, execution monitoring systems, diagnosis systems, and knowledge bases with inference engines, among others. The temporal latencies associated with the decision cycles for these functionalities are often far slower than for functionalities which exist in the reactive and control layers. The decision cycles for the different layers are necessarily asynchronous where each of the sets of functionalities run concurrently. The interfaces between the deliberative layer and the reactive and control layers are essential for transitioning output from deliberative functionality into useful processes at the lower layers of the architecture. In some sense, much of what happens is a form of dynamic compilation which results in transforming qualitative goal-directed assertions into actual actuation commands associated with the control layer of the architecture which contribute to the achievement of these goals.

A deliberative functionality, motion planning, and the dynamic compilation of its output have already been described in Sect. 37.5. This section will describe two additional deliberative functionalities central to high autonomy in the HDRC3 architecture. Section 37.7.2 describes an automated task-based planner, TALplanner, and its integration in the HDRC3 architecture. Section 37.7.3 describes a novel execution monitoring system based on specifying monitoring queries in terms of temporal logical formulas and evaluating these formulas online in real time. Due to the complex nature of these functionalities, an open research issue is how one might provide verification and validation techniques for the use and integration of these functionalities, in addition to the formal specification of high-level missions. The section therefore begins with a brief presentation of a Temporal Action Logic (TAL) which provides a formal basis for specifying the semantics of TSTs, TALplanner, and the execution monitor (Sect. 37.7.1). Additionally, in Sect. 37.7.4, it is shown that high-level missions can in fact be specified in terms of TSTs. Consequently, high-level missions are not only provided with a formal semantics, but due to the relation between these specifications and TSTs, there is a natural means of coupling declarative specification of missions with their procedural execution.

37.7.1 Temporal Action Logic

Temporal Action Logic, TAL, is a well-established non-monotonic logic for representing and reasoning about actions (Doherty et al. 1998; Doherty and Kvarnström 2008). This logic provides both clear intuitions and a formal semantics for a highly expressive class of action specifications which supports temporal aspects, concurrent execution, and incomplete knowledge about the environment and the effects of an action. Therefore it is a highly suitable basis for describing and reasoning the elementary actions used in realistic mission specifications. Here a limited subset of TAL is described, and the reader is referred to Doherty and Kvarnström (2008) for further details.

TAL provides an extensible macro language, $\mathcal{L}(\text{ND})$, that supports the knowledge engineer and allows reasoning problems to be specified at a higher abstraction level than plain logical formulas. The basic ontology includes parameterized features

$f(\bar{x})$ that have values v at specific timepoints t , denoted by $[t]f(\bar{x}) \hat{=} v$, or over intervals, $[t, t']f(\bar{x}) \hat{=} v$. Incomplete information can be specified using disjunctions of such facts. Parameterized actions can occur at specific intervals of time, denoted by $[t_1, t_2]A(\bar{x})$. To reassign a feature to a new value, an action uses the expression $R([t]f(\bar{x}) \hat{=} v)$. Again, disjunction can be used inside $R()$ to specify incomplete knowledge about the resulting value of a feature. The value of a feature at a timepoint is denoted by $value(t, f)$.

The logic is based on scenario specifications represented as *narratives* in $\mathcal{L}(\text{ND})$. Each narrative consists of a set of statements of specific types, including *action-type specifications* defining named actions with preconditions and effects. The basic structure, which can be elaborated considerably (Doherty and Kvarnström 2008), is as follows:

$$[t_1, t_2]A(\bar{v}) \rightsquigarrow (\Gamma_{pre}(t_1, \bar{v}) \implies \Gamma_{post}(t_1, t_2, \bar{v})) \wedge \Gamma_{cons}(t_1, t_2, \bar{v})$$

stating that if the action $A(\bar{v})$ is executed during the interval $[t_1, t_2]$, then given that its preconditions $\Gamma_{pre}(t_1, \bar{v})$ are satisfied, its effects, $\Gamma_{post}(t_1, t_2, \bar{v})$, will take place. Additionally, $\Gamma_{cons}(t_1, t_2, \bar{v})$ can be used to specify logical constraints associated with the action. For example, the following defines the elementary action **fly-to**: If a AV should fly to a new position (x', y') within the temporal interval $[t, t']$, it must initially have sufficient fuel. At the next timepoint $t + 1$, the AV will not be hovering, and in the interval between the start and the end of the action, the AV will arrive and its fuel level will decrease. Finally, there are two logical constraints bounding the possible duration of the flight action.

$$[t, t']\text{fly-to}(av, x', y') \rightsquigarrow$$

$$\begin{aligned} & [t] \text{fuel}(av) \geq \text{fuel-usage}(av, x(av), y(av), x', y') \rightarrow \\ & R([t + 1] \text{hovering}(av) \hat{=} \text{False}) \wedge \\ & R((t, t'] x(av) \hat{=} x') \wedge R((t, t'] y(av) \hat{=} y') \wedge \\ & R((t, t'] \text{fuel}(av) \hat{=} value(t, \text{fuel}(av) - \text{fuel-usage}(av, x(av), y(av), x', y'))) \wedge \\ & t' - t \geq value(t, \text{min-flight-time}(av, x(av), y(av), x', y')) \wedge \\ & t' - t \leq value(t, \text{max-flight-time}(av, x(av), y(av), x', y')) \end{aligned}$$

The translation function $Trans()$ translates $\mathcal{L}(\text{ND})$ expressions into $\mathcal{L}(\text{FL})$, a first-order logical language (Doherty and Kvarnström 2008). This provides a well-defined formal semantics for narratives in $\mathcal{L}(\text{ND})$. This separation between the macro language and the base logic makes TAL highly extensible. When adding new constructs to the formalism, new expression types are defined in $\mathcal{L}(\text{ND})$, and $Trans()$ is extended accordingly, generally with no extension required in the base language $\mathcal{L}(\text{FL})$.

The $\mathcal{L}(\text{FL})$ language is order-sorted, supporting both types and subtypes for features and values. This is also reflected in $\mathcal{L}(\text{ND})$, where one often assumes variable types are correlated to variable names – for example, av_3 implicitly ranges over AVs. There are a number of sorts for values \mathcal{V}_i , including the Boolean sort \mathcal{B} with the constants $\{\text{true}, \text{false}\}$. \mathcal{V} is a superset of all value sorts. There are a number of sorts for features \mathcal{F}_i , each one associated with a value sort

$dom(\mathcal{F}_i) = \mathcal{V}_j$ for some j . The sort \mathcal{F} is a superset of all fluent sorts. There is also an action sort \mathcal{A} and a temporal sort \mathcal{T} . Generally, t, t' will denote temporal variables, while $\tau, \tau', \tau_1, \dots$ are temporal terms. $\mathcal{L}(\text{FL})$ currently uses the following predicates, from which formulas can be defined inductively using standard rules, connectives, and quantifiers of first-order logic:

- *Holds*: $\mathcal{T} \times \mathcal{F} \times \mathcal{V}$, where *Holds*(t, f, v) expresses that a feature f has a value v at a timepoint t , corresponding to $[t] f \hat{=} v$ in $\mathcal{L}(\text{ND})$.
- *Occlude*: $\mathcal{T} \times \mathcal{F}$, where *Occlude*(t, f) expresses that a feature f is permitted to change values at time t . This is implicit in reassignment, $R([t] f \hat{=} v)$, in $\mathcal{L}(\text{ND})$.
- *Occurs*: $\mathcal{T} \times \mathcal{T} \times \mathcal{A}$, where *Occurs*(t_s, t_e, A) expresses that a certain action A occurs during the interval $[t_s, t_e]$. This corresponds to $[t_s, t_e]A$ in $\mathcal{L}(\text{ND})$.

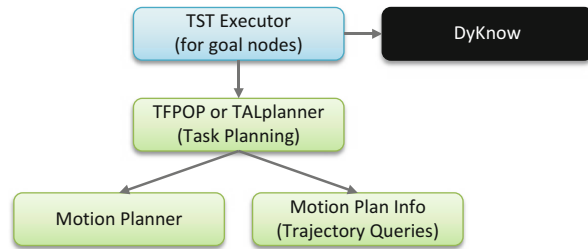
When a narrative is translated, *Trans*() first generates the appropriate $\mathcal{L}(\text{FL})$ formulas corresponding to each $\mathcal{L}(\text{ND})$ statement. Foundational axioms such as unique names and domain closure axioms are appended when required. Logical entailment then allows the reasoner to determine when actions must occur, but the fact that they *cannot* occur at other times than explicitly stated is not logically entailed by the translation. This problem is handled in a general manner through filtered circumscription, which also ensures that fluents can change values only when explicitly affected by an action or dependency constraint (Doherty and Kvarnström 2008).

The structure of $\mathcal{L}(\text{ND})$ statements ensures that the second-order circumscription axioms are reducible to equivalent first-order formulas, a reduction that can often be performed through predicate completion. Therefore, classical first-order theorem proving techniques can be used for reasoning about TAL narratives (Doherty and Kvarnström 2008). For unmanned systems, however, the logic will primarily be used to ensure a correct semantics for planners, execution monitors, and mission specification languages and correlate this semantics closely to the implementation. Using TAL does not require theorem proving onboard.

37.7.2 Task Planning Using TALplanner

When developing the architecture for a system capable of autonomous action execution and goal achievement, one can envision a spectrum of possibilities ranging from each behavior and task being explicitly coded into the system, regardless of complexity, up to the other extreme where the system itself generates complex solutions composed from a set of very primitive low-level actions. With the former end of the spectrum generally leading to more computationally efficient solutions and the latter end generally being far more flexible in the event of new and potentially unexpected tasks being tackled, the proper choice is usually somewhere between the two extremes. In fact, several different points along the spectrum might be appropriate for use in different parts of a complex system. This is also the case in the HDRC3 architecture: Elementary action nodes in TSTs provide a set of high-level actions such as “takeoff” and “fly-to point A,” but one also makes use

Fig. 37.27 Task planning called by TST executors for goal nodes



of automated planning techniques to *compose* such actions into plans satisfying a set of declaratively specified goals. The goals themselves will vary depending on the nature of a mission. For example, they could involve having acquired images of certain buildings or having delivered crates of emergency supplies to certain locations after a natural disaster.

Support for planning is closely integrated in the HDRC3 architecture through the use of *goal nodes* in Task Specification Trees (Sect. 37.4.3). Any given TST can contain multiple goal nodes, each of which describes a simple or complex goal to be achieved at that particular point in the corresponding mission. When a goal node is assigned to a specific agent, the executor for that goal node can call DyKnow (Sect. 37.6) or use any other means at its disposal to acquire essential information about the current state of the world and the AV's own internal state (Fig. 37.27). The node is then *expanded* by that agent through a call to an onboard task planner, which in turn can call the motion planner and Motion Plan Info functionalities discussed earlier (Sect. 37.5.1). The resulting concurrent plan is converted to a new “plan subtree” attached under the goal node. Since TST structures are inherently distributable, the entire plan can then be distributed to the participating agents. This is the principal means of invoking a task planning functionality.

Note that the resulting TST directly represents a plan using sequence and concurrency nodes together with elementary action nodes corresponding to plan actions. Therefore, a plan can be executed at the appropriate time in the standard manner using TST executors. This obviates the need for a separate plan execution system and results in an integrated means of executing a mission, regardless of the techniques or combination of techniques that was used to generate it.

The planner that is used to expand a goal node can be changed freely, and in a heterogeneous system different platforms may use different planners. Currently the planners most commonly used in the HDRC3 architecture are TALplanner (Doherty and Kvarnström 2001; Kvarnström 2005) and TFPOP (Kvarnström 2011), two domain-independent concurrent temporal planners that were developed explicitly for use in this architecture. These planners are similar in certain respects: Both have a semantics based on Temporal Action Logic (Sect. 37.7.1), and both use *domain-specific control formulas* in this logic to guide the search for a solution, leading to very rapid plan generation. However, they use different search spaces and plan

structures, and they differ in the expressivity of the temporal constraints they permit. TALplanner will be described here, while the reader is referred to Kvarnström (2011) for further information about TFPOP.

TALplanner. TALplanner (Doherty and Kvarnström 1999; Kvarnström and Doherty 2000; Kvarnström 2005) is a forward-chaining temporal concurrent planner where planning domains and problem instances are specified as *goal narratives* in a version of TAL extended with new macros and statement types for plan operators, resource constraints, goal specifications, and other types of information specific to the task of plan generation. These macros serve to simplify planning domain specifications, but retain a basis in standard TAL through an extended translation function. For example, plan operators are converted into standard TAL action-type specifications, and actions in a plan are represented as standard timed action occurrences of the form $[t_1, t_2] A(\bar{v})$.

Control Formulas. In addition to providing a declarative first-order semantics for planning domains, TAL is also used to specify a set of domain-specific temporal *control formulas* acting as constraints on the set of valid solutions: A plan is a *solution* only if its final state satisfies the goal *and* all control formulas are satisfied in the complete state sequence that would result from executing the plan, which can be viewed as a logical model.

The use of control formulas serves two separate purposes. First, it allows the specification of complex temporally extended goals such as safety conditions that must be upheld throughout the (predicted) execution of a plan. Second, the additional constraints on the final solution often allow the planner to prune entire branches of the search tree – whenever it can be proven that every search node on the branch corresponds to a state sequence that violates at least one control rule. Formulas can then be written to guide the planner towards those parts of the search space that are more likely to contain plans of high quality, in terms of time usage or other quality measures.

As an example, consider three simple control rules that could be used in an airplane-based logistics domain. First, a package should only be loaded onto a plane if a plane is required to move it: if the goal requires it to be at a location in another city. Regardless of which operator is used to load a package, one can detect this through the fact that it is in a plane at time $t + 1$, but was *not* in the same plane at time t .

$$\begin{aligned} &\forall t, obj, plane, loc. \\ &[t] \neg in(obj, plane) \wedge at(obj, loc) \wedge [t+1] in(obj, plane) \rightarrow \\ &\exists loc' [goal(at(obj, loc')) \wedge [t] city_of(loc) \neq city_of(loc')] \end{aligned}$$

Second, if a package has been unloaded from a plane, there must be a valid reason for this: It must be the case that the package should be in the city where the plane has landed.

$$\begin{aligned}
& \forall t, obj, plane, loc. \\
& [t] \text{in}(obj, plane) \wedge \text{at}(plane, loc) \wedge [t+1] \neg \text{in}(obj, plane) \rightarrow \\
& \exists loc' [\text{goal}(\text{at}(obj, loc')) \wedge [t] \text{city_of}(loc) \doteq \text{city_of}(loc')]
\end{aligned}$$

Third, if a package is at its destination, it should not be moved.

$$\begin{aligned}
& \forall t, obj, loc. \\
& [t] \text{at}(obj, loc) \wedge \text{goal}(\text{at}(obj, loc)) \rightarrow [t+1] \text{at}(obj, loc)
\end{aligned}$$

Surprisingly, such simple hints to an automated planner can often improve planning performance by orders of magnitude given that the planner has the capability to make use of the hints.

Concurrent Plan Structure. Forward-chaining planning is most often used for sequential plans, where each new action is added immediately after the previous one in the plan. TALplanner uses a similar technique for concurrent plan generation, with a relaxed constraint on where a new action occurrence is placed.

Definition 37.1. (Concurrent Plan). A *concurrent plan* for a goal narrative \mathcal{G} is a tuple of ground fluent-free action occurrences with the following constraints. First, the empty tuple is a concurrent plan for \mathcal{G} . Second, given a concurrent plan $p = \langle [\tau_1, \tau'_1] o_1(\bar{c}_1), \dots, [\tau_n, \tau'_n] o_n(\bar{c}_n) \rangle$ for \mathcal{G} , its successors are the sequences that add one new action occurrence $[\tau_{n+1}, \tau'_{n+1}] o_{n+1}(\bar{c}_{n+1})$ and satisfy the following constraints:

1. Let $\mathcal{G}' = \mathcal{G} \cup \{[\tau_1, \tau'_1] o_1(\bar{c}_1), \dots, [\tau_n, \tau'_n] o_n(\bar{c}_n)\}$ be the original goal narrative \mathcal{G} combined with the existing plan. Then, the new action $o_{n+1}(\bar{c}_{n+1})$ must be applicable over the interval $[\tau_{n+1}, \tau'_{n+1}]$ in \mathcal{G}' . This implies that its preconditions are satisfied, that its effects are not internally inconsistent and do not contradict the effects of the operator instances already present in the sequence, and that the duration $\tau'_{n+1} - \tau_{n+1}$ is consistent with the duration given in the operator specification.
2. $\tau_1 = 0$: The first action starts at time 0.
3. $\tau_{n+1} \geq \tau_n$: The new action cannot be invoked before any of the actions already added to the plan.
4. $\tau_{n+1} \leq \max(\tau'_1, \dots, \tau'_n)$: There can be no gap between the time interval covered by the current plan and the start of the newly added action.

Generating Concurrent Plans. The concurrent TALplanner algorithm will now be briefly described (Algorithm 3). The reader is referred to Kvarnström (2005) for further details.

First, the planner conjoins all goal statements (line 2) and uses the control formulas in the goal narrative to generate a set of *pruning constraints*. These constraints allow control formulas to be verified efficiently when actions are incrementally added to a plan. Specifically, each control formula may result in an *initial constraint* to be tested initially, a set of operator-specific *incremental*

Algorithm 3 Concurrent TALplanner**Input:** A goal narrative \mathcal{G} .**Output:** A plan narrative entailing the goal $\mathcal{N}_{\text{goal}}$ and the control formulas $\mathcal{N}_{\text{control}}$.

```

1  procedure TALplanner-concurrent( $\mathcal{G}$ )
2   $\gamma \leftarrow \bigwedge \mathcal{G}_{\text{goal}}$  Conjunction of all goal statements
3   $\langle \text{init}, \text{incr}, \text{final} \rangle \leftarrow \text{generate-pruning-constraints}(\mathcal{G}_{\text{control}})$ 
4   $\text{node} \leftarrow \langle \text{init}, \emptyset, 0, 0, \{\} \rangle$  (cond. queue, visited states, latest invocation time,  $t_{\text{max}}$ , plan)
5   $\text{Open} \leftarrow \langle \text{node} \rangle$  Stack (depth first search)
6  while  $\text{Open} \neq \{\}$  do
7     $\langle C, S, \tau_0, \tau_{\text{max}}, p \rangle \leftarrow \text{pop}(\text{Open})$  Current plan candidate
8     $\mathcal{G}' \leftarrow \mathcal{G} \cup p \cup \text{occlude-all-after}(\mathcal{G}, \tau_0)$  No knowledge about future
9    for all constraints  $\alpha$  in  $C$  do Check queued constraints
10     if  $\text{Trans}^+(\mathcal{G}') \models \text{Trans}(\alpha)$  then  $C \leftarrow C \setminus \{\alpha\}$  Remove satisfied constraint
11     elseif  $\text{Trans}^+(\mathcal{G}') \models \text{Trans}(\neg\alpha)$  then backtrack Constraint violated
12      $\mathcal{G}'' \leftarrow \mathcal{G} \cup p \cup \{t_{\text{max}} = \tau_{\text{max}}\}$  Narrative with complete knowledge
13     if  $\text{Trans}^+(\mathcal{G}'') \models \text{false}$  then backtrack Consistency check
14     if  $\exists s \in S. \text{better-or-equal}(s, \text{final state of current plan})$  then backtrack
15     if  $\text{Trans}^+(\mathcal{G}'') \models \text{Trans}(\gamma \wedge C \wedge \text{final})$  then Goal + queued + final ctrl satisfied
16       return  $\mathcal{G}''$ 
17     else Not a solution, but check children
18        $S' \leftarrow S \cup \{\text{final state of current plan}\}$ 
19       for all successor actions  $\mathbf{A} = [\rho, \rho'] \circ^i(\bar{c})$  for  $p$  according to Def 37.1 do
20          $C' \leftarrow C \cup \text{incr}_i[\rho, \bar{c}]$  Old conditions + incr control
21          $C' \leftarrow C' \cup \{\text{prevail condition of } \mathbf{A}\}$  Add prevail condition
22         push  $\langle C', S', \rho, \max(\tau_{\text{max}}, \rho'), \langle p; \mathbf{A} \rangle \rangle$  onto  $\text{Open}$ 
23  fail

```

constraints to be tested whenever an instance of a particular operator is added to a plan, and a *final* constraint to be tested in the final solution.

The initial search node is created (line 4), as is a stack of open search nodes (for depth first search).

As long as the search space has not been exhausted (line 6), the planner retrieves the topmost node in the stack of open nodes. This node consists of a queue of constraints that remain to be evaluated (C), a set of visited states to be used in cycle checking (S), the latest invocation time of any action in the current plan (τ_0), the latest end time of any action in the current plan (t_{max}), and the current plan candidate (p).

Assuming a completely specified initial state and deterministic actions, the narrative $\mathcal{G} \cup p$ would now contain complete information about the development of the world that would result from executing exactly the plan p and nothing else. This then corresponds to a unique infinite state sequence specifying the values of all TAL fluents at all points in time. However, the current search node must only be pruned if *all possible extensions* of p violate a control formula. Given the constraints placed on successors in Definition 37.1, no action added to p can be invoked earlier than τ_0 , and therefore, all effects of actions added to p must take place at $\tau_0 + 1$ or later. Line 8 therefore generates the narrative $\mathcal{G} \cup p \cup \text{occlude-all-after}(\mathcal{G}, \tau_0)$ which has additional formulas disclaiming knowledge about fluents after time t_0 .

In lines 9–11, the planner determines whether a queued constraint is now definitely true (in which case it can be removed from the queue) or definitely false (in which case the planner has to backtrack).

Then, the planner must verify certain conditions under the assumption that no additional actions are added. A narrative \mathcal{G}'' assuming complete information is therefore generated in line 12. If this is inconsistent (which in practice can be tested efficiently given the expressivity of TALplanner operators), the planner must backtrack (line 13). If a state that is better or equal has already been visited, the planner should also backtrack (line 14). The *better-or-equal* relation can, for example, take into account resource availability, where a state that is equal in all respects except that it provides more of a particular resource can be considered strictly better.

In lines 15–16, TALplanner determines whether the current plan candidate satisfies the goal and all remaining pruning constraints. If this is the case, a solution is found and can be returned. If not, the node is expanded through the generation of a set of successors. Each successor may be given additional incremental pruning constraints, instantiated with the actual invocation timepoints and actual arguments of the corresponding action (line 20). Similarly, any *prevail conditions* (similar to preconditions but relating to the entire execution interval of an action) are added to the condition queue, to be tested when more information about future states is available.

37.7.3 Execution Monitoring

Regardless of the effort spent modeling all possible contingencies, actions and plans may fail. Robust performance in a noisy environment therefore requires some form of supervision, where the execution of a plan is constantly monitored in order to detect and recover from potential or actual failures. For example, since an AV might accidentally drop its cargo, it should monitor the condition that whenever it carries a crate, the crate remains until the AV reaches its intended destination. This is an example of a *safety constraint*, a condition that must be maintained during the execution of an action or across the execution of multiple actions. The carrier can also be too heavy, which means that one must be able to detect takeoff failures where the AV fails to gain sufficient altitude. This can be called a *progress constraint*: Instead of maintaining a condition, a condition must be achieved within a certain period of time. While some of these constraints would best be monitored at the control level, there are also many cases where monitoring and recovery should be lifted into a higher-level *execution monitor* (De Giacomo et al. 1998; Fernández and Simmons 1998; Fichtner et al. 2003; Gat et al. 1990; Ben Lamine and Kabanza 2002).

The execution monitoring system described here is based on an intuition similar to the one underlying the temporal control formulas used in TALplanner. As a plan is being executed, information about the surrounding environment is sampled at a given frequency by DyKnow (Sect. 37.6). Each new sampling point generates a

new state which provides information about all state variables used by the current monitor formulas, thereby providing information about the *actual* state of the world as opposed to what could be *predicted* from the domain specification. The resulting sequence of states corresponds to a partial logical interpretation, where “past” and “present” states are completely specified whereas “future” states are completely undefined.

Note that simply comparing the actual and predicted states and signaling a violation as soon as a discrepancy is found is not sufficient, because not all discrepancies are fatal – for example, if the altitude was predicted to be 5.0m and the current measurement turns out to be 4.984m. Also, some information might be expensive or difficult to sense, in which case the ground operator should be given more control over which information is actually gathered and used for monitoring. Sensing may even require special actions that interfere with normal mission operations. Finally, the richer the planner’s domain model is, the more it can predict about the development of the world. This should not necessarily lead to all those conditions being monitored, if they are not relevant to the correct execution of a plan. Therefore, most conditions to be monitored are *explicitly* specified, though many conditions *can* be automatically generated within the same framework if so desired.

Execution Monitor Formulas in TAL . Execution monitor formulas are expressed in a variation of TAL where the high-level language $\mathcal{L}(\text{ND})$ is augmented with a set of *tense operators* similar to those used in modal tense logics such as LTL (Emerson 1990) and MTL (Koymans 1990). Tense operators allow the expression of complex metric temporal conditions and are amenable to incremental evaluation as each new state is generated. This allows violations to be detected as early and as efficiently as possible using a *formula progression* algorithm, while the basis in TAL provides a common formal semantic ground for planning and monitoring.

Three tense operators have been introduced into $\mathcal{L}(\text{ND})$: $\text{U}(\text{until})$, $\Diamond(\text{eventually})$, and $\Box(\text{always})$. Like all expressions in $\mathcal{L}(\text{ND})$, these operators are macros on top of the first-order base language $\mathcal{L}(\text{FL})$.

Definition 37.2. (Monitor Formula). A *monitor formula* is one of the following:

- $\tau \leq \tau'$, $\tau < \tau'$, or $\tau = \tau'$, where τ and τ' are temporal terms.
- $\omega \leq \omega'$, $\omega < \omega'$, or $\omega = \omega'$, where ω and ω' are value terms.
- f , where f is a Boolean fluent term (state variable term).
- $f \hat{=} \omega$, where f is a fluent term and ω is a value term of the corresponding sort.
- $\phi \text{ U}_{[\tau, \tau']} \psi$, where ϕ and ψ are monitor formulas and τ and τ' are temporal terms.
- $\Diamond_{[\tau, \tau']} \phi$, where ϕ is a monitor formula and τ and τ' are temporal terms.
- $\Box_{[\tau, \tau']} \phi$, where ϕ is a monitor formula and τ and τ' are temporal terms.
- A combination of monitor formulas using standard logical connectives and quantification over values.

The shorthand notation $\phi \text{ U } \psi \equiv \phi \text{ U}_{[0, \infty)} \psi$, $\Diamond \phi \equiv \Diamond_{[0, \infty)} \phi$, and $\Box \phi \equiv \Box_{[0, \infty)} \phi$ is also permitted.

Tense operators use relative time, where each formula is evaluated relative to a “current” timepoint. The semantics of these formulas satisfies the following conditions (see Doherty et al. (2009) for details):

- The formula $\phi \mathbf{U}_{[\tau, \tau']} \psi$ (“until”) holds at time t iff ψ holds at some state with time $t' \in [t + \tau, t + \tau']$ and ϕ holds until then (at all states in $[t, t']$, which may be an empty interval).
- The formula $\Diamond_{[\tau, \tau']} \phi$ (“eventually”) is equivalent to $\text{true} \mathbf{U}_{[\tau, \tau']} \phi$ and holds at t iff ϕ holds in some state with time $t' \in [t + \tau, t + \tau']$.
- The formula $\Box_{[\tau, \tau']} \phi$ is equivalent to $\neg \Diamond_{[\tau, \tau']} \neg \phi$ and holds at t iff ϕ holds in all states with time $t' \in [t + \tau, t + \tau']$.

Example 37.3. Suppose that an AV supports a maximum continuous power usage of M , but can exceed this by a factor of f for up to τ units of time, if this is followed by normal power usage for a period of length at least τ' . The following formula can be used to detect violations of this specification:

$$\Box \forall av. (\text{power}(av) > M \rightarrow \text{power}(av) < f \cdot M \mathbf{U}_{[0, \tau]} \Box_{[0, \tau']} \text{power}(av) \leq M) \quad \Box$$

Note that this does not in itself *cause* the AV to behave in the desired manner. That has to be achieved in the lower-level implementations of the helicopter control software. The monitor formula instead serves as a method for detecting the failure of the helicopter control software to function according to specifications.

Monitors and Actions. In many cases, conditions that should be monitored are closely related to the actions currently being executed. Two extensions are made to facilitate the specification of such formulas.

First, monitor formulas can be explicitly associated with specific operator types. Unlike global monitor formulas, such formulas are not activated before plan execution but before the execution of a particular *step* in the plan, which provides the ability to contextualize a monitor condition relative to a particular action. An operator-specific monitor formula can also directly refer to the arguments of the associated operator.

Example 37.4. Execution should be monitored when an AV attempts to pick up a box. Since the arguments of *pickup-box* include the *av* and the *box*, the following operator-specific monitor formula can be used:

$$\Diamond_{[0, 5000]} \Box_{[0, 1000]} \text{carrying}(av, box)$$

Within 5,000 ms, the AV should detect that it is carrying the box, and it should detect this for at least 1,000 ms. The latter condition protects against problems during the pickup phase, where the box may be detected during a very short period of time even though the ultimate result is failure. \Box

Second, a set of *execution flags* allow monitor formulas to query the internal execution state of an agent. This is useful when one wants to state that a certain fact

should *hold during* the execution of an action or that an effect should be *achieved during* the execution of an action or *before* the execution of another action. For example, if an AV *picks up a crate*, it should sense the weight of the crate until it deliberately *puts down the crate*.

An execution flag is an ordinary Boolean state variable which holds exactly when the corresponding action is being executed. By convention, this state variable will generally be named by prepending “executing-” to the name of the corresponding operator: The pickup-box operator is associated with the *executing-pickup-box* execution flag, which takes a subset of the operator’s parameters. TST executors for elementary action nodes can set this flag when execution starts and clear it when execution ends.

Example 37.5. Consider the *climb(av)* operator, which should cause the AV to ascend to its designated flight altitude. Here, one may wish to monitor the fact that the AV truly ends up at its flight altitude. This can be achieved using the formula $\text{executing-climb}(av) \cup \text{altitude}(av) \geq 7.0$. \square

When it is clear from context which operator is intended, the shorthand notation EXEC can be used to refer to its associated execution flag with default parameters: $\text{EXEC} \cup \text{altitude}(av) \geq 7.0$.

Example 37.6. Whenever an AV picks up a box, it should detect the box within 5,000ms and keep detecting it until it is explicitly put down. Using an operator-specific monitor formula for pickup-box:

$\text{EXEC} \cup_{[0,5000]} (\text{carrying}(av, box) \cup \text{executing-putdown}(av, box))$

Automatic Generation of Monitor Formulas. The use of a single logical formalism for modeling both planning and execution monitoring provides ample opportunities for the automatic generation of conditions to be monitored. For example, one can automatically generate formulas verifying that preconditions and prevail conditions (which must hold throughout the execution of an action) are satisfied, that expected effects take place, and that bounds on action durations are not violated. Similarly one can automatically extract *causal links*, where one action achieves one condition that is later needed by another, and generate a formula verifying that this condition is never violated in the interval between these actions. Since there are pragmatic reasons for *not* generating all possible monitor formulas, automatic generation only takes place for those conditions that are flagged for monitoring. This provides the benefits of automatic formula generation while keeping the control in the hands of the domain designer. See Doherty et al. (2009) for details.

37.7.3.1 Recovery from Failures

Any monitor formula violation signals a potential or actual failure from which the system must attempt to recover in order to achieve its designated goals.

Recovery is a complex topic, especially when combined with the stringent safety regulations associated with autonomous flight. Goals and constraints may be time dependent, making local repairs difficult: An AV might only be allowed to fly in certain areas at certain times. Also, recovering from a failure for one AV may require changing the plans of another AV: If **heli1** fails to deliver a box of medicine on time, **heli2** might have to be rerouted. Therefore, the main focus has been on recovery through replanning. Given that the planner is sufficiently fast when generating new plans, this does not adversely affect the execution of a fully autonomous mission.

Having detected a failure, the first action of an AV is to suspend the execution of the current plan, execute an emergency break if required, and then go into autonomous hover mode. Currently, one takes advantage of the fact that the UAS Tech Lab RMAX and LinkQuads are rotor based and can hover. For fixed-wing platforms, this is not an option, and one would have to go into a loiter mode if the recovery involves time-consuming computation.

This is followed by the execution of a *recovery operator*, if one is associated with the violated monitor formula. The recovery operator can serve two purposes: It can specify emergency recovery procedures that must be initiated immediately without waiting for replanning, and it can permit the execution system to adjust its assumptions about what can and cannot be done. For example, if an AV fails to take off with a certain carrier, the associated recovery operator can adjust the AV's assumptions about how many boxes it is able to lift. This feeds back information from the failure into the information given to the planner for replanning. The implementation of a recovery operator can also detect the fact that the AV has attempted and failed to recover from the same fault too many times and choose whether to give up, try another method, remove some goals in order to succeed with the remaining goals, or contact a human for further guidance.

37.7.3.2 Execution Monitoring with Inaccurate Sensors

Monitoring should not only maximize the probability that a failure is detected but also minimize the probability of false positives, where a failure is signaled but none has occurred. Some problems, such as those caused by dropouts and communication delays, can be ameliorated by extrapolating historical values and by delaying state generation slightly to allow values to propagate through the distributed system. Noise could be minimized through sensor value smoothing techniques and sensor fusion techniques. However, inaccuracies in the detected state sequence can never be completely eliminated. Therefore, the fact that state values may be inaccurate should be taken into consideration when writing monitor formulas.

For example, the meaning of the condition $\Box \forall av. \text{speed}(av) \leq T$ is that the *sensed and approximated* speed of an AV must never exceed the threshold T . Since a single observation of $\text{speed}(av)$ above the threshold might be an error or a temporary artifact, a more robust solution would be to signal a failure if the sensed speed has been above the threshold during an interval $[0, \tau]$ instead of at a single timepoint. This can be expressed as $\Box \Diamond_{[0, \tau]} \text{speed}(av) \leq T$: It should always be the case that within the interval $[0, \tau]$ from now, the sensed speed returns to being below the threshold.

This formula is somewhat weak: It only requires that a single measurement in every interval of length τ is below the threshold. An alternative would be to require that within τ time units, there will be an *interval* of length τ' during which the AV stays within the limits: $\Box(\text{speed}(av) > T \rightarrow \Diamond_{[0,\tau]} \Box_{[0,\tau']} \text{speed}(av) \leq T)$.

37.7.3.3 Formula Progression

To promptly detect violations of monitor conditions during execution, a *formula progression* algorithm is used (Bacchus and Kabanza 1998). By definition, a formula ϕ holds in the state sequence $[s_0, s_1, \dots, s_n]$ iff $\text{Progress}(\phi, s_0)$ holds in $[s_1, \dots, s_n]$. In essence, this evaluates those parts of the monitor formula that refer to s_0 , returning a new formula to be progressed in the same manner once s_1 arrives.

If the formula \perp (false) is returned, then sufficient information has been received to determine that the monitor formula must be violated regardless of the future development of the world. For example, this will happen as soon as the formula $\Box \text{speed} < 50$ is progressed through a state where $\text{speed} \geq 50$. Similarly, if \top (true) is returned, the formula must hold regardless of what happens “in the future.” This will occur if the formula is of the form $\Diamond \phi$ (eventually, ϕ will hold), and one has reached a state where ϕ indeed does hold. In other cases, the state sequence complies with the constraint “so far,” and progression will return a new and potentially modified formula that should be progressed again as soon as another state is available.

Definition 37.3. (Progression of Monitor Formulas). The following algorithm is used for progression of monitor formulas. Note that states are not first-class objects in TAL and are therefore identified by a timepoint τ and an interpretation \mathcal{I} . Special cases for \Box and \Diamond can be introduced for performance.

```

1  procedure Progress( $\phi, \tau, \mathcal{I}$ )
2  if  $\phi = f(\bar{x}) \hat{=} v$ 
3    if  $\mathcal{I} \models \text{Trans}([\tau] \phi)$  return  $\top$  else return  $\perp$ 
4  if  $\phi = \neg \phi_1$  return  $\neg \text{Progress}(\phi_1, \tau, \mathcal{I})$ 
5  if  $\phi = \phi_1 \otimes \phi_2$  return  $\text{Progress}(\phi_1, \tau, \mathcal{I}) \otimes \text{Progress}(\phi_2, \tau, \mathcal{I})$ 
6  if  $\phi = \forall x. \phi$  // where  $x$  belongs to the finite domain  $X$ 
7    return  $\bigwedge_{c \in X} \text{Progress}(\phi[x \mapsto c], \tau, \mathcal{I})$ 
8  if  $\phi = \exists x. \phi$  // where  $x$  belongs to the finite domain  $X$ 
9    return  $\bigvee_{c \in X} \text{Progress}(\phi[x \mapsto c], \tau, \mathcal{I})$ 
10 if  $\phi$  contains no tense operator
11   if  $\mathcal{I} \models \text{Trans}(\phi)$  return  $\top$  else return  $\perp$ 
12 if  $\phi = \phi_1 \cup_{[\tau_1, \tau_2]} \phi_2$ 
13   if  $\tau_2 < 0$  return  $\perp$ 
14   elseif  $0 \in [\tau_1, \tau_2]$  return  $\text{Progress}(\phi_2, \tau, \mathcal{I}) \vee (\text{Progress}(\phi_1, \tau, \mathcal{I}) \wedge$ 
       $(\phi_1 \cup_{[\tau_1-1, \tau_2-1]} \phi_2))$ 
15   else return  $\text{Progress}(\phi_1, \tau, \mathcal{I}) \wedge (\phi_1 \cup_{[\tau_1-1, \tau_2-1]} \phi_2)$ 

```

The result of Progress is simplified using the rules $\neg \perp = \top$, $(\perp \wedge \alpha) = (\alpha \wedge \perp) = \perp$, $(\perp \vee \alpha) = (\alpha \vee \perp) = \alpha$, $\neg \top = \perp$, $(\top \wedge \alpha) = (\alpha \wedge \top) = \alpha$,

and $(\top \vee \alpha) = (\alpha \vee \top) = \top$. Further simplification is possible using identities such as $\Diamond_{[0,\tau]} \phi \wedge \Diamond_{[0,\tau']} \phi \equiv \Diamond_{[0,\min(\tau,\tau'')]} \phi$.

For this approach to be useful, it must be possible to progress typical monitor formulas through the state sequences generated during a mission using the often limited computational power available in an autonomous robotic system. The following evaluation uses one synthetic test where one can study complex combinations of time and modality. An earlier version of the actual DRC computer onboard a UASTL RMAX was used to run progression tests for formulas having a form that is typical for monitor formulas in many applications. State sequences are constructed to exercise both the best and the worst cases for these formulas.

The evaluation shows that even with the worst possible inputs, complex formulas can be evaluated in less than 1 ms per state and formula on this CPU (1.4 GHz Pentium M). One example formula is $\Box \neg p \rightarrow \Diamond_{[0,1000]} \Box_{[0,999]} p$, corresponding to the fact that if the property p is false, then within 1,000 ms, there must begin a period lasting at least 1,000 ms where p is true. For example, the previously discussed formula $\Box(\text{speed}(av) > T \rightarrow \Diamond_{[0,\tau]} \Box_{[0,\tau']} \text{speed}(av) \leq T)$ has this general form. To estimate the cost of evaluating this formula, it was progressed through several different state streams corresponding to the best case, the worst case, and two intermediate cases. A new state in the stream was generated every 100 ms, which means that all formulas must be progressed within this time limit or monitoring will fall behind. Figure 37.28 shows the average time required to progress a certain number of formulas through each state in a sequence. This indicates that 100 ms is sufficient for progressing between 1,500 and 3,000 formulas of this form on the computer onboard the UASTL RMAX, depending on the state stream. Similar results have been shown for formulas of different forms, such as $\Box \neg p \rightarrow \Diamond_{[0,1000]} \Box_{[0,999]} p$ and $\Box \neg p \rightarrow \Diamond_{[0,1000]} \Box_{[0,999]} p$ (Doherty et al. 2009).

37.7.4 High-Level Mission Specification

The ability to clearly and concisely specify missions is fundamentally important for an unmanned system to be practically useful and requires a suitable *mission specification language* that should satisfy a variety of requirements and desires. The language should be comprehensible to humans and not only useful as an intermediate representation both generated and received by software. At the same time intuitions are not sufficient: A strict formal semantics must be available. There should also be a close connection to mission execution, allowing the actual semantics of the language to be used to specify the correct operation of the system and thereby facilitating the validation of the system as a whole. A principled foundation where these issues are considered, for single platforms (vehicles) as well as for fleets of homogeneous or heterogeneous platforms, is essential for these types of systems to be accepted by aviation authorities and in society.

The mission specification language used in the HDRC3 architecture is designed to allow partial mission specifications with constraints, including resource

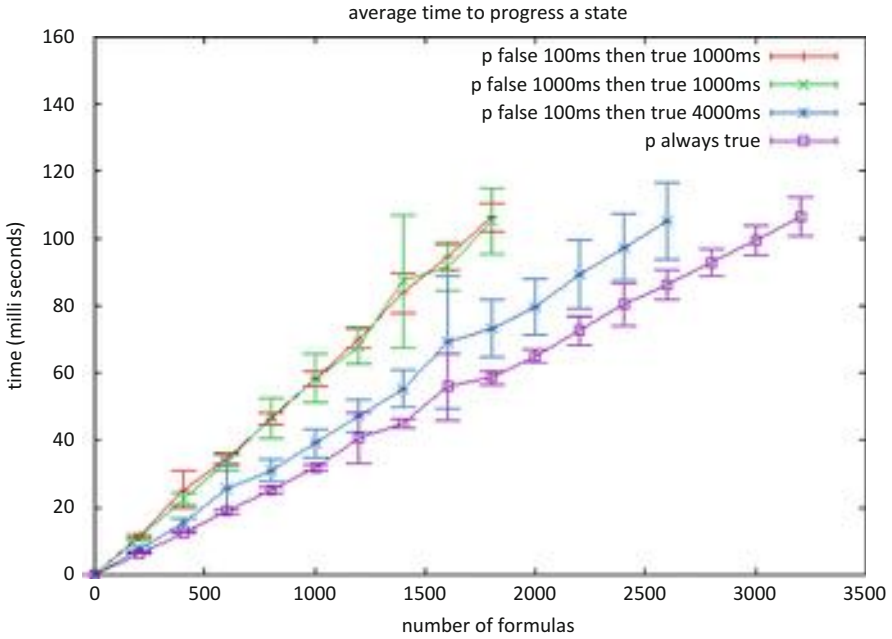


Fig. 37.28 Average progression time: always $\neg p \rightarrow$ eventually always p

requirements and temporal deadlines. It extends the support for highly expressive *elementary actions* in Temporal Action Logic (TAL, Sect. 37.7.1) with *temporal composite actions* that also provide a formal semantics for the high-level structure of a mission (Doherty et al. 2012). The composite action constructs in the extended logic TALF correspond closely to the control structures supported by Task Specification Trees (Sect. 37.4.3). As shown in Fig. 37.29, missions can therefore be defined either in TALF or as TSTs and can be translated in either direction. The *delegation* functionality discussed in Sect. 37.8 can then be used to delegate a mission in the shape of a TST to one or more platforms for execution, resulting in a distributed task structure.

Composite actions in TALF are characterized recursively through the general construct “with VARS do TASK where CONS”: Any composite action consists of a task TASK that should be executed in a context characterized by a set of variables VARS constrained by a set of constraints CONS. The TASK, in turn, can be an elementary TAL action or consist of a combination of composite actions using constructs such as sequential composition, parallel (concurrent) composition, conditional, (unbounded) loops, while-do, and a concurrent for-each operator allowing a variable number of actions to be executed concurrently. At the mission specification level considered here, each constraint definition can be as general as a logical formula in TAL, giving it a formal semantics. For pragmatic use in a robotic architecture, a wide class of formulas can be automatically transformed into constraints processed

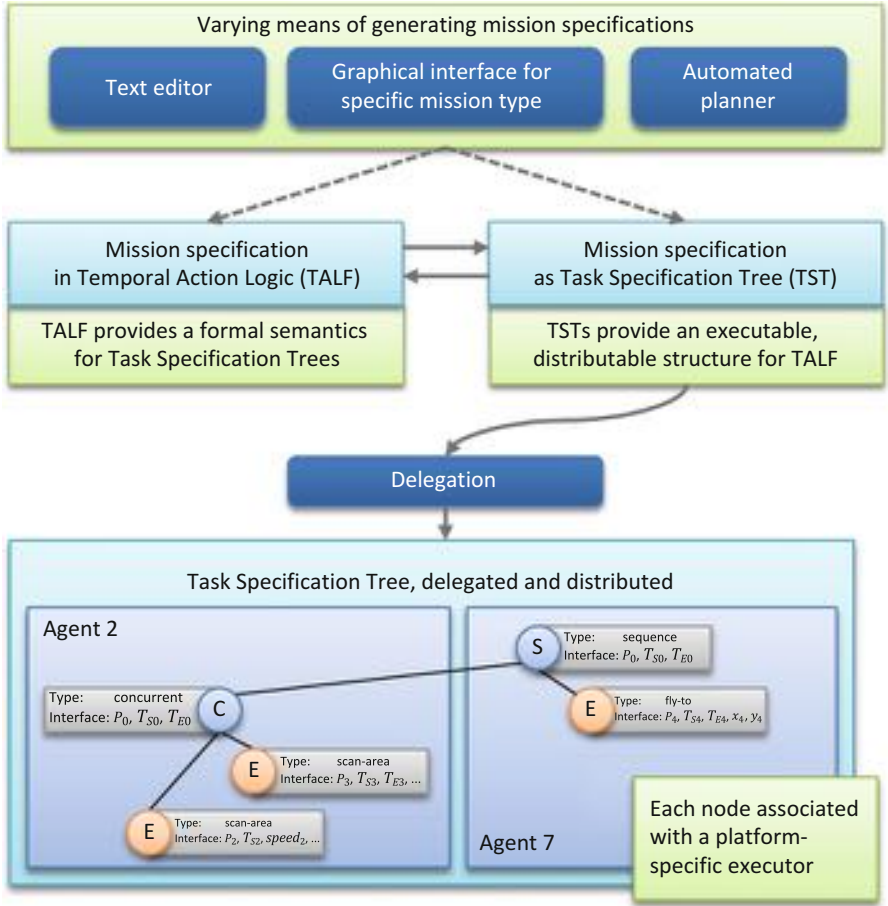


Fig. 37.29 Mission specification, translation, and delegation

by a constraint satisfaction solver, allowing a robotic system to formally verify the consistency of a (distributed) task through the use of (distributed) constraint satisfaction techniques.

A *composite action-type specification* declares a named composite action. This is useful in order to define a library of meaningful composite actions to be used in mission specifications. Each specification is of the form $[t, t'] \text{comp}(\bar{v}) \rightsquigarrow A(t, t', \bar{v})$ where $\text{comp}(\bar{v})$ is a *composite action term* such as $\text{monitor-pattern}(x, y, \text{dist})$, consisting of an action name and a list of parameters, and $A(t, t', \bar{v})$ is a *composite action expression* where only variables in $\{t, t'\} \cup \bar{v}$ may occur free. A composite action expression (C-ACT), in turn, allows actions to be composed at a high level of abstraction using familiar programming language constructs such as sequences ($A; B$), concurrency ($A \parallel B$), conditions, and loops. The associated syntax is defined as follows:

$$\begin{aligned}
\text{C-ACT} &::= [\tau, \tau'] \text{ with } \bar{x} \text{ do TASK where } \phi \\
\text{TASK} &::= [\tau, \tau'] \text{ELEM-ACTION-TERM} \mid \\
&\quad [\tau, \tau'] \text{COMP-ACTION-TERM} \mid \\
&\quad (\text{C-ACT}; \text{C-ACT}) \mid \\
&\quad (\text{C-ACT} \parallel \text{C-ACT}) \mid \\
&\quad \text{if } [\tau] \psi \text{ then C-ACT else C-ACT} \mid \\
&\quad \text{while } [\tau] \psi \text{ do C-ACT} \mid \\
&\quad \text{foreach } \bar{x} \text{ where } [\tau] \psi \text{ do conc C-ACT}
\end{aligned}$$

where \bar{x} is a potentially empty sequence of variables (where the empty sequence can be written as ϵ), ϕ is a TAL formula representing a set of constraints, ELEM-ACTION-TERM is an elementary action term such as **fly-to**(av, x, y), COMP-ACTION-TERM is a composite action term, and $[\tau] \psi$ is a TAL formula referring to facts at a single timepoint τ . For brevity, omitting “with \bar{x} do” is considered equivalent to specifying the empty sequence of variables, and omitting “where ϕ ” is equivalent to specifying “where TRUE.” Note that the $;$ and \parallel constructs are easily extended to allow an arbitrary number of actions, as in $(A; B; C; D)$.

Like elementary actions, every composite action C-ACT is annotated with a temporal execution interval. This also applies to each “composite sub-action.” For example,

$$\begin{aligned}
&[t_1, t_2] \text{ with } av, t_3, t_4, t_5, t_6 \text{ do} \\
&\quad ([t_3, t_4] \text{fly-to}(av, x, y); [t_5, t_6] \text{collect-video}(av, x, y)) \\
&\quad \text{where } [t_1] \text{has-camera}(av)
\end{aligned}$$

denotes a composite action where one elementary action takes place within the interval $[t_3, t_4]$, the other one within the interval $[t_5, t_6]$, and the entire sequence within $[t_1, t_2]$.

The with-do-where construct provides a flexible means of constraining variables as desired for the task at hand. In essence, “ $[t_1, t_2]$ with \bar{x} do TASK where ϕ ” states that there exists an instantiation of the variables in \bar{x} such that the specified TASK (which may make use of \bar{x} as illustrated above) is executed within the interval $[t_1, t_2]$ in a manner satisfying ϕ . The constraint ϕ may be a combination of temporal, spatial, and other types of constraints. Above, this constraint is used to ensure the use of an av that has a camera rather than an arbitrary av .

As the aim is to maximize temporal flexibility, the sequence operator ($;$) does not implicitly constrain the two actions **fly-to** and **collect-video** to cover the entire temporal interval $[t_1, t_2]$. Instead, the actions it sequentializes are only constrained to occur *somewhere within* the execution interval of the composite action, and gaps are permitted between the actions – but all actions in a sequence must occur in the specified order without overlapping in time. Should stronger temporal constraints be required, they can be introduced in a where clause. For example, $t_1 = t_3 \wedge t_4 = t_5 \wedge t_6 = t_2$ would disallow gaps in the sequence above. Also, variations such as gapless sequences can easily be added as first-class language constructs if desired.

Formal Semantics. To define a formal semantics for TALF, the base logic $\mathcal{L}(\text{FL})$ is first extended with support for *fixpoint formulas*. This is required for unbounded loops and unbounded recursion, which cannot be characterized in the first-order logic $\mathcal{L}(\text{FL})$. A translation is then defined from TALF into the extended base logic $\mathcal{L}(\text{FL}_{\text{FP}})$. As a result of this translation, a composite action is a theory in $\mathcal{L}(\text{FL}_{\text{FP}})$. Questions about missions thereby become queries relative to an inference mechanism, allowing operators to analyze mission properties both during pre- and post-mission phases. This also provides a formal semantics for Task Specification Trees, which can be translated into TALF, and thereby for the execution system used in the HDRC3 architecture. Details regarding translations and the resulting formal semantics are specified in Doherty et al. (2012).

Examples. In November 2011, a powerful earthquake off the coast of Japan triggered a tsunami with devastating effects, including thousands of dead and injured as well as extensive damage to cities and villages. Another effect, which became increasingly apparent over time, was the extensive damage to the Fukushima Daiichi nuclear plant which later resulted in a complete meltdown in three reactors. The exact level of damage was initially difficult to assess due to the danger in sending human personnel into such conditions. Here unmanned aircraft could immediately have assisted in monitoring radiation levels and transmitting video feeds from a closer range. Several composite actions that can be useful for the problem of information gathering in this situation will now be considered. The focus is on demonstrating the $\mathcal{L}(\text{ND})$ composite action constructs, and some aspects of the actions below are simplified for expository reasons.

Assume the existence of a set of elementary actions whose meaning will be apparent from their names and from the explanations below: **hover-at**, **fly-to**, **monitor-radiation**, **collect-video**, and **scan-cell**. Each elementary action is assumed to be defined in standard TAL and to provide suitable preconditions, effects, resource requirements, and (completely or incompletely specified) durations. For example, only an AV with suitable sensors can execute **monitor-radiation**.

In the following composite action, an AV hovers at a location $(x_{\text{av}}, y_{\text{av}})$ while using its onboard sensors to monitor radiation and collect video at $(x_{\text{targ}}, y_{\text{targ}})$:

$$\begin{aligned}
 & [t, t'] \text{monitor-single}(av, x_{\text{av}}, y_{\text{av}}, x_{\text{targ}}, y_{\text{targ}}) \rightsquigarrow \\
 & [t, t'] \text{with } t_1, t_2, t_3, t_4, t_5, t_6 \text{ do (} \\
 & \quad [t_1, t_2] \text{hover-at}(av, x_{\text{av}}, y_{\text{av}}) \parallel \\
 & \quad [t_3, t_4] \text{monitor-radiation}(av, x_{\text{targ}}, y_{\text{targ}}) \parallel \\
 & \quad [t_5, t_6] \text{collect-video}(av, x_{\text{targ}}, y_{\text{targ}}) \\
 & \quad \left. \right) \text{ where } [t] \text{surveil-equipped}(av) \wedge \text{radiation-hardened}(av) \wedge t_1 = t_3 = t_5 \\
 & \quad = t \wedge t_2 = t_4 = t_6 = t'
 \end{aligned}$$

The first part of the constraint specified in the *where* clause ensures that an AV involved in a monitoring action is equipped for surveillance and is radiation-hardened (in addition to the conditions placed on **monitor-radiation**, which include

the existence of radiation sensors). The temporal constraints model a requirement for these particular actions to be synchronized in time and for the AV to hover in a stable location throughout the execution of monitor-single. These constraints could easily be relaxed, for example, by stating that hovering occurs throughout the action but monitoring occurs in a subinterval.

The following action places four AVs in a diamond pattern to monitor a given location such as a nuclear reactor at a given distance, counted in grid cells. The AVs involved are not specified as parameters to the monitoring action but are chosen freely among available AVs, subject to the constraints modeled by sub-actions such as monitor-single:

```
[t, t'] monitor-pattern(x, y, dist)  $\rightsquigarrow$ 
[t, t'] with  $s_1, \dots, w_4, av_1, av_2, av_3, av_4$  do (
  ( $[s_1, s_2]$  fly-to( $av_1, x + dist, y$ );  $[s_3, s_4]$  monitor-single( $av_1, x + dist, y, x, y$ )) ||
  ( $[u_1, u_2]$  fly-to( $av_2, x - dist, y$ );  $[u_3, u_4]$  monitor-single( $av_2, x - dist, y, x, y$ )) ||
  ( $[v_1, v_2]$  fly-to( $av_3, x, y + dist$ );  $[v_3, v_4]$  monitor-single( $av_3, x, y + dist, x, y$ )) ||
  ( $[w_1, w_2]$  fly-to( $av_4, x, y - dist$ );  $[w_3, w_4]$  monitor-single( $av_4, x, y - dist, x, y$ )))
where
 $s_3 = u_3 = v_3 = w_3 \wedge s_4 = u_4 = v_4 = w_4 \wedge s_4 - s_3 \geq \text{minduration}$ 
```

Four sequences are executed in parallel. Within each sequence, a specific AV flies to a suitable location and then monitors the target. The target must be monitored simultaneously by all four AVs ($s_3 = u_3 = v_3 = w_3$ and $s_4 = u_4 = v_4 = w_4$), while $s_4 - s_3 \geq \text{minduration}$ ensures this is done for at least the specified duration. As flying does not need to be synchronized, the intervals for the **fly-to** actions are only constrained *implicitly* through the definition of a sequence. For example, the translation ensures that $t \leq s_1 \leq s_2 \leq s_3 \leq s_4 \leq t'$, so that each **fly-to** must end before the corresponding monitor-single.

All grid cells must also be scanned for injured people. The following generic action uses all available AVs with the proper capabilities, under the assumption that each such AV has been assigned a set of grid cells to scan. An assignment could be generated by another action or provided as part of the narrative specification. For clarity, this includes several clauses (with ϵ do, where TRUE) that could easily be omitted:

```
[t, t'] scan-with-all-uavs()  $\rightsquigarrow$ 
[t, t'] with  $\epsilon$  do
  foreach  $av$  where  $[t]$  can-scan( $av$ ) do conc
     $[t, t']$  with  $u, u'$  do  $[u, u']$  scan-for-people( $av$ ) where TRUE
  where TRUE
```

As shown below, each AV involved in this task iterates while there remains at least one cell (x, y) that it has been assigned (“owns”) and that is not yet scanned. In each iteration the variables (x', y') declared in the nested with clause range over

arbitrary coordinates, but the associated where clause ensures that only coordinates that belong to the given AV and that have not already been scanned can be selected. Also in each iteration, t_c is bound to the time at which the constraint condition is tested and u, u' are bound to the timepoints at which the inner composite action is performed. The repeated use of u, u' is intentional. The **scan-cell** action will occur over exactly the same interval as the enclosing composite action construct:

$$\begin{aligned}
 [t, t'] \text{ scan-for-people}(av) \rightsquigarrow & \\
 [t, t'] \text{ with } \epsilon \text{ do} & \\
 \quad \text{while } [t_c] \exists x, y [\text{owns}(av, x, y) \wedge \neg \text{scanned}(x, y)] \text{ do} & \\
 \quad \quad [u, u'] \text{ with } x', y' \text{ do } [u, u'] \text{scan-cell}(av, x', y') \text{ where } [t_c] \text{owns} & \\
 \quad \quad (av, x', y') \wedge \neg \text{scanned}(x', y') & \\
 \quad \text{where TRUE} &
 \end{aligned}$$

It is now possible to define a small mission to occur within the interval $[0, 1000]$, where scanning may use the entire interval while the grid cell $(20, 25)$ is monitored at a distance of 3 cells and must terminate before time 300:

$$[0, 1000]([0, 1000] \text{ scan-with-all-uavs() } || [0, 300] \text{ monitor-pattern}(20, 25, 3))$$

It should be emphasized that in the expected case, the task of generating specifications of this kind would be aided by libraries of predefined domain-related actions as well as by user interfaces adapted to the task at hand. The structure and high-level nature of the language remains important when ensuring that these tools and their output are both correct and comprehensible to a human operator inspecting a mission definition.

37.8 Collaborative Systems

Though the main focus of this chapter has been on the operation of a single unmanned aircraft, issues related to cooperation and collaboration are also essential for the successful use of such systems.

At the cooperative level, the combination of an aircraft or other robotic *platform* and its associated software is viewed as an *agent*. Humans interacting with platforms through, for example, ground control stations and other interfaces are also considered to be agents. Taken together, these agents form a collaborative system where all participants can cooperate to perform missions. One aspect of a collaborative system is that all agents are conceptually equal and independent in the sense that there is no predefined control structure, hierarchical or otherwise. A consequence is that the control structure can be determined on a mission-to-mission basis and dynamically changed during a mission.

When an unmanned system is viewed as an agent acting on behalf of humans, it is also natural to view the assignment of a complex mission to that system as *delegation* – by definition, the act of assigning authority and responsibility to another person, or in a wider sense an agent, in order to carry out specific activities. Delegation is therefore part of the foundation for collaboration in this architecture: A *delegator* can ask a *contractor* to take the responsibility for a specific *task* to be performed under a set of *constraints*. Informally, an agent receiving a delegation request must verify that to the best of its knowledge, it will be able to perform the associated task under the given constraints, which may for example, concern resource usage or temporal and spatial aspects of the mission. To ensure that a complex task is carried out in its entirety, an agent may have to enlist the aid of others, which can then be delegated particular parts of the task. This results in a network of responsibilities between the agents involved and can continue down to the delegation of elementary, indivisible actions. If such a network can be generated in a way that satisfies the associated constraints, the contractor can accept the delegation and is then committed to doing everything in its power to ensure the task is carried out. If not, it must refuse. See Doherty et al. (2013) for a formal characterization of delegation in terms of speech acts as well as a practical delegation protocol that also determines how to allocate tasks to specific agents.

Delegation requires a flexible task structure with a clear formal semantics. This role is played by Task Specification Trees (Sect. 37.4.3), whose hierarchical nature also leads to a natural recursive decomposition of tasks where the children of a node are the subtasks that can be re-delegated to other agents. For example, an automated multi-agent planner (Sect. 37.7.2) can generate a TST where elementary action nodes are not assigned to specific agents. Then, the delegation process and its embedded *task allocation* functionality (Doherty et al. 2013) can recursively determine how the plan can be executed in a way that satisfies associated constraints. The formal semantics of the task being delegated is specified through a close connection to TALF (Sect. 37.7.4). See Doherty et al. (2011, 2013) and Doherty and Meyer (2012) for further details about delegation, its close relation to *adjustable autonomy* and *mixed-initiative interaction*, and its integration with automated planning.

Legacy Systems. When an agent-based architecture is used together with an existing platform such as an unmanned aircraft, there may already be an existing legacy system providing a variety of lower-level functionalities such as platform-specific realizations of elementary tasks and resources. Existing interfaces to such functionalities can vary widely. The current instantiation of the architecture (Fig. 37.30a) directly supports the use of such legacy functionalities through the use of an agent layer and a gateway. The *agent layer* (Fig. 37.30b) encapsulates higher-level deliberative functionalities and provides a common interface for multi-agent collaboration in complex missions, including support for mission specification languages, delegation, and planning. The *gateway* must have a platform-specific implementation, but provides a common platform-independent external interface to

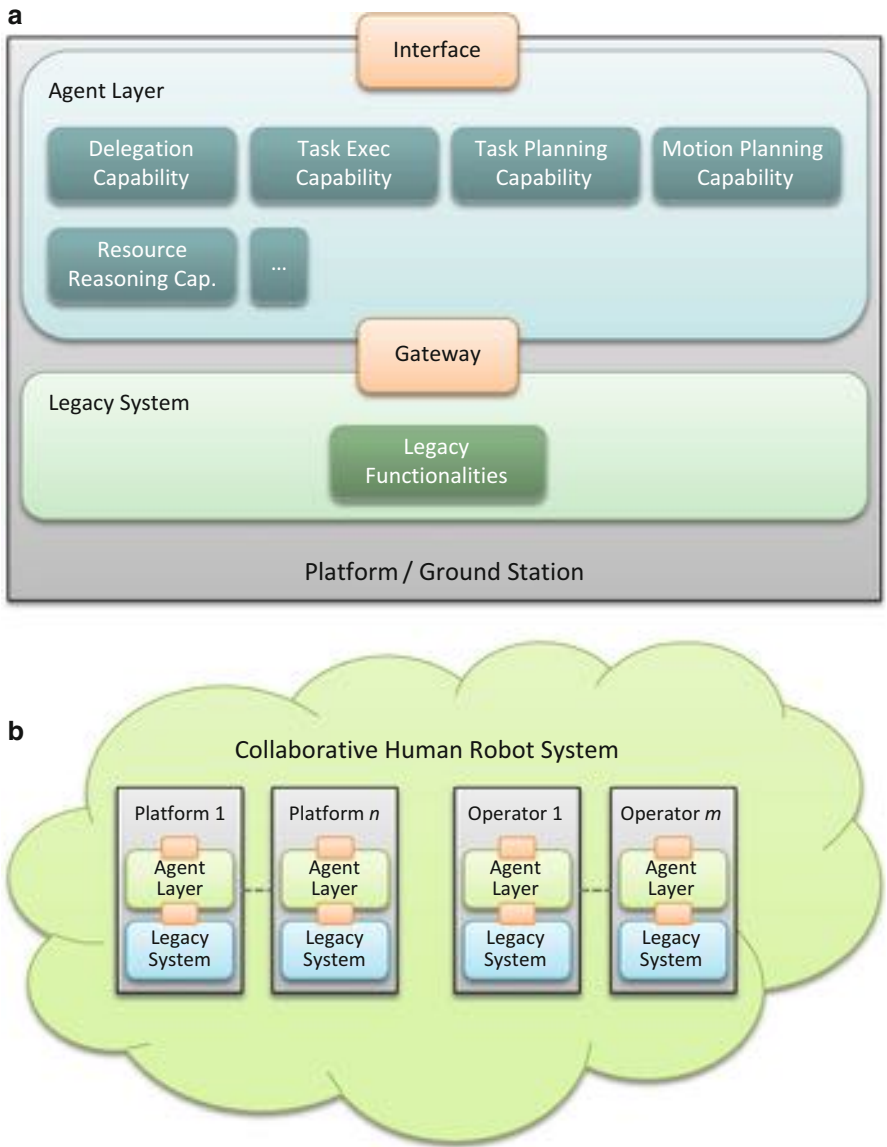


Fig. 37.30 Agents in a collaborative system. (a) Agentified platform or ground control station. (b) Overview of the collaborative human/robot system

the available legacy functionalities. In essence, this allows newly developed higher-level functionalities to be seamlessly integrated with existing systems, without the need to modify either the agent layer or the existing system. The agent layer can then be developed independently of the platforms being used.

Legacy control stations and user interfaces that human operators use to interact with robotic systems are treated similarly, through the addition of an agent layer. The result is a collaborative human/robot system consisting of a number of human operators and robotic platforms each having an agent layer and possibly a legacy system, as shown in [Fig. 37.30b](#).

37.9 Mission Applications

The research methodology used during the development of the HDRC3 architecture has been very much scenario-based where very challenging scenarios out of reach of current systems are specified and serve as longer-term goals to drive both theoretical and applied research. Most importantly, attempts are always made to close the theory/application loop by implementing and integrating results in AVs and deploying them for empirical testing at an early stage. One then iterates and continually increases the robustness and functionalities of the targeted components.

Due to the architecture described in the previous sections, it is relatively easy to build on top of existing functionalities and to add new ones in order to put together sophisticated autonomous missions. Below, two such example mission applications are described.

37.9.1 Emergency Services Assistance

The first application focuses again on the ambitious emergency services scenario discussed in the introduction. An emergency relief scenario in this context can be divided into two separate legs or parts.

37.9.1.1 Mission Leg I: Body Identification

In the first leg of the mission, a large region should be cooperatively scanned with one or more AVs to identify injured civilians. The result of this scan is a *saliency map* pinpointing potential victims, their locations, and associated information such as high resolution photos and thermal images. This information could then be used directly by emergency services or passed on to other AVs as a basis for additional tasks.

A multi-platform area coverage path-planning algorithm computes paths for n heterogeneous platforms guaranteeing complete camera coverage, taking into account sensor properties and platform capabilities.

Two video sources (thermal and color) are used, allowing for high rate human detection at larger distances than in the case of using the video sources separately with standard techniques. The high processing rate is essential in case of video collected onboard an AV in order not to miss potential victims.

A thermal image is first analyzed to find human body sized silhouettes. The corresponding regions in a color image are subjected to a human body classifier which is configured to allow weak classifications. This focus of attention allows

Algorithm 4 Saliency map construction

```

1: Initialize data structures
2: while scanning not finished do
3:   Simultaneously grab two images:  $img_{color}$ ,  $img_{thermal}$ 
4:   Analyze  $img_{thermal}$  to find potential human body regions
5:   for each region in  $img_{thermal}$  do
6:     Find corresponding region  $r_{color}$  in  $img_{color}$ 
7:     Compute geographical location  $loc$  of  $r_{color}$ 
8:     Execute human body classifier on  $r_{color}$ 
9:     if classification positive then
10:      if  $loc$  is new then
11:        add location  $loc$  to map, initialize certainty factor  $p_{body}(loc)$ 
12:      else
13:        update certainty factor  $p_{body}(loc)$ 
14:      end if
15:    end if
16:  end for
17: end while

```

for maintaining body classification at a rate up to 25 Hz. This high processing rate allows for collecting statistics about classified humans and pruning false classifications of the “weak” human body classifier. Detected human bodies are geolocalized on a map which can be used to plan supply delivery. The technique presented has been tested onboard the UASTL RMAX helicopter platform and is an important component in the lab’s research with autonomous search and rescue missions.

Saliency Map Construction. Information obtained from thermal and color video streams must be fused in order to create saliency maps of human bodies. An overview of the method used is presented in Algorithm 4. The execution of this algorithm starts when the host AV arrives at the starting position of the area to scan and is terminated when the scanning flight is finished. Its output is a set of geographical locations loc_i and certainty factors $p_{body}(loc_i)$. Refer to Doherty and Rudol (2007) and Rudol and Doherty (2008) for additional details.

After initializing the necessary data structures (line 1), the algorithm enters the main loop (line 2), which is terminated when the entire area has been scanned. The main loop begins with simultaneously grabbing two video frames. The thermal image is analyzed first (line 4) to find a set of regions of intensities which correspond to human body temperatures (details below). Then (line 6), for each of these subregions, a correspondence in the color frame, as well its geographical location loc , is calculated (details below). The calculated corresponding region in the color frame is analyzed with a human body classifier to verify the hypothesis that the location loc contains a human body (details below). If the classification is positive and the location loc has not been previously identified, then loc is added to the map and its certainty factor initialized (line 11). Otherwise, the certainty factor of that location is updated (line 13, details below).

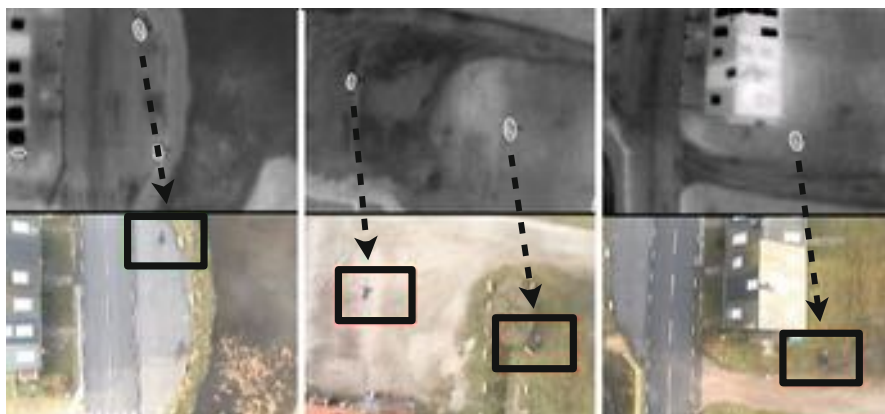


Fig. 37.31 Example input to the image processing algorithm: Thermal images and corresponding color images

Thermal Image Processing. The image processing algorithm takes a pair of images as input and starts by analyzing the thermal image (top row of Fig. 37.31). The image is first thresholded to find regions of certain intensities which correspond to human body temperature. The image intensity corresponding to a certain temperature is usually given by the camera manufacturer or can be calibrated by the user. The shapes of the thresholded regions are analyzed, and those which do not resemble a human body, due to the wrong ratio between minor and major axes of the fitted ellipse or due to incorrect sizes, are rejected. Once human body candidates are found in the thermal image, corresponding regions in the color image are calculated.

Image Correspondences and Geolocation. Finding corresponding regions using image registration or feature matching techniques is infeasible because of the different appearance of features in color and thermal images. Therefore, a closed form solution, which takes into account information about the cameras' pose in the world, is preferred. In short, a geographical location corresponding to pixel coordinates for one of the cameras is calculated. It takes into account the camera's extrinsic and intrinsic parameters and assumes a flat world. The obtained geographical location is then projected back into the other camera image.

The method can be extended to relax the flat world assumption given the elevation model. A geographical location of a target can be found by performing ray tracing along the line going from the camera center through a pixel to find the intersection with the ground elevation map.

The accuracy of the correspondence calculation is influenced by several factors. All inaccuracies of parameters involved in calculating the position and attitude of cameras in the world contribute to the overall precision of the solution. The evaluation of the accuracy involves investigating the AV state estimation errors, pan-tilt unit position accuracy, camera calibration errors, etc. In the case of the UASTL

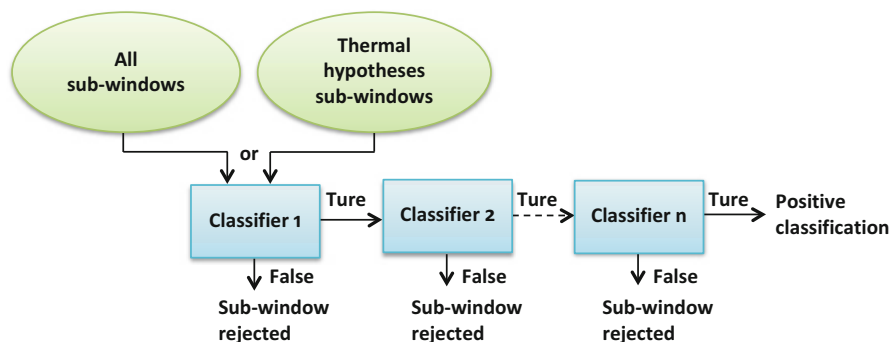


Fig. 37.32 Schematic description of a cascade classifier

RMAX, during the experimental validation, the corresponding image coordinates were within a subregion of 20 % of the video frame size (see the marked subregions in the color images in Fig. 37.31).

Human Body Classification. After calculating the coordinates of the pixel in the color image, a region with the P_c as center (the black rectangles in the bottom row images of Fig. 37.31) is analyzed by an object classifier. The classifier used was first suggested in Viola and Jones (2001). It uses a cascade of classifiers for object detection and includes a novel image representation, the integral image, for quick detection of features. Classifiers in the initial stages remove a large number of negative examples. Successive classifiers process a smaller number of sub-windows. The initial set of sub-windows can include all possible sub-windows or can be the result of a previous classification in a thermal image additionally improving processing rate. The method was also improved, for example, in Lienhart and Maydt (2002) by extending the original feature set.

The classifier requires training with positive and negative examples. During the learning process, the structure of a classifier is built using boosting. The use of a cascade of classifiers allows for dramatic speed up of computations by skipping negative instances and only computing features with high probability for positive classification. The speed up comes from the fact that the classifier, as it slides a window at all scales, works in stages and is applied to a region of interest until at some stage the candidate is rejected or all the stages are passed (see Fig. 37.32). This way, the classifier quickly rejects subregions that most probably do not include the features needed for positive classification (i.e., background processing is quickly terminated).

The implementation of the classifier used in this work is a part of the Open Source Computer Vision Library (<http://opencv.org/>), and the trained classifier for upper, lower, and full human body is a result of Kruppa et al. (2003). The classifier is best suited for pedestrian detection in frontal and backside views which is exactly the type of views an AV has when flying above the bodies lying on the ground.

The classifier parameters have been adjusted to minimize false negatives: In case of rescue operations, it is better to find more false positives than to miss a potential victim. The number of neighboring rectangles needed for successful identification has been set to 1 which makes the classifier accept very weak classifications. The search window is scaled up by 20 % between subsequent scans.

Map Building. Since the body classifier is configured to be “relaxed,” it delivers sporadic false-positive classifications. The results are pruned as follows. Every salient point in the map has two parameters which are used to calculate the certainty of a location being a human body: T_{frame} which describes the amount of time a certain location was in the camera view and T_{body} which describes the amount of time a certain location was classified as a human body. The certainty factor is calculated by $p_{body}(loc_i) = \frac{T_{body}}{T_{frame}}$. A location is considered a body if $p_{body}(loc_i)$ is larger than a certain threshold (0.5 was used during the flight tests) and T_{frame} is larger than a desired minimal observation time. Locations are considered equal if geographical distance between them is smaller than a certain threshold (depending on the geolocation accuracy) and the final value of a geolocated position is an average of the observations.

Experimental Validation. The presented technique for geolocation and saliency map building has been integrated as a perception functionality of the Control Kernel (Sect. 37.3.2) and tested in flight. A mission has then been constructed in the shape of a Task Specification Tree that indirectly makes use of the takeoff, hovering, path-following, and landing flight control modes (through FCL commands, Sect. 37.4.2.2), as well as the pan-tilt unit (through PPCL commands, Sect. 37.4.2.3).

Test flights were then performed at the Swedish Rescue Services Agency Test Area (Fig. 37.33a). Since this is a city-like closed area with road structures, buildings, etc., the video streams included different types of textures such as grass, asphalt, gravel, water, and building rooftops. An example complete push button mission setup was as follows:

- Two UASTL RMAX helicopters were used starting from H_1 and H_2 in Fig. 37.33b.
- An operator selected a rectangular area on a map where the saliency map was to be built (Fig. 37.33b).
- Onboard systems calculated the mission plan taking into account properties of the onboard sensors (such as the field of view) of both AVs. The plan consisted of two separate flight plans for the two AVs.
- The mission started with simultaneous takeoffs and flying to starting positions S_1 and S_2 in Fig. 37.33b. After arriving at the starting positions, the execution of the scanning paths autonomously began, and the saliency map building was initiated.
- Upon finishing the scanning paths at positions E_1 and E_2 , the AVs flew to the takeoff positions and performed autonomous landings.

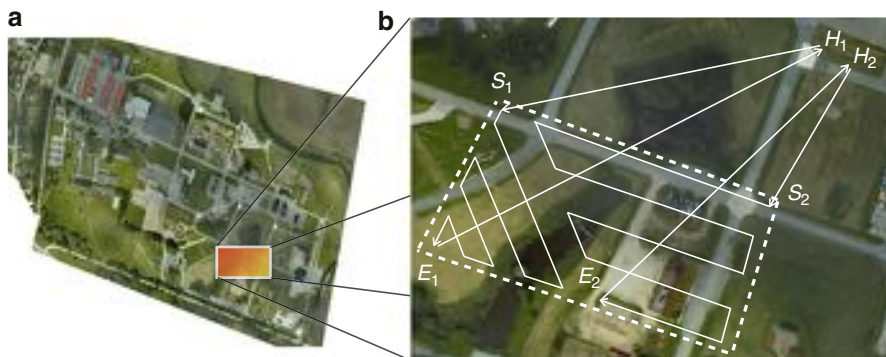


Fig. 37.33 (a) Map of the Swedish Rescue Services Agency Test Area in Revinge, (b) A closeup view of the area where the saliency map was built. Approximate flight paths are marked with *solid lines*

Experimental Results. All 11 human bodies placed in the area were found and geolocated. Corresponding color and thermal images for the identified objects are displayed vertically as pairs in Fig. 37.34. Images 7, 9, and 14 present three falsely identified objects, caused by configuring the human body classifier to accept weak classifications. A more restrictive setup could instead result in missing potential victims. The images could additionally be judged by a human operator to filter out the false-positive classifications. Both human bodies and dummies were detected despite the lower temperature of the latter.

Figure 37.35 presents the generated saliency map. The scanning pattern segments for the platform starting in position H_2 is marked with a solid line, and its final position is marked with a cross icon. The fields of view of the color and thermal cameras are depicted with light-gray and black rectangles, respectively. These differ slightly as the two cameras have different properties as identified during calibration procedures.

The circles indicate the identified body positions. The lighter the shade of the color, the more certain the classification. As can be seen, the most certain body positions are objects number 2 and 11 (in Fig. 37.34). It is due to the fact that these body images are clearly distinguishable from the homogeneous background. Nevertheless, even body images with more cluttered backgrounds were identified.

The accuracy of the body geolocation calculation was estimated by measuring GPS positions (without differential correction) of bodies after an experimental flight. The measurement has a bias of approximately 2 m in both east and north directions. It is the sum of errors in GPS measurement, accuracy of the camera platform mounting, PTU measurement, and camera calibration inaccuracies. The spread of measurement samples of approximately 2.5 m in both east and north directions is a sum of errors of the AV's attitude measurement, the system of springs in the camera platform, and time differences between the AV state estimate, PTU angle measurement, and image processing result acquisition.



Fig. 37.34 Images of classified bodies. Corresponding thermal images are placed under color images



Fig. 37.35 The resulting map with salient points marked as *circles*. The lighter the shade of the color, the higher the detection certainty

The presented algorithm requires only a single pair of images for human body classification. In practice, however, the more pairs available, the more certain the result of a classification can become. Additionally, thanks to using the results of the thermal image analysis to focus the classification in the color image subregions, a high rate of processing is achieved (above 20 Hz for the presented results).

37.9.1.2 Mission Leg II: Package Delivery

After successful completion of leg I of the mission scenario, one can assume that a saliency map has been generated with geolocated positions of the injured civilians. In the next phase of the mission, the goal is to deliver configurations of medical, food, and water supplies to the injured. In order to achieve this leg of the mission, one would require a task planner to plan for logistics, a motion planner to get one or more AVs to supply and delivery points, and an execution monitor to monitor the execution of highly complex plan operators. Each of these functionalities would also have to be tightly integrated in the system.

For these logistics missions, the use of one or more AVs with diverse roles and capabilities is assumed. Initially, there are n injured body locations, several supply depots, and several supply carrier depots (see Fig. 37.36). The logistics



Fig. 37.36 A supply depot (*left*) and a carrier depot (*right*)



Fig. 37.37 The AV logistics simulator

mission is comprised of one or more AVs transporting boxes containing food and medical supplies between different locations (Fig. 37.37). Plans are generated in the millisecond to seconds range using TALplanner (see Sect. 37.7.2), and empirical testing shows that this approach is promising in terms of integrating high-level deliberative capability with lower-level reactive and control functionality.



Fig. 37.38 Two frames of video presenting the prototype mechanism for carrying and releasing packages using an electromagnet. An *arrow* points to a package being carried. The *top left* picture presents the onboard camera view

Achieving the goals of such a logistics mission with full autonomy requires the ability to pick up and deliver boxes without human assistance. Thus, each AV has a device for attaching to boxes and carrying them beneath the AV. The action of picking up a box involves hovering above the box, lowering the device, attaching to the box, and raising the device, after which the box can be transported to its destination. There can also be a number of carriers, each of which is able to carry several boxes. By loading boxes onto such a carrier and then attaching to the carrier, the transportation capacity of an AV increases manifold over longer distances. The ultimate mission for the AVs is to transport the food and medical supplies to their destinations as efficiently as possible using the carriers and boxes at their disposal.

A physical prototype of a mechanism for carrying and releasing packages has been developed and tested. Figure 37.38 presents two images of the prototype system. The logistics scenario has also been tested in a simulated AV environment with hardware in-the-loop, where TALplanner generates a detailed mission plan which is then sent to a simulated execution system using the same helicopter flight control software as the physical AV. The execution monitor system has been tested in this simulation as well, with a large variety of deviations tested through fault injection in the simulation system. The simulator makes use of the Open Dynamics Engine (<http://www.ode.org>), a library for simulating rigid body dynamics, in order to realistically emulate the physics of boxes and carriers. This leads to effects such as boxes bouncing and rolling away from the impact point should they accidentally be dropped, which is also an excellent source of unexpected situations that can be used for validating both the domain model and the execution monitoring system.

37.9.2 Map Building Using a Laser Range Finder

The second application presented here deals with the construction of an elevation map using a laser range finder allowing AVs to navigate safely in unknown outdoor environments.

As described in Sect. 37.5.1, the path-planning algorithms used here are based on a geometrical description of the environment to generate collision-free paths. The safety of a UAS operation therefore depends on having a sufficiently accurate 3D model of the environment. Predefined maps may become inaccurate or outdated over time because of the environment changes, for example, due to new building structures and vegetation growth. Therefore, adequate sensors and techniques for updating or acquiring new 3D models of the environment are necessary in many cases.

Among the many sensors available for providing 3D information about an operational environment, laser range finders provide high-accuracy data at a reasonable weight and power consumption. One of the reasons for the innovation in this particular sensor technology is its wide use in many industries, but laser range finders have also received a great deal of interest from the robotics community, where their main usage is in navigation and mapping tasks for ground robotic systems, such as localization (Burgard et al. 1997), 2D Simultaneous Localization and Mapping (SLAM (Montemerlo and Thrun 2007)), 3D SLAM (includes 3D position (Cole and Newman 2006)), and 6D SLAM (includes 3D position and attitude (Nüchter et al. 2004)).

The device integrated with the UASTL RMAX system is the popular LMS-291 from SICK AG (<http://www.sick.com>). This unit does not require any reflectors or markers on the targets nor scene illumination to provide real-time measurements. It performs very well both in indoor and outdoor environments. The system is equipped with a rotating mirror which allows for obtaining distance information in one plane in front of the sensor with a selectable field of view of 100 or 180° (Fig. 37.39). It gives a resolution of 1 cm with a maximum range of 80 m, or a resolution of 1 mm with a range of 8 m; an angular resolution of 0.25, 0.5, or 1.0°; and a corresponding response time of 53, 26, or 13 ms.

The laser unit has been modified to reduce its weight from 4.5 to 1.8 kg. It has then been mounted on an in-house developed rotation mechanism supporting continuous rotation of the sensor around the middle laser beam (solid line in Fig. 37.39), which allows for obtaining half-sphere 3D point clouds even when the vehicle is stationary. A similar approach to the integration of the laser range finder with an UASTL RMAX platform is used by Whalley et al. (2008).

System Integration. The 3D map building algorithm based on the data provided by LRF sensor has been integrated as a perception functionality of the Control Kernel (Sect. 37.3.2) and tested in flight. A Task Specification Tree node has been specified to achieve the required mission goals (Sect. 37.4.3). Similar to the previous application example, the takeoff, hovering, path-following, and landing flight control modes were used during the mission through Flight Control Language

Fig. 37.39 Top view of the LMS-291 scanning field and the axis of rotation using the rotation mechanism

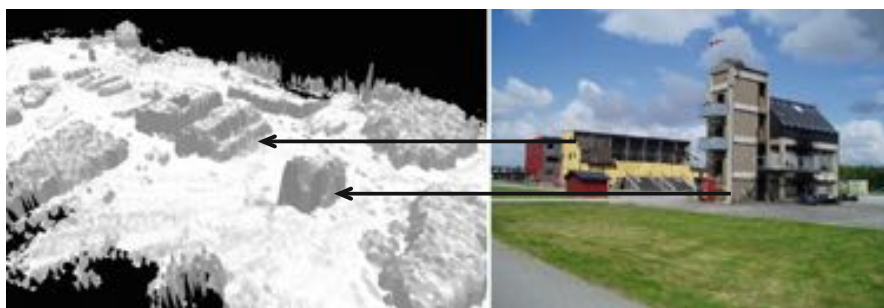
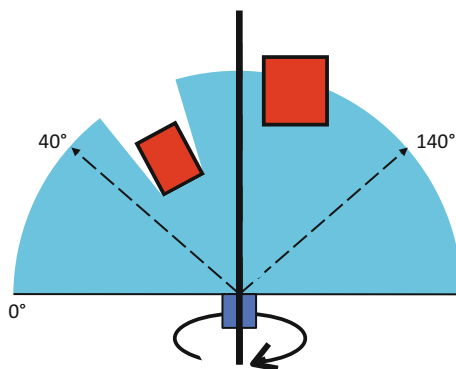


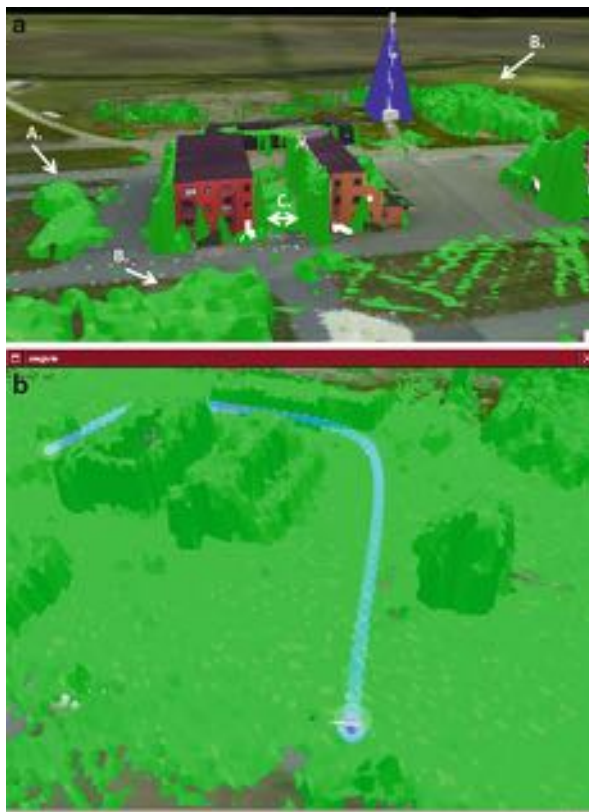
Fig. 37.40 Overview of the reconstructed elevation map of the Revinge flight test area based on the laser range finder data (*left*) and a photo of corresponding building structures (*right*)

commands (Sect. 37.4.2.2). Additionally, the Payload and Perception Control Language (Sect. 37.4.2.3) are used to set a constant angular speed of the LRF rotational mechanism as well as the distance and angular resolutions of the sensor. The 3D maps acquired are saved in the GIS Service database and available to the deliberative services (such as path planners, discussed in Sect. 37.5).

Experimental Results. Several missions were performed during which both the LRF data and AV state estimates were collected and integrated. Figure 37.40 presents a reconstructed elevation map of the Revinge flight test area (the left side) focusing on two building structures. A photo of corresponding buildings is presented on the right side of the figure. The elevation map is built by sampling the LRF data with 1 m resolution and constructing a set of triangles in order to represent the elevation.

In order to assess the fidelity of the newly generated model, an overlay with the existing model was generated. The result is presented in Fig. 37.41a. The new map

Fig. 37.41 Results of the map building procedure. (a) Overlay of the new elevation map with the existing Revinge flight test area model. (b) Example of path planner use in the reconstructed map of the Revinge area



includes some changes in the environment, including a metal container on the left in the figure (A.) and new vegetation (B.) that was not present in the existing model.

The new models stored by the GIS Service can be used by the AV platform for path planning in order to generate collision-free paths (see Sect. 37.5.1). Since generated models are based on noisy measurements, a safety margin during the planning is used. For the UASTL RMAX, a safety margin of 6 m is used. An example of path generated by the path planner using the new model is presented in Fig. 37.41b.

The accuracy of models built with the raw LRF point clouds (without applying the scan matching algorithms) is sufficient for navigation purposes if the necessary safety margins are used. The inaccuracies introduced by the measurement errors and the uncertainty of the AV state estimate might result in narrowing down the operational environment of the AV. For example, in Fig. 37.41a a narrow passage (C.) can be excluded from the collision-free space although the corridor between the two buildings is wide enough to fly through. Further investigation of methods for improving the model quality is ongoing at the time of writing this chapter.

37.10 Conclusions

The goal of this research is the development of autonomous intelligent systems for unmanned aircraft. This chapter has described a hybrid deliberative/reactive architecture which has been developed through the years and successfully deployed on a number of unmanned aircraft. The focus was on an instantiation of this architecture with the UASTL RMAX. The architecture was described in a manner that should be useful to the unmanned aircraft research community, because it isolates generic functionality and architectural solutions that can be transferred and used on existing unmanned aircraft and also those envisioned for the future. The experimental development of such systems has wider impact since the HDRC3 architecture and its instantiation on the UASTL RMAX are an example of a new type of software system commonly known as a *software intensive system* or *cyber physical system*. Such systems are characterized by a number of features: the complex combination of both hardware and software, the high degree of concurrency used, the distributed nature of the software, the network-centric nature of the environment in which they reside, the open nature of these systems, and finally, the requirement that they are both scalable and adaptive since specification of such systems is necessarily open due to their interaction with other components not part of the system. The latter features are particularly important in the context of collaborative missions with other heterogeneous systems.

The HDRC3 architecture encapsulates many of these features. Emphasis has been placed on the importance of clean transitional interfaces between the different layers of the architecture which are inhabited by computational processes with diverse requirements in terms of temporal latency in the decision cycles and the degree to which the processes use internal models. The HDRC3 architecture is also highly modular and extensible.

The use of HCSMs separates specification of the behavior of the system from the atomic conditions and activities which are implemented procedurally. This permits the use of a real-time interpreter for HCSMs and alleviates the requirement for recompilation when new HCSMs are added to the system. The approach is also amenable to distribution where federations of HCSM interpreters can intercommunicate with each other on a single platform as is the case with the UASTL RMAX system, but also across platforms if required.

The Platform Server provides a clean declarative interface to both the suite of flight control modes and perception control modes through the use of two languages, FCL and PPCL, respectively. This approach provides a primitive language of building blocks that can be structured and used by other functionalities in the system to define higher-level task specifications. Extending the suite of basic actions in the UASTL RMAX system is done by defining new HCSMs which interface to new or existing control modes and then extending the language interface in the Platform Server.

The task specifications themselves have a well-defined and extensible language, Task Specification Trees, which are used by both the reactive and deliberative layers of the architecture as a declarative specification language for tasks. Primitive actions

defined through the Platform Server interface can be used with TSTs, or if required, new actions can be defined by specifying new node types. The execution and procedural implementation of these basic action and new node types is separated from their declarative specification through the use of node executors and a node execution facility in the reactive layer of the HDRC3 architecture.

This clean separation has a number of advantages besides extensibility and modularity. A declarative specification of tasks permits the sharing of tasks implemented in different ways across platforms. This is especially important in the context of collaborative robotics and heterogeneous robotic systems. It also permits the translation of other task languages into a common representation. The translation of the output of automated task planners is a case in point. There is a direct mapping from the declarative output of a planner to grounded executable components in the architecture.

The DyKnow system is another extensible and modular functionality. Any source in an architecture can become a stream-based data source. One naturally thinks of sensors as specific sources of data, but at another extreme, one can think of another platform as a source of streamed data or a data base or the Internet. The DyKnow system provides means for specifying, constructing, generating, using, and managing these diverse sources of data contextually at many different levels of abstraction. This is a unique component in the HDRC3 architecture that has had widespread use by other functionalities such as the execution monitoring system and the task and motion planners.

Traditionally, verification and validation of autonomous systems has been a central research issue. In the context of unmanned aircraft, a great deal of research has focused on the Control Kernel and to some extent the lower parts of the reactive layer. In terms of the reactive layer itself and in particular the highly nondeterministic functionality associated with the deliberative layer, very little effort has been put into verification and validation because tools simply do not exist yet. A great deal of research effort has gone into improving this absence of effort during the development of key functionalities in the HDRC3 architecture. Many of the functionalities there, such as the task planner, the task specification language, the mission specification language, and the execution monitoring system, each have a formal semantics based on the use of Temporal Action Logics.

In fact the use of logic is highly integrated with many of the processes evoked by deliberative functionality. The execution monitoring system is essentially a real-time dynamic model-checking system where temporal models are generated dynamically by DyKnow during the course of operation, and various constraints are checked relative to those models by evaluating formulas through progression and checking for their satisfiability. The output of TALplanner is a formal narrative structure in Temporal Action Logic. Consequently both the process of generating a plan and the resulting output can be reasoned about using inference mechanisms associated with TALplanner.

In summary, this chapter has described an empirically tested unmanned aircraft architecture that combines many years of both engineering and scientific insight acquired through the successful deployment of the UASTL RMAX system in highly

complex autonomous missions. To the extent possible, these insights have been described in a generic manner in the hope that many of the functionalities described can serve as a basis for use by others in the continued and exciting development of such systems.

Acknowledgments This work is partially supported by the EU FP7 project SHERPA (grant agreement 600958); the Swedish Foundation for Strategic Research (SSF) Collaborative Unmanned Aircraft Systems (CUAS) project; the Swedish Research Council (VR) Linnaeus Center for Control, Autonomy, and Decision-making in Complex Systems (CADICS); and the ELLIIT network organization for Information and Communication Technology.

References

- J.S. Albus, F.G. Proctor, A reference model architecture for intelligent hybrid control systems, in *IFAC 13th Triennial World Congress*, San Francisco (IFAC, 1996), pp. 483–488
- R.C. Arkin, *Behavior-Based Robotics* (MIT, Cambridge, 1998)
- F. Bacchus, F. Kabanza, Planning for temporally extended goals. *Ann. Math. Artif. Intell.* **22**, 5–27 (1998)
- K. Ben Lamane, F. Kabanza, Reasoning about robot actions: a model checking approach, in *Revised Papers from the International Seminar on Advances in Plan-Based Control of Robotic Agents*, Dagstuhl Castle (Springer, 2002), pp. 123–139
- G. Berry, The Esterel synchronous programming language: design, semantics, implementation. *Sci. Comput. Program.* **19**, 87–152 (1992). doi:10.1016/0167-6423(92)90005-V
- G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide* (Addison-Wesley, Upper Saddle River, 2005). ISBN:978-0321267979
- R. Brooks, A robot that walks: emergent behaviors from a carefully evolved network, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Scottsdale, 1989, pp. 692–696
- A. Brooks, T. Kaupp, A. Makarenko, S. Williams, A. Örebäck, Towards component-based robotics, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, 2005
- W. Burgard, D. Fox, S. Thrun, Active mobile robot localization, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Nagoya (Morgan Kaufmann, 1997)
- D. Cole, P. Newman, Using laser range data for 3D SLAM in outdoor environments, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, 2006
- G. Conte, Navigation functionalities for an autonomous UAV helicopter. Licentiate thesis 1307, Department of Computer and Information Science, Linköping University, 2007
- G. Conte, Vision-based localization and guidance for unmanned aerial vehicles. Ph.D. thesis, Department of Computer and Information Science, Linköping University, Linköping, 2009
- G. Conte, P. Doherty, Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP J. Adv. Signal Process.* **2009**, 1–18 (2009)
- G. Conte, S. Duranti, T. Merz, Dynamic 3D path following for an autonomous helicopter, in *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, 2004
- J. Cremer, J. Kearney, Y. Papelis, HCSM: a framework for behavior and scenario control in virtual environments. *ACM Trans. Model. Comput. Simul.* **5**, 242–267 (1995)
- G. De Giacomo, R. Reiter, M. Soutchanski, Execution monitoring of high-level robot programs, in *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Trento (Morgan Kaufmann, 1998), pp. 453–465
- P. Doherty, J. Kvarnström, TALplanner: an empirical investigation of a temporal logic-based forward chaining planner, in *6th International Workshop on Temporal Representation and Reasoning (TIME-99)*, Orlando (IEEE Computer Society, 1999), pp. 47. ISBN:0-7695-0173-7

- P. Doherty, J. Kvarnström, TALplanner: a temporal logic-based planner. *Artif. Intell. Mag.* **22**(3), 95–102 (2001)
- P. Doherty, J. Kvarnström, Temporal action logics, in *The Handbook of Knowledge Representation*, chapter 18, ed. by F. Van Harmelen, V. Lifschitz, B. Porter (Elsevier, Amsterdam/Boston, 2008), pp. 709–757
- P. Doherty, J.-J.C. Meyer, On the logic of delegation: relating theory and practice, in *The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi*, ed. by F. Paglieri, L. Tummolini, R. Falcone, M. Miceli (College Publications, London, 2012)
- P. Doherty, P. Rudol, A UAV search and rescue scenario with human body detection and geolocalization, in *AI'07: Proceedings of the Australian Joint Conference on Advances in Artificial Intelligence*, Gold Coast (Springer, 2007), pp. 1–13. ISBN:3-540-76926-9, 978-3-540-76926-2
- P. Doherty, J. Gustafsson, L. Karlsson, J. Kvarnström, (TAL) temporal action logics: language specification and tutorial. *Electron. Trans. Artif. Intell.* **2**(3–4), 273–306 (1998). ISSN 1403–3534
- P. Doherty, P. Haslum, F. Heintz, T. Merz, P. Nyblom, T. Persson, B. Wingman, A distributed architecture for intelligent unmanned aerial vehicle experimentation, in *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, Toulouse, 2004
- P. Doherty, J. Kvarnström, F. Heintz, A temporal logic-based planning and execution monitoring framework for unmanned aircraft systems. *J. Auton. Agents Multi-Agent Syst.* **19**(3), 332–377 (2009)
- P. Doherty, F. Heintz, D. Landén. A distributed task specification language for mixed-initiative delegation, in *Principles and Practice of Multi-Agent Systems: 13th International Conference, PRIMA 2010, Kolkata, India, November 12–15, 2010, Revised Selected Papers*, ed. by N. Desai, A. Liu, M. Winikoff. *Lecture Notes in Computer Science*, vol. 7057 (Springer, Berlin/Heidelberg, 2010), pp. 42–57. ISBN: 978-3-642-25919-7, e-978-3-642-25920-3. DOI: 10.1007/978-3-642-25920-3_4
- P. Doherty, F. Heintz, D. Landén, A delegation-based architecture for collaborative robotics, in *Agent-Oriented Software Engineering XI: Revised Selected Papers*, Toronto. *Lecture Notes in Computer Science*, vol. 6788 (Springer, 2011), pp. 205–247
- P. Doherty, J. Kvarnström, A. Szalas, Temporal composite actions with constraints, in *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Rome, 2012, pp. 478–488
- P. Doherty, F. Heintz, J. Kvarnström, High-level mission specification and planning for collaborative unmanned aircraft systems using delegation. *Unmanned Syst.* **1**(1):75–119 (2013). ISSN:2301-3850, EISSN:2301-3869
- S. Duranti, G. Conte, In-flight identification of the augmented flight dynamics of the RMAX unmanned helicopter, in *Proceedings of the IFAC Symposium on Automatic Control in Aerospace*, Toulouse, 2007
- M. Egerstedt, X. Hu, A. Stotsky, Control of mobile platforms using a virtual vehicle approach. *IEEE Trans. Autom. Control* **46**(11), 1777–1782 (2001)
- J. Eker, J. Janneck, E. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, Y. Xiong, Taming heterogeneity – the Ptolemy approach. *Proc. IEEE* **91**, 127–144 (2003)
- E.A. Emerson, Temporal and modal logic, in *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, ed. by J. van Leeuwen (Elsevier/MIT, Amsterdam/Cambridge, 1990), pp. 997–1072
- J.L. Fernández, R.G. Simmons, Robust execution monitoring for navigation plans, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Victoria, 1998
- M. Fichtner, A. Grossmann, M. Thielscher, Intelligent execution monitoring in dynamic environments. *Fundam. Inform.* **57**(2–4), 371–392 (2003). ISSN:0169-2968
- R.J. Firby, An investigation into reactive planning in complex domains, in *Proceedings of the National Conference on Artificial Intelligence*, Seattle, 1987, vol. 1, pp. 202–206

- Foundation for Intelligent Physical Agents, FIPA communicative act library specification (2002), <http://www.fipa.org>
- E. Gat, On three-layer architectures, in *Artificial Intelligence and Mobile Robots*, ed. by D. Kortenkamp, R.P. Bonasso, R. Murphy (AAAI/MIT, Menlo Park, 1997)
- E. Gat, M.G. Slack, D.P. Miller, R.J. Firby, Path planning and execution monitoring for a planetary rover, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Cincinnati (IEEE Computer Society, 1990), pp. 20–25
- M. Ghallab, On chronicles: representation, on-line recognition and learning, in *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Cambridge, 1996, pp. 597–607. ISBN 1-55860-421-9
- S. Gottschalk, M.C. Lin, D. Manocha, OBBTree: a hierarchical structure for rapid interference detection, in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, New Orleans, 1996
- D. Harel, Statecharts: a visual formalism for complex systems. *Sci. Comput. Program.* **8**(3), 231–274 (1987)
- F. Heintz, DyKnow: a stream-based knowledge processing middleware framework. Ph.D. thesis, Department of Computer and Information Science, Linköping University, 2009
- F. Heintz, P. Doherty, DyKnow: an approach to middleware for knowledge processing. *J. Intell. Fuzzy Syst.* **15**(1), 3–13 (2004)
- F. Heintz, P. Doherty, DyKnow: a knowledge processing middleware framework and its relation to the JDL data fusion model. *J. Intell. Fuzzy Syst.* **17**(4), 335–351 (2006)
- F. Heintz, Z. Dragisic, Semantic information integration for stream reasoning, in *Proceedings of the International Conference on Information Fusion*, Singapore, 2012
- F. Heintz, P. Rudol, P. Doherty, From images to traffic behavior – a UAV tracking and monitoring application, in *Proceedings of the International Conference on Information Fusion*, Québec, 2007, pp. 1–8
- F. Heintz, J. Kvarnström, P. Doherty, Bridging the sense-reasoning gap: DyKnow – stream-based middleware for knowledge processing. *Adv. Eng. Inform.* **24**(1), 14–26 (2010)
- F. Heintz, J. Kvarnström, P. Doherty, Stream-based hierarchical anchoring. *Künstl. Intell.* **27**, 119–128 (2013). doi:10.1007/s13218-013-0239-2
- I. Horrocks, Ontologies and the semantic web. *Commun. ACM* **51**(12), (2008). ISSN:00010782, doi:10.1145/1409360.1409377
- L.E. Kavraki, P. Švestka, J. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996)
- M. Kleinehagenbrock, J. Fritsch, G. Sagerer, Supporting advanced interaction capabilities on a mobile robot with a flexible control system, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, 2004
- K. Konolige, Colbert: a language for reactive control in Sapphira, in *KI-97: Advances in Artificial Intelligence*, Freiburg. Lecture Notes in Computer Science, vol. 1303 (Springer, 1997), pp. 31–52
- K. Konolige, K. Myers, E. Ruspini, A. Saffiotti, The Sapphira architecture: a design for autonomy. *J. Exp. Theor. Artif. Intell.* **9**(2–3), 215–235 (1997)
- T. Koo, F. Hoffmann, F. Mann, H. Shim, Hybrid control of an autonomous helicopter, in *IFAC Workshop on Motion Control*, Grenoble, 1998
- R. Koymans, Specifying real-time properties with metric temporal logic. *Real-Time Syst.* **2**(4), 255–299 (1990)
- H. Kruppa, M. Castrillon-Santana, B. Schiele, Fast and robust face finding via local context, in *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France, 2003
- J.J. Kuffner, S.M. LaValle, RRT-connect: an efficient approach to single-query path planning, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, 2000, pp. 995–1001
- J. Kvarnström, TALplanner and other extensions to temporal action logic. Ph.D. thesis, Department of Computer and Information Science, Linköping University, 2005

- J. Kvarnström, Planning for loosely coupled agents using partial order forward-chaining, in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, Freiburg (AAAI, 2011), pp. 138–145. ISBN:978-1-57735-503-8, 978-1-57735-504-5
- J. Kvarnström, P. Doherty, TALplanner: a temporal logic based forward chaining planner. *Ann. Math. Artif. Intell.* **30**, 119–169 (2000)
- D. Landén, F. Heintz, P. Doherty, Complex task allocation in mixed-initiative delegation: A UAV Case Study, in *Principles and Practice of Multi-Agent Systems: 13th International Conference, PRIMA 2010, Kolkata, India, November 12–15, 2010, Revised Selected Papers*, ed. by N. Desai, A. Liu, M. Winikoff. *Lecture Notes in Computer Science* vol. 7057 (Springer, Berlin/Heidelberg, 2010), pp. 288–V303. ISBN: 978-3-642-25919-7, e-978-3-642-25920-3. DOI: 10.1007/978-3-642-25920-3_20
- S.M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, 2004)
- R. Lienhart, J. Maydt, An extended set of Haar-like features for rapid object detection, in *Proceedings of International Conference on Image Processing*, Rochester, 2002, pp. 900–903
- P. Mantegazza, E.L. Dozio, S. Papacharalambous, RTAI: real time application interface. *Linux J.* **2000**, (72es) (2000)
- G.H. Mealy, A method for synthesizing sequential circuits. *Bell Syst. Tech. J.* **34**(5), 1045–1079 (1955)
- T. Merz, Building a system for autonomous aerial robotics research, in *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, Lisbon, 2004
- T. Merz, S. Duranti, G. Conte, Autonomous landing of an unmanned aerial helicopter based on vision and inertial sensing, in *Proceedings of the International Symposium on Experimental Robotics*, Singapore, 2004
- T. Merz, P. Rudol, M. Wzorek, Control system framework for autonomous robots based on extended state machines, in *International Conference on Autonomic and Autonomous Systems*, Silicon Valley, 2006, 0:14. doi:<http://doi.ieeecomputersociety.org/10.1109/ICAS.2006.19>
- M. Montemerlo, S. Thrun, *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Springer Tracts in Advanced Robotics, vol. 27 (Springer, Berlin, 2007). ISBN:978-3-540-46402-0
- E.F. Moore, Gedanken-experiments on sequential machines, in *Automata Studies*, ed. by C.E. Shannon, J. McCarthy, W.R. Ashby (Princeton University Press, Princeton, 1956), pp. 129–153
- A. Nüchter, H. Surmann, K. Lingermann, J. Hertzberg, S. Thrun, 6D SLAM with an application in autonomous mine mapping, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, 2004, pp. 1998–2003
- P.-O. Pettersson, Using randomized algorithms for helicopter path planning. Licentiate thesis, Department of Computer and Information Science, Linköping University, 2006
- P.-O. Pettersson, P. Doherty, Probabilistic roadmap based path planning for an autonomous unmanned helicopter. *J. Intell. Fuzzy Syst.* **17**(4), 395–405 (2006)
- M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, ROS: an open-source robot operating system, in *ICRA Workshop on Open Source Software*, Kobe, 2009
- P. Rudol, P. Doherty, Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery, in *Proceedings of the IEEE Aerospace Conference*, Big Sky, 2008
- S. Sekhavat, P. Svestka, J.-P. Laumond, M.H. Overmars, Multi-level path planning for nonholonomic robots using semi-holonomic subsystems. *Int. J. Robot. Res.* **17**, 840–857 (1996)
- M.K. Smith, C. Welty, D.L. McGuinness, OWL Web Ontology Language Guide, 2004. See <http://www.w3.org/TR/owl-guide/>.
- P. Viola, M.J.-C. Jones, Rapid object detection using a boosted cascade of simple features, in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Kauai, 2001
- M. Whalley, G. Schulein, C. Theodore, Design and flight test results for a hemispherical LADAR developed to support unmanned rotorcraft urban operations research, in *American Helicopter Society 64th Annual Forum*, Montréal, 2008

-
- F. White, A model for data fusion, in *Proceedings of the National Symposium for Sensor Fusion*, Orlando, 1988
- M. Wzorek, Selected aspects of navigation and path planning in unmanned aircraft systems. Licentiate thesis 1509, Department of Computer and Information Science, Linköping University, 2011
- M. Wzorek, G. Conte, P. Rudol, T. Merz, S. Duranti, P. Doherty, From motion planning to control – a navigation framework for an unmanned aerial vehicle, in *Proceedings of the Bristol International Conference on UAV Systems*, Bristol, 2006a
- M. Wzorek, D. Landen, P. Doherty, GSM technology as a communication media for an autonomous unmanned aerial vehicle, in *Proceedings of the Bristol International Conference on UAV Systems*, Bristol, 2006b

Ivan Maza, Aníbal Ollero, Enrique Casado,
and David Scarlatti

Contents

| | | |
|--------|--|-----|
| 38.1 | Introduction | 954 |
| 38.2 | General Concepts and Classification | 955 |
| 38.2.1 | Coordination and Cooperation | 955 |
| 38.2.2 | Communication and Networking | 956 |
| 38.2.3 | Classification of Multi-UAV Architectures | 957 |
| 38.2.4 | General Model for Each UAV in the Team | 959 |
| 38.3 | Physical Coupling: Joint Load Transportation | 961 |
| 38.4 | Vehicle Formations and Coordinated Control | 964 |
| 38.5 | Swarms | 966 |
| 38.6 | Intentional Cooperation Schemes | 967 |
| 38.7 | UAVs Networked with Sensors and Actuators in the Environment | 970 |
| 38.8 | Conclusions | 972 |
| | References | 972 |

Abstract

This chapter presents a classification of different schemes for the cooperation of multiple UAVs, taking into account the coupling between the vehicles and the type of cooperation. Then, the research and development activities in load

I. Maza (✉)

Robotics, Vision and Control Group, University of Seville, Seville, Spain

e-mail: imaza@us.es

A. Ollero

Robotics, Vision and Control Group, University of Seville, Seville, Spain

Center for Advanced Aerospace Technologies (CATEC), Parque Tecnológico y Aeronáutico de Andalucía, La Rinconada, Spain

e-mail: aollero@us.es

E. Casado • D. Scarlatti

Boeing Research & Technology Europe, Madrid, Spain

e-mail: Enrique.Casado@boeing.com; David.Scarlatti@boeing.com

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_119,

© Springer Science+Business Media Dordrecht 2015

953

transportation, formation control, swarm approaches, and intentional cooperation architectures are revised. The chapter also considers UAVs networked with other elements in the environment to support their navigation and, in general, their operation. The chapter refers theoretical work but also emphasizes practical field outdoor demonstrations involving aerial vehicles.

38.1 Introduction

This chapter considers the cooperation of multiple UAVs performing jointly missions such as search and rescue, reconnaissance, surveying, detection and monitoring in dangerous scenarios, exploration and mapping, and hazardous material handling. The coordination of a team of autonomous vehicles allows to accomplish missions that no individual autonomous vehicles can accomplish on its own. Team members can exchange sensor information, collaborate to track and identify targets, perform detection and monitoring activities (Ollero and Maza 2007), or even actuate cooperatively in tasks such as the transportation of loads.

The advantages of using multiple UAVs when comparing to a single powerful one can be categorized as follows:

- *Multiple simultaneous interventions.* A single autonomous vehicle is limited at any one time to sense or actuate in a single point. However, the components of a team can simultaneously collect information from multiple locations and exploit the information derived from multiple disparate points to build models that can be used to take decisions. Moreover, multiple UAVs can apply simultaneously forces at different locations to perform actions that could be very difficult for a single UAV.
- *Greater efficiency.* The execution time of missions such as exploration and searching for targets can be decreased when using simultaneously multiple vehicles.
- *Complementarities of team members.* Having a team with multiple heterogeneous vehicles offers additional advantages due to the possibility of exploiting their complementarities. Thus, for example, ground and/or aerial vehicles with quite different characteristics and onboard sensors can be integrated in the same platform. For instance, the aerial vehicles could be used to collect information from locations that cannot be reached by the ground vehicles, while these ground members of the team could be equipped with heavy actuators. Then, the aerial and ground vehicles could be specialized in different roles. But even considering the UAVs themselves complementarities can be found: the fixed-wing airplanes typically have longer flight range and time of flight, whereas helicopters have vertical take-off and landing capability, better maneuverability, and therefore can hover to obtain detailed observations of a given target.
- *Reliability.* The multi-UAV approach leads to redundant solutions offering greater fault tolerance and flexibility including reconfigurability in case of failures of individual vehicles.
- *Technology evolution.* The development of small, relatively low-cost UAVs is fuelled by the progress of embedded systems together with the developments on

technologies for integration and miniaturization. Furthermore, the progress on communication technologies experienced in the last decade plays an important role in multiple vehicle systems.

- *Cost.* A single vehicle with the performance required to execute some tasks could be an expensive solution when comparing to several low-cost vehicles performing the same task. This is clear for UAVs and particularly in small-size, light, and low-cost versions, where constraints such as power consumption, weight, and size play an important role.

Section 38.2 of this chapter will deal with the general concepts and contains a rough classification of systems with multiple autonomous UAVs. Then, the joint load transportation, formation control, swarm approaches, and teams with intentional cooperation are examined in more detail along Sects. 38.3–38.6. The networking of UAVs with other sensors and actuators in the environment is considered in Sect. 38.7. Finally, Sect. 38.8 concludes the chapter.

38.2 General Concepts and Classification

In the first part of this section, the concepts of coordination and cooperation are briefly presented due to their relevance in any system with multiple autonomous vehicles. Then, the important role of communications in these systems is summarized. Finally, a classification based on the coupling between the vehicles is outlined.

38.2.1 Coordination and Cooperation

In platforms involving multiple vehicles, the concepts of coordination and cooperation play an important role. In general, the coordination deals with the sharing of resources, and both temporal and spatial coordination should be considered. The temporal coordination relies on synchronization among the different vehicles, and it is required in a wide spectrum of applications. For instance, for objects monitoring, several synchronized perceptions of the objects could be required. In addition, spatial coordination of UAVs deals with the sharing of the space among them to ensure that each UAV will be able to perform safely and coherently regarding the plans of the other UAVs and the potential dynamic and/or static obstacles. Some formulations are based on the extension of robotics path planning concepts. In this context, the classical planning algorithms for a single robot with multiple bodies (Latombe 1990; LaValle 2006) may be applied without adaptation for centralized planning (assuming that the state information from all the UAVs is available). The main concern, however, is that the dimension of the state space grows linearly in the number of UAVs. Complete algorithms require time that is at least exponential in dimension, which makes them unlikely candidates for such problems. Sampling-based algorithms are more likely to scale well in practice when there are many UAVs, but the resulting dimension might still be too high. For such cases, there

are also decoupled path planning approaches such as the prioritized planning that considers one vehicle at a time according to a global priority.

Cooperation can be defined as a “joint collaborative behavior that is directed toward some goal in which there is a common interest or reward” (Barnes and Gray 1991). According to Cao et al. (1997), given some task specified by a designer, a multiple-robot system displays cooperative behavior if, due to some underlying mechanism (i.e., the “mechanism of cooperation”), there is an increase in the total utility of the system. The cooperation of heterogeneous vehicles requires the integration of sensing, control, and planning in an appropriated decisional architecture. These architectures can be either centralized or decentralized depending on the assumptions on the knowledge’s scope and accessibility of the individual vehicles, their computational power, and the required scalability. A centralized approach will be relevant if the computational capabilities are compatible with the amount of information to process and the exchange of data meets both the requirements of speed (up-to-date data) and expressivity (quality of information enabling well-informed decision-taking). On the other hand, a distributed approach will be possible if the available knowledge within each distributed vehicle is sufficient to perform “coherent” decisions, and this required amount of knowledge does not endow the distributed components with the inconveniences of a centralized system (in terms of computation power and communication bandwidth requirements). One way to ensure that a minimal global coherence will be satisfied within the whole system is to enable communication between the vehicles of the system, up to a level that will warranty that the decision is globally coherent. One of the main advantages of the distributed approach relies on its superior suitability to deal with the scalability of the system.

38.2.2 Communication and Networking

It should be noticed that communication and networking also play an important role in the implementation of these schemes for multiple unmanned vehicles. Single vehicle communication systems usually have an unshared link between the vehicle and the control station. The natural evolution of this communication technique toward multi-vehicle configurations is the star-shaped network configuration. While this simple approach to vehicles intercommunication may work well with small teams, it could not be practical or cost-effective as the number of vehicles grows. Thus, for example, in multi-UAV systems, there are some approaches of a wireless heterogeneous network with radio nodes mounted at fixed sites, on ground vehicles, and in UAVs. The routing techniques allow any two nodes to communicate either directly or through an arbitrary number of other nodes which act as relays. When autonomous teams of UAVs should operate in remote regions with little/no infrastructure, using a mesh of ground stations to support communication between the mobile nodes is not possible. Then, networks could be formed in an ad hoc fashion, and the information exchanges occur only via the wireless networking equipment carried by the individual UAVs. Some autonomous configurations

(such as close formation flying) result in relatively stable topologies. However, in others, rapid fluctuations in the network topology may occur when individual vehicles suddenly veer away from one another or when wireless transmissions are blocked by terrain features, atmospheric conditions, signal jamming, etc. In spite of such dynamically changing conditions, vehicles in an autonomous team should maintain close communications with others in order to avoid collisions and facilitate collaborative team mission execution. In order to reach these goals, two different approaches have been adopted. One, closer to the classical networks architecture, establishes a hierarchical structure and routes data in the classical down-up-down traversing as many levels of the hierarchy as needed to reach destination. The other prospective direction to assist routing in such an environment is to use location information provided by positioning devices, such as global positioning systems (GPS), thus using what it is called location aware protocols. These two techniques are compatible and can be mixed. For example, some of the levels in a hierarchical approach could be implemented using location aware methods.

38.2.3 Classification of Multi-UAV Architectures

Multi-UAV systems can be classified from different points of view. One possible classification is based on the coupling between the UAVs (see Fig. 38.1):

1. *Physical coupling.* In this case, the UAVs are connected by physical links and then their motions are constrained by forces that depend on the motion of other UAVs. The lifting and transportation of loads by several UAVs lies in this category and will be addressed in Sect. 38.3 of this chapter. The main problem is the motion-coordinated control, taking into account the forces constraints. From the point of view of motion planning and collision avoidance, all the members of the team and the load can be considered as a whole. As the number of vehicles is usually low, both centralized and decentralized control architectures can be applied.
2. *Formations.* The vehicles are not physically coupled, but their relative motions are strongly constrained to keep the formation. Then, the motion planning problem can be also formulated considering the formation as a whole. Regarding the collision avoidance problem within the team, it is possible to embed it in the formation control strategy. Scalability properties to deal with formations of many individuals are relevant, and then, decentralized control architectures are usually preferred. Section 38.4 of the chapter will deal with the formations and will also show how the same techniques can be applied to control coordinated motions of vehicles even if they are not in formation.
3. *Swarms.* They are homogeneous teams of many vehicles which interactions generate emerging collective behaviors. The resulting motion of the vehicles does not lead necessarily to formations. Scalability is the main issue due to the large number of vehicles involved, and then pure decentralized control architectures are mandatory. Section 38.5 of the chapter will be devoted to the swarm approaches.

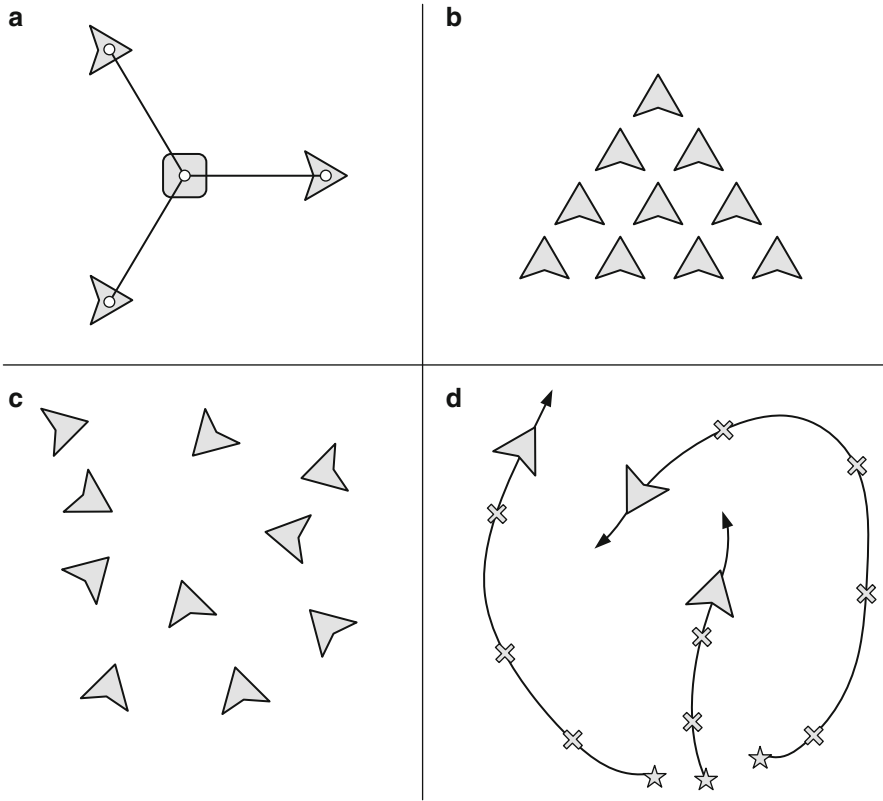


Fig. 38.1 Graphical illustration of a possible classification for multiple UAVs systems: (a) physical coupling (3 UAVs transporting one object), (b) formations, (c) swarms, and (d) team executing tasks represented by crosses following an intentional cooperation approach. The UAVs are represented by *gray arrows*

4. *Intentional cooperation.* The UAVs of the team move according to trajectories defined by individual tasks that should be allocated to perform a global mission (Parker 1998). These UAV trajectories typically are not geometrically related as in the case of the formations. This cooperation will be considered in Sect. 38.6 of this chapter. In this case, problems such as multi-UAV task allocation, high-level planning, plan decomposition, and conflict resolution should be solved, taking into account the global mission to be executed and the different UAVs involved. In this case, both centralized and decentralized decisional architectures can be applied.

In the rest of sections of this chapter, each type of multi-vehicle system is discussed in further detail. But before to proceed with each one, a general model for each UAV of the team is presented. This model can be particularized to fit any of the types of the above classification, as it will be shown later.

38.2.4 General Model for Each UAV in the Team

Let us consider a team of UAVs that plan their actions according to a set of coordination and cooperation rules R . In particular, it is assumed that the set R includes k possible tasks $\Omega = \{\tau_1, \tau_2, \dots, \tau_k\}$ with n logical conditions requiring a change of task in the current plan. Let $E = \{e_1, e_2, \dots, e_n\}$ be a set of discrete events associated with n logical conditions requiring a change of task during the execution. Each task has a set of m parameters $\Pi = \{\pi^1, \pi^2, \dots, \pi^m\}$ defining its particular characteristics.

Systems composed of a physical plant and a decisional and control engine implementing such kind of cooperation rules R can be modeled as hybrid systems (Fierro et al. 2002; Chaimowicz et al. 2004; Fagiolini et al. 2007; Li et al. 2008). Figure 38.2 shows a simplified hybrid model that summarizes the different interactions that can be found in each member of the classification presented above.

The i -th UAV's current task has a discrete dynamics $\delta : \Omega \times E \rightarrow \Omega$, i.e.,

$$\tau_i^+ = \delta(\tau_i, e_i), \quad (38.1)$$

where $e_i \in E$ is an event (internal or external) requiring a change of task from τ_i to τ_i^+ , both from the set of tasks Ω .

It should be noticed that each task can have a different control algorithm or a different set of parameters for the same controller. The control reconfiguration is triggered in the transition between tasks associated to different events. Event activation is generated by

$$e_i = \Phi(q_i, \varepsilon_i, X_i, \bar{\mu}_i), \quad (38.2)$$

where ε_i represents the internal events (such as changes in the execution states of the tasks) and $\bar{\mu}_i$ is a vector $\bar{\mu}_i = (\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_{N_m}})$ containing the messages coming from N_m UAVs cooperating with the i -th UAV. Those messages are used, for example, in the negotiation processes involved in the intentional cooperation mechanisms and are generated onboard each UAV by a decisional module Δ (see Fig. 38.2). This module encompasses high-level reasoning and planning, synchronization among different UAVs, negotiation protocols for task allocation and conflict resolution purposes, task management and supervision, complex task decomposition, etc.

Regarding the perception of the environment, a database ED with “a priori” knowledge about the environment, including static obstacles, objects of interest, and threats can be available and updated with the new information gathered during the mission. On the other hand, object detection and localization (Merino 2007) is usually required in many applications. The state x of the object to be tracked obviously includes its position p , and for moving objects, it is also convenient to add the velocity \dot{p} into the kinematic part of the state to be estimated. But further information is needed in general. An important objective in some missions is to

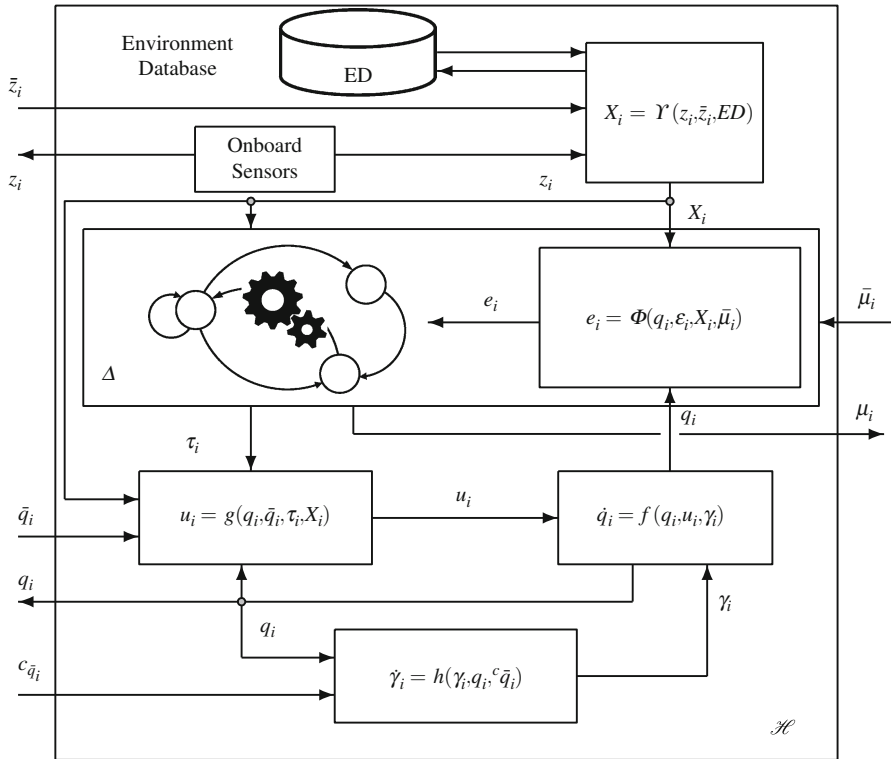


Fig. 38.2 General blocks and interactions considered in the hybrid model for each UAV described in Sect. 38.2

confirm that an object belongs to a certain class within a set Ξ (for instance, in the case of fire alarms detection, this set will include as classes of fire alarms and false alarms). Therefore, the state will include information regarding the classification of the object. Also, in certain applications, some appearance information could be needed to characterize an object, which also can help in the task of data association between different UAVs with different cameras. Additionally, this information could even include the 3D volume of the object that can be added to the obstacles database. In general, the appearance information is static and will be represented by θ .

The complete dynamic state to be estimated is composed by the status of all the objects, N_o , and the number of objects can vary with the time. The state estimated by the i -UAV at time t is then represented by the vector $x_i = [x_{i1}^T, \dots, x_{iN_o}^T]^T$. Each potential object m is defined by $x_{im} = [p_{im} \dot{p}_{im} \theta_{im}]^T$. The information about the objects will be inferred from all the measurements z_i from the sensors onboard the UAVs and \bar{z}_i gathered by the fleet of N_s UAVs that can communicate with the i -th UAV $\{\bar{z}_j, j = 1, \dots, N_s\}$. The latter vector can be completed with the measurements from sensors located around the environment, such as static

surveillance cameras, or nodes from wireless sensor networks (WSNs) deployed in the area of interest. Notice that z_i also contains the forces/torques derived from the interaction with the environment that are measured with the sensors onboard.

On the other hand, let $q_i \in \Theta$ be a vector describing the state of the i -th UAV taking values in the configuration space Θ , and let $\tau_i \in \Omega$ be the task τ that the i -th UAV is currently executing. This UAV's configuration q_i has a continuous dynamics

$$\dot{q}_i = f(q_i, u_i, \gamma_i), \quad (38.3)$$

where $u_i \in \mathcal{U}$ is a control input and $\gamma_i \in \Gamma$ models the dynamics associated to the possible physical interaction with the environment and/or other UAVs

$$\dot{\gamma}_i = h(\gamma_i, q_i, {}^c\bar{q}_i), \quad (38.4)$$

with vector ${}^c\bar{q}_i = (q_{i_1}, q_{i_2}, \dots, q_{i_{N_c}})$ containing the configurations of the N_c neighbors physically connected to the i -th UAV. Then $\gamma_i \neq 0$ only if there is physical interaction.

Regarding u_i , it is a feedback law generated by a low-level controller $g : \Theta \times \bar{\Theta} \times \Omega \times \Upsilon \rightarrow \mathcal{U}$, i.e.,

$$u_i = g(q_i, \bar{q}_i, \tau_i, X_i), \quad (38.5)$$

so that the UAV's trajectory $q_i(t)$ corresponds to the desired current task τ_i , taking into account the configurations of the N neighbors $\bar{q}_i = (q_{i_1}, q_{i_2}, \dots, q_{i_N})$ with influence in the control of the i -th UAV. This influence can be found, for example, in the control problem of swarms and formations. On the other hand, Eq. (38.5) also includes the vector $X_i \in \Upsilon$, taking values in the environment model space Υ , that encompasses estimations about forces/torques derived from the interaction with the environment, targets to the tracked, obstacles detected during the mission and/or known "a priori," threats to be avoided, etc.

In conclusion, the hybrid dynamics \mathcal{H} of the i -th UAV shown in Fig. 38.2 has \bar{z}_i , $\bar{\mu}_i$, \bar{q}_i , and ${}^c\bar{q}_i$ as inputs and z_i , μ_i , and q_i as outputs. This diagram is not intended to be exhaustive or to cover all the possible architectures and existing systems. Instead, it is aimed at providing a general overview of all the possible interactions in order to put into context the approaches presented in the next sections of the chapter.

38.3 Physical Coupling: Joint Load Transportation

The transportation of a single object by multiple autonomous vehicles is a natural extension of the moving by several persons of a large and heavy object that cannot be handled by a single person. The coordinated control of the motion of each vehicle should consider the involved forces induced by the other vehicles and the load itself. Thus, in the scheme depicted in Fig. 38.2, there is a term $\gamma_i \neq 0$ modelling those forces, which is taken into account in the design of the controller in Eq. 38.5.

It should be also mentioned that γ_i can be measured using onboard sensors. For instance, in the case of several UAVs transporting a load using ropes, a force sensor in the rope can provide a measurement of the influence of the other UAVs and the load being transported. Each UAV could be controlled around a common compliance center attached to the transported object. Under the assumption that each UAV holds the object firmly with rigid links, the real trajectories of all of the UAVs are equal to the real trajectory of the object. However, in some transportation problems, this assumption cannot be applied, and the transported object moves with a dynamic behavior that can be expressed by means of Eq. 38.4.

A suitable approach for the required coordinated control is the leader-follower scheme that will be more detailed in the next section. In this scheme, the desired trajectory is the trajectory of the leader. The followers estimate the motion of the leader by themselves through the motion of the transported object. Several examples of this approach can be found in the robotics community. The leader-follower scheme extended to multiple followers and to robots with non-holonomic constraints (Kosuge and Sato 1999) has been implemented in an experimental system with three tracked mobile robots with a force sensor. In Sugar and Kumar (2002), the decentralized control of cooperating mobile manipulators is studied with a designated lead robot being responsible for task planning. The control of each robot is decomposed (mechanically decoupled) into the control of the gross trajectory and the control of the grasp. The excessive forces due to robot positioning errors and odometry errors are accommodated by the compliant arms. In Huntsberger et al. (2004), distributed coordinated control of two rovers carrying a 2.5 m long mockup of a photovoltaic tent is presented and demonstrated as an example of the CAMPOUT behavior-based control architecture. Borenstein (2000) details the OmniMate system, which uses a compliant linkage platform between two differential drive mobile robots (Labmate) that provide a loading deck for up to 114 kg of payload.

Lifting and transportation of loads by using multiple helicopters has been also a research topic for many years motivated by the payload constraints of these vehicles and the high cost of helicopters with significant payload. In addition, the use of multiple manned helicopters is also problematic and only simple operations, like load transportation with two helicopters, can be performed by extremely skillful and experienced pilots. The level of stress is usually very high, and practical applications are therefore rarely possible. Load transportation and deployment by one and several helicopters is very useful for many applications including the delivery of first-aid packages to isolated victims in disasters (floods, earthquakes, fires, industrial disasters, and many others) and is also a basic technology for other future applications: the building of platforms for evacuation of people in rescue operations and the installation of platforms in uneven terrains for landing of manned and unmanned VTOL aircrafts. This later application would first require the installation of the supporting units defining the horizontal surface and later the installation of the surface itself.

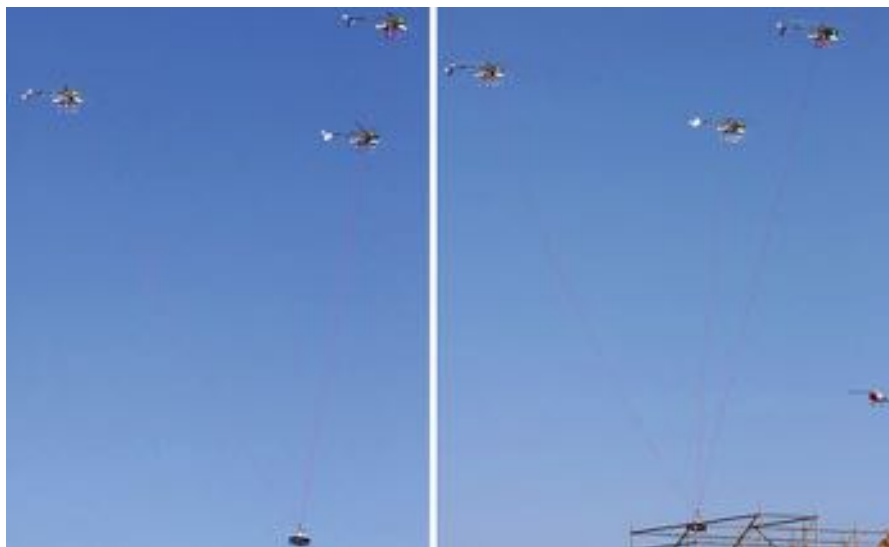


Fig. 38.3 Three autonomous helicopters from the Technical University of Berlin (*TUB-H* model) transporting a wireless camera to the top floor of a building with a height of 12 m in May 2009. A device onboard each helicopter is equipped with a force sensor to estimate the influence of the other helicopters and the load itself – term γ_i in Eq. 38.4. The images show the mission during the actual load transportation (*left*) and shortly before the load deployment (*right*). A fourth helicopter which was used to acquire airborne video footage of the mission is visible on the *right*

The autonomous lifting and transportation by two helicopters (twin lift) has been studied since the beginning of the nineties by means of nonlinear adaptive control (Mittal et al. 1991) and H_∞ control (Reynolds and Rodriguez 1992). In Lim et al. (1999) an interactive *modeling, simulation, animation, and real-time control* (MoSART) tool to study the twin lift helicopter system is presented. However, only simulation experiments have been found until December 2007, when lifting and transportation of a load by means of three autonomous helicopters (Bernard and Kondak 2009) was demonstrated experimentally in the framework of the AWARE project. After that first successful test, the load transportation system was used again in 2009 to deploy a camera on the roof of a building with a height of 12 m (see Fig. 38.3) in the framework of the same project (Bernard et al. 2011). Notice that in this case, the physical coupling between UAS are involved through direct interactions of each unmanned aerial vehicle with the joint load.

Small-size single or multiple autonomous quadrotors are also considered for load transportation and deployment in Michael et al. (2011), Palunko et al. (2012), and Sreenath et al. (2013). Dynamically coupled quadrotors should cooperate safely to transport load, in contrast to the existing results on formation control of decoupled multi-UAV systems that are addressed in the next section.

38.4 Vehicle Formations and Coordinated Control

In the formations, the members of the group of vehicles must keep user-defined distances with the other group members. The control problem consists of maintaining these user-defined distances, and consequently the configurations of the N neighbors $\bar{q}_i = (q_{i1}, q_{i2}, \dots, q_{iN})$ in the formation should be taken into account in the control law (see Eq. 38.5). Those configurations can be either received via inter-vehicle communication or estimated using the sensors onboard. Anyway, formation control involves the design of distributed control laws with limited and disrupted communication, uncertainty, and imperfect or partial measurements.

Vehicle formation is a basic strategy to perform multi-vehicle missions including searching and surveying, exploration and mapping, active reconfigurable sensing systems, and space-based interferometry. An added advantage of the formation paradigm is that new members can be introduced to expand or upgrade the formation or to replace a failed member. The stability of the formation has been studied by many researchers that have proposed robust controllers to provide insensitivity to possibly large uncertainties in the motion of nearby agents, transmission delays in the feedback path, and the consideration of the effect of quantized information.

The close formation flight control of homogeneous teams of fixed-wing UAVs airplanes received attention in the last 10 years. The large group formation of small UAVs also offers benefits in terms of drag reduction and then increased payoffs in the ability to maintain persistent coverage of a large area. Both linear (Giulietti et al. 2000) and nonlinear control laws (Schumacher and Singh 2000) have been proposed and tested in simulation. However, practical implementations are still very scarce. In How et al. (2004) a demonstration of two fixed-wing UAVs simultaneous flying, the same flight plan (tracking waypoints in open-loop formation) is reported. In the same paper, two UAVs were linked to the same receding horizon trajectory planner, and independent timing control was performed about the designed plans.

The leader-follower approach mentioned in the previous section has been also used to control general formations where the desired positions of followers are defined relative to the actual state of a leader. It should be noted that every formation can be further divided into simplest leader-follower schemes. Then, in this approach, some vehicles are designated as leaders and track predefined trajectories, while the followers track transformed versions of these trajectories according to given schemes. In the leader-follower approach, path planning only needs to be performed in the leader workspace. The leader-follower pattern is adopted in Yun et al. (2010) to maintain a fixed geometrical formation of unmanned helicopters while navigating following certain trajectories. The leader is commanded to fly on some predefined trajectories, and each follower is controlled to maintain its position in formation using the measurement of its inertial position and the information of the leader position and velocity, obtained through a wireless modem. In Gu et al. (2006) two-aircraft formation flights confirmed the performance of a formation

controller designed to have an inner- and outerloop structure, where the outerloop guidance control laws minimized the forward, lateral, and vertical distance error by controlling the engine propulsion and generating the desired pitch and roll angles to be tracked by the innerloop controller. In the formation flight configuration, a radio control pilot maintained ground control of the leader aircraft, while the autonomous follower aircraft maintained a predefined position and orientation with respect to the leader aircraft. The leader-follower approach is applied in Galzi and Shtessel (2006) in the design of robust and continuous controllers to achieve collision-free path-tracking formation in the presence of unknown bounded disturbances acting on each UAV. In Bayraktar et al. (2004) an experiment with two fixed-wing UAVs is presented. The leader UAV was given a predetermined flight plan, and the trajectory of the UAV was updated once per second in real time through the ground station to keep the follower at a fixed distance offset from the leader. Finally, the vehicles platooning can be considered as a particular case consisting of a leader followed by vehicles in a single row. Both lateral and longitudinal control to keep the safe headway, and lateral distance should be considered. The simplest approach relies on individual vehicle control from the data received from the single immediate front vehicle (Bom et al. 2005).

Other methods are based on a virtual leader, a moving reference point whose purpose is to direct, herd, and/or manipulate the vehicle group behavior. The lack of a physical leader among the vehicles implies that any vehicle is interchangeable with any other in the formation. A solution based on a virtual leader approach combined with an extended local potential field is presented in Paul et al. (2008) for formation flight and formation reconfiguration of small-scale autonomous helicopters. And for fixed-wing models, experimental results with YF-22 research aircrafts can be found in Campa et al. (2007), validating the performance of a formation control law using also a virtual leader configuration.

Practical applications of formation control should include a strategy for obstacle avoidance and reconfiguration of the formation. The avoidance of big obstacles could be performed by changing the trajectory of the whole formation to go around the obstacle or to pass through a narrow tunnel (Desai et al. 2001). If the obstacles are smaller than the size of the formation, the vehicles should be able to compromise the formation until the obstacle is passed. In order to do so, the obstacle avoidance behavior should be integrated in the control strategy of the individual members of the formation to avoid/bypass obstacles. Hybrid control techniques have been applied to avoid obstacles and solve the formation reconfiguration (Zelinski et al. 2003).

Formation is not the only cooperative scheme for UAVs in applications such as exploration and mapping. The cooperation of multiple UAVs can be also examined from the point of view of the intentionality to achieve a given mission. Then, according to Parker (1998), it is possible to distinguish between intentional cooperation and swarm-type cooperation. Those approaches are considered in the following two sections:

38.5 Swarms

The key concept in the swarms is that complex collective global behaviors can arise from simple interactions between large numbers of relatively unintelligent agents. This swarm cooperation is based on concepts from biology (Sharkey 2006) and typically involves a large number of homogeneous individuals, with relatively simple sensing and actuation, and local communication and control that collectively achieve a goal. This can be considered as a bottom-up cooperation approach. It usually involves numerous repetitions of the same activity over a relatively large area. The agents execute the same program and interact only with other nearby agents by measuring distances and exchanging messages.

Thus, according to Fig. 38.2 the configurations of the N neighbors $\bar{q}_i = (q_{i_1}, q_{i_2}, \dots, q_{i_N})$ should be considered as well as the messages $\bar{\mu}_i = (\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_{N_m}})$ coming from N_m UAVs cooperating with the i -th UAV. Nevertheless, it should be mentioned that depending on the particular communication and sensing capabilities of the UAVs in the swarm, simplified mechanisms based on partial or imperfect information could be required. For example, the estimation of the full vector \bar{q}_i is not possible in many swarm-based systems, and partial information such as the distances with the neighbors is the only measurement available. The same is applicable to the messages interchanged that can range from data packets sent through wireless links to simple visual signals based on lights of different colors.

The concept of operations for a micro-UAV system is adopted from nature from the appearance of flocking birds, movement of a school of fish, and swarming bees among others. This “emergent behavior” is the aggregate result of many simple interactions occurring within the flock, school, or swarm. Exploration of this emergent behavior in a swarm is accomplished through a high-performance computing parallel discrete event simulation in Corner and Lamont (2004). In Kube and Zhang (1993) different mechanisms that allow populations of behavior-based robots to perform collectively tasks without centralized control or use of explicit communication are presented. Mataric (1992) provides the results of implementing group behaviors such as dispersion, aggregation, and flocking on a team of robots. In Kovacina et al. (2002) a rule-based, decentralized control algorithm that relies on constrained randomized behavior and respects UAV restrictions on sensors, computation, and flight envelope is presented and evaluated in a simulation of an air vehicle swarm searching for and mapping a chemical cloud within a patrolled region. Another behavior-based decentralized control strategy for UAV swarming by using artificial potential functions and sliding-mode control technique is presented in Han et al. (2008). Individual interactions for swarming behavior are modeled using the artificial potential functions. For tracking the reference trajectory of the swarming of UAVs, a swarming center is considered as the object of control. The sliding-mode control technique is adopted to make the proposed swarm control strategy robust with respect to the system uncertainties and varying mission environment.

The bio-inspired motivation of swarms can be found, for example, in Zhang et al. (2007), which describes an adaptive task assignment method for a team of fully distributed vehicles with initially identical functionalities in unknown task environments. The authors employ a simple self-reinforcement learning model inspired by the behavior of social insects to differentiate the initially identical vehicles into “specialists” of different task types, resulting in stable and flexible division of labor; on the other hand, in dealing with the cooperation problem of the vehicles engaged in the same type of task, the so-called *ant system* algorithm was adopted to organize low-level task assignment. Dasgupta (2008) presents a multiagent-based prototype system that uses swarming techniques inspired from insect colonies to perform automatic target recognition using UAVs in a distributed manner within simulated scenarios. In Altshuler et al. (2008) a swarm of UAVs is used for searching one or more evading targets, which are moving in a predefined area while trying to avoid a detection by the swarm (Cooperative Hunters problem). By arranging themselves into efficient geometric flight configurations, the UAVs optimize their integrated sensing capabilities, enabling the search of a maximal territory.

In general, the above approaches deal with homogeneous teams without explicit consideration of tasks decomposition and allocation, performance measures, and individual efficiency constraints of the members of the team. Those aspects are considered in the intentional cooperation schemes described in the next section.

38.6 Intentional Cooperation Schemes

In the intentional cooperation approaches, each individual executes a set of tasks (subgoals that are necessary for achieving the overall goal of the system and that can be achieved independently of other subgoals) explicitly allocated to perform a given mission in an optimal manner according to planning strategies (Parker 1998). The UAVs cooperate explicitly and with purpose but also has the limitation of independent subgoals: If the order of task completion is mandatory, additional explicit knowledge has to be provided to state ordering dependencies in the preconditions. It is also possible to follow a design based on “collective” interaction, in which entities are not aware of other entities in the team, yet they do share goals, and their actions are beneficial to their teammates (Parker 2008).

Key issues in these systems include determining which UAV should perform each task (task allocation problem) so as to maximize the efficiency of the team and ensuring the proper coordination among team members to allow them to successfully complete their mission. In order to solve the multi-robot task allocation problem, some metrics to assess the relevance of assigning given tasks to particular robots are required. In Gerkey and Matarić (2004) a domain-independent taxonomy for the multiagent task allocation problem is presented. In the last years, a popular approach to solve this problem in a distributed way is the application of market-based negotiation rules. An usual implementation of those distributed negotiation rules (Botelho and Alami 1999; Dias and Stenz 2002; Gerkey and Matarić 2002) is

based on the Contract Net Protocol (Smith 1980). In those approaches, the messages $\bar{\mu}_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{iN_m})$ coming from N_m UAVs cooperating with the i -th UAV are those involved in the negotiation process: announce a task, bid for a task, allocate a task, ask for the negotiation token, etc.

Once the tasks have been allocated, it is necessary to coordinate the motions of the vehicles, which can be done by means of suitable multi-vehicle path/velocity planning strategies, as mentioned in Sect. 38.2. The main purpose is to avoid potential conflicts among the different trajectories when sharing the same working space. It should be mentioned that even if the vehicles are explicitly cooperating through messages, a key element in many motion coordination approaches is the updated information about the configurations of the N neighbors $\bar{q}_i = (q_{i1}, q_{i2}, \dots, q_{iN})$. Formal approaches to the collision avoidance problem and different approaches that can be applied to solve it can be found in LaValle (2006) and Latombe (1990).

On the other hand, teams composed by heterogeneous members involve challenging aspects, even for the intentional cooperation approach. In Ollero and Maza (2007) the current state of the technology, existing problems, and potentialities of platforms with multiple UAVs (with emphasis on systems composed by heterogeneous UAVs) are studied. This heterogeneity is twofold: firstly in the UAV platforms looking to exploit the complementarities of the aerial vehicles, such as helicopters and airships, and secondly in the information-processing capabilities onboard, ranging from pure remotely teleoperated vehicles to fully autonomous aerial robots.

The multi-UAV coordination and control architecture developed in the European COMETS project (Gancet et al. 2005) was demonstrated for the autonomous detection and monitoring of fires (Ollero and Maza 2007) by using two helicopters and one airship (see Fig. 38.4). Regarding teams involving aerial and ground vehicles, the CROMAT architecture also implemented cooperative perception and task allocation techniques (Viguria et al. 2010) that have been demonstrated in fire detection, monitoring, and extinguishing. Multiagent (combined ground and air) tasking and cooperative target localization have been also demonstrated recently (Hsieh et al. 2007) as well as multi-target tracking (ground vehicles) with a micro-UAV (He et al. 2010).

In Maza et al. (2011) a distributed architecture for the autonomous coordination and cooperation of multiple UAVs for civil applications is presented. The architecture is endowed with different modules that solve the usual problems that arise during the execution of multipurpose missions, such as task allocation, conflict resolution, and complex task decomposition. One of the main objectives in the design of the architecture was to impose few requirements to the execution capabilities of the autonomous vehicles to be integrated in the platform. Basically, those vehicles should be able to move to a given location and activate their payload when required. Thus, heterogeneous autonomous vehicles from different manufacturers and research groups can be integrated in the architecture developed, making it easily usable in many multi-UAV applications. The software implementation of the architecture was tested in simulation and finally validated in field experiments with four autonomous helicopters. The validation process included several multi-UAV missions for civil applications in a simulated urban setting: surveillance applying

Fig. 38.4 Coordinated flights in the COMETS project involving an airship and two autonomous helicopters



the strategies for multi-UAV cooperative searching presented in Maza and Ollero (2007); fire confirmation, monitoring, and extinguishing; load transportation and deployment with single and multiple UAVs; and people tracking.

Finally, cooperative perception can be considered as an important tool in many applications based on intentional cooperation schemes. It can be defined as the task of creating and maintaining a consistent view of a world containing dynamic objects by a group of agents each equipped with one or more sensors. Thus, a team of vehicles can simultaneously collect information from multiple locations and exploit the information derived from multiple disparate points to build models that can be used to take decisions. In particular, cooperative perception based on artificial vision has become a relevant topic in the multi-robot domain, mainly in structured environments (Thrun 2001; Schmitt et al. 2002). In Merino et al. (2006) cooperation perception methods for multi-UAV system are proposed. Each UAV extracts knowledge, by applying individual perception techniques, and the overall cooperative perception is performed by merging the individual results. This approach requires knowing the relative position and orientation of the UAVs. In many outdoor applications, it is assumed that the position of all the UAVs can be obtained by means of GPS and broadcasted through the communication system. However, if this is not the case, the UAVs should be capable of identifying and of localizing each other (Konolige et al. 2003) which could be difficult with the onboard sensors. Another approach consists of identifying common objects in the scene. Then, under certain assumptions, the relative pose displacement between the vehicles can be computed from these correspondences. In Merino et al. (2006) this strategy has been demonstrated with heterogeneous UAVs. In the ANSER project (see, e.g., Sukkarieh et al. 2003), decentralized sensor data fusion using multiple aerial vehicles is also researched and experimented with fixed-wing UAVs with navigation and terrain sensors.

38.7 UAVs Networked with Sensors and Actuators in the Environment

The development of wireless communication technologies in the last 10 years makes possible the integration of autonomous vehicles with the environment infrastructure. Particularly, the integration with wireless sensor and actuator networks is very promising. The benefit of this integration can be seen from two different points of view:

- The use of UAVs to complement the information collected by the wireless sensor network (WSN), to perform as mobile “data mules,” to act as communication relays, to improve the connectivity of the network, and to repair it in case of malfunctioning nodes.
- The use of WSNs as an extension of the sensorial capabilities of the UAVs. In this case, the information about the objects in the environment will be inferred from all the measurements z_i from the sensors onboard and \bar{z}_i gathered by the fleet of N_s UAVs and nodes that can communicate with the i -th UAV $\{\bar{z}_j, j = 1, \dots, N_s\}$.

Static wireless sensor networks have important limitations as far as the required coverage and the short communication range in the nodes are concerned. The use of mobile nodes could provide significant improvements. Thus, they can provide the ability to dynamically adapt the network to environmental events and to improve the network connectivity in case of static nodes failure. Node mobility for ad hoc and sensor networks has been studied by many researchers (Grossglauser and Tse 2002; Venkitasubramaniam et al. 2004). Moreover, mobile nodes with single-hop communication and the ability to recharge batteries, or refueling, have been proposed as data mules of the network, gathering data while they are near of fixed nodes and saving energy in static node communications (Jain et al. 2006). The coordinated motion of a small number of nodes in the network to achieve efficient communication between any pair of other mobile nodes has been also proposed.

An important problem is the localization of the nodes of a WSN. This is an open problem because GPS-based solutions in all the nodes are usually not viable due to the cost, the energy consumption, and the satellite visibility from each node. In Caballero et al. (2008) a probabilistic framework for the localization of an entire WSN based on a vehicle is presented. The approach takes advantage of the good localization capabilities of the vehicle and its mobility to compute estimation of the static nodes positions by using the signal strength of the messages interchanged with the network.

However, in many scenarios, the motion of the mobile nodes installed on ground vehicles or carried by persons is very constrained, due to the characteristics of the terrain or the dangerous conditions involved, such as in civil security and disaster scenarios. The cooperation of aerial vehicles with the ground wireless sensor network offers many potentialities. The use of aircrafts as data sinks when they fly over the fixed sensor networks following a predictable pattern in order to



Fig. 38.5 Sensor deployment from an autonomous helicopter in the AWARE project experiments carried out in 2009

gather data from them has been proposed by several authors in the WSN community. In Corke et al. (2003) an algorithm for path computation and following is proposed and applied to guide the motion of an autonomous helicopter flying very close to the sensor nodes deployed on the ground.

It should be noticed that flight endurance and range of the currently available low-cost UAVs is very constrained (Ollero and Merino 2004). Moreover, reliability and fault tolerance is a main issue in the cooperation of the aerial vehicles. Furthermore, these autonomous vehicles need communication infrastructure to cooperate or to be teleoperated by humans in emergency conditions. Usually this infrastructure is not available, or the required communication range is too large for the existing technology. Then, the deployment of this communication infrastructure is a main issue. In the same way, in most wireless sensor networks projects, it is assumed that the wireless sensor network has been previously fully deployed without addressing the problems to be solved when the deployment is difficult. Moreover, in the operation of the network, the infrastructure could be damaged or simply the deployment is not efficient enough. Then, the problem is the repairing of the coverage or the connectivity of the network by adding suitable sensor and communication elements. In Corke et al. (2004) the application of an autonomous helicopter for the deployment and repairing of a wireless sensor network is proposed. This approach has been also followed in the AWARE project (Maza et al. 2010), whose platform has self-deployment and self-configuration features for the operation in sites without sensing and communication infrastructure. The deployment includes not only wireless sensors (see Fig. 38.5) but also heavier loads such as communication equipment that require the transportation by using several helicopters (see Fig. 38.3).

38.8 Conclusions

The concepts of coordinated and cooperative control of multiple UAVs deserved significant attention in the last years in the control, robotics, artificial intelligence, and communication communities. The implementation of these concepts involve integrated research in the control, decision, and communication areas. For instance, the communication and networking technologies play an important role in the practical implementation of any multi-vehicle system. Thus, the integrated consideration of communication and control problems is a relevant research and development topic.

This chapter has first reviewed the existing work on the transportation of a single load by different autonomous vehicles. In order to solve this problem, control theory based on models of the vehicles and their force interactions have been applied. The chapter also studied formation control. In this problem, the application of control theory based on models of the vehicles is dominant. However, behavior-based approaches that do not use these models have been also demonstrated. The work on swarms has been also reviewed. Approaches inspired in biology and multiagent systems are common. The problems are typically formulated for large number of individuals, but up to now, the practical demonstrations involve few physical UAVs. The intentional task-oriented cooperation of robotic vehicles, possibly heterogeneous, has been also addressed. The task allocation problem and the path planning techniques play an important role here, as well as the application of cooperative perception methods.

Finally, the chapter has explored the integration and networking of one or many UAVs with sensors and actuators in the environment pointing out the benefits of this integration. The self-deployment of the network and the motion planning to maintain quality of service are promising approaches that have been preliminarily studied but still require significant attention.

References

- Y. Altshuler, V. Yanovsky, I. Wagner, A. Bruckstein, Efficient cooperative search of smart targets using UAV swarms. *Robotica* **26**(4), 551–557 (2008)
- D. Barnes, J. Gray, Behaviour synthesis for co-operant mobile robot control, in *International Conference on Control*, Edinburgh, 1991, vol. 2, pp. 1135–1140
- S. Bayraktar, G.E. Fainekos, G.J. Pappas, Experimental cooperative control of fixed-wing unmanned aerial vehicles, in *Proceedings of the IEEE Conference on Decision and Control*, Atlantis, Paradise Island, the Bahamas, 2004
- M. Bernard, K. Kondak, Generic slung load transportation system using small size helicopters, in *Proceedings of the International Conference on Robotics and Automation*, Kobe, Japan (IEEE, 2009), pp. 3258–3264
- M. Bernard, K. Kondak, I. Maza, A. Ollero, Autonomous transportation and deployment with aerial robots for search and rescue missions. *J. Field Robot.* **28**(6), 914–931 (2011)
- J. Bom, B. Thuilot, F. Marmoint, P. Martinet, Nonlinear control for urban vehicles platooning, relying upon a unique kinematic GPS, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 4138–4143
- J. Borenstein, The OmniMate: a guidewire- and beacon-free AGV for highly reconfigurable applications. *Int. J. Prod. Res.* **38**(9), 1993–2010 (2000)

- S.C. Botelho, R. Alami, M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, 1999, vol. 2, pp. 1234–1239
- F. Caballero, L. Merino, P. Gil, I. Maza, A. Ollero, A probabilistic framework for entire WSN localization using a mobile robot. *Robot. Auton. Syst.* **56**(10), 798–806 (2008)
- G. Campa, Y. Gu, B. Seanor, M. Napolitano, L. Pollini, M. Fravolini, Design and flight-testing of non-linear formation control laws. *Control Eng. Pract.* **15**(9), 1077–1092 (2007)
- Y.U. Cao, A.S. Fukunaga, A. Kahng, Cooperative mobile robotics: antecedents and directions. *Auton. Robots* **4**(1), 7–27 (1997)
- L. Chaimowicz, V. Kumar, M.F.M. Campos, A paradigm for dynamic coordination of multiple robots. *Auton. Robots* **17**(1), 7–21 (2004)
- P. Corke, R. Peterson, D. Rus, Networked robots: flying robot navigation using a sensor net, in *Proceedings of the International Symposium of Robotic Research*, Siena, 2003
- P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, G. Sukhatme, Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle, in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004, pp. 3602–3608
- J.J. Corner, G.B. Lamont, Parallel simulation of UAV swarm scenarios, in *Proceedings of the 36th Conference on Winter Simulation, WSC'04*, Washington, DC, USA, 2004, pp. 355–363
- P. Dasgupta, A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **38**(3), 549–563 (2008)
- J.P. Desai, J.P. Ostrowski, V. Kumar, Modeling and control of formations of nonholonomic mobile robots. *IEEE Trans. Robot. Autom.* **17**(6), 905–908 (2001)
- M.B. Dias, A. Stenz, Opportunistic optimization for market-based multirobot control, in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, 2002, pp. 2714–2720
- A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, A. Bicchi, Decentralized intrusion detection for secure cooperative multi-agent systems, in *Proceedings of the IEEE International Conference on Decision and Control*, New Orleans, LA, USA, 2007, pp. 1553–1558
- R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J.P. Ostrowski, G. Pappas, C.J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, B. Southall, A framework and architecture for multi-robot coordination. *Int. J. Robot. Res.* **21**(10–11), 977–995 (2002)
- D. Galzi, Y. Shtessel, UAV formations control using high order sliding modes, in *Proceedings of the American Control Conference*, Minneapolis, MN, USA, 2006, vol. 2006, pp. 4249–4254
- J. Gancet, G. Hattenberger, R. Alami, S. Lacroix, Task planning and control for a multi-UAV system: architecture and algorithms, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 1017–1022
- B. Gerkey, M. Mataric, Sold!: auction methods for multi-robot coordination. *IEEE Trans. Robot. Autom.* **18**(5), 758–768 (2002)
- B. Gerkey, M. Mataric, A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **23**(9), 939–954 (2004)
- F. Giulietti, L. Pollini, M. Innocenti, Autonomous formation flight. *IEEE Control Syst. Mag.* **20**(6), 34–44 (2000)
- M. Grossglauser, D.N.C. Tse, Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.* **10**(4), 477–486 (2002)
- Y. Gu, B. Seanor, G. Campa, M. Napolitano, L. Rowe, S. Gururajan, S. Wan, Design and flight testing evaluation of formation control laws. *IEEE Trans. Control Syst. Technol.* **14**(6), 1105–1112 (2006)
- K. Han, J. Lee, Y. Kim, Unmanned aerial vehicle swarm control using potential functions and sliding mode control. *Proc. Inst. Mech. Eng., Part G: J. Aerosp. Eng.* **222**(6), 721–730 (2008)

- R. He, A. Bachrach, N. Roy, Efficient planning under uncertainty for a target-tracking micro-aerial vehicle, in *IEEE International Conference on Robotics and Automation, 2010*, Kobe, Japan, 2010
- J. How, E. King, Y. Kuwata, Flight demonstrations of cooperative control for UAV teams, in *Proceedings of the AIAA 3rd Unmanned-Unlimited Technical Conference, Workshop, and Exhibit*, Chicago, 2004, vol. 1, pp. 505–513
- M.A. Hsieh, L. Chaimowicz, A. Cowley, B. Grocholsky, J.F. Keller, V. Kumar, C.J. Taylor, Y. Endo, R.C. Arkin, B. Jung, D.F. Wolf, G. Sukhatme, D.C. MacKenzie, Adaptive teams of autonomous aerial and ground robots for situational awareness. *J. Field Robot.* **24**(11), 991–1014 (2007)
- T.L. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, P.S. Schenker, P. Pirjanian, H.D. Nayar, Distributed control of multi-robot systems engaged in tightly coupled tasks. *Auton. Robots* **17**(1), 79–92 (2004)
- S. Jain, R.C. Shah, W. Brunette, G. Borriello, S. Roy, Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Netw. Appl.* **11**(3), 327–339 (2006)
- K. Konolige, D. Fox, B. Limketkai, J. Ko, B. Stewart, Map merging for distributed robot navigation, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, 2003, pp. 212–217
- K. Kosuge, M. Sato, Transportation of a single object by multiple decentralized-controlled non-holonomic mobile robots, in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Kyongju, Korea, 1999, vol. 3, pp. 1681–1686
- M. Kovacina, D. Palmer, G. Yang, R. Vaidyanathan, Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles, in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002, vol. 3, pp. 2782–2788
- C.R. Kube, H. Zhang, Collective robotics: from social insects to robots. *Adapt. Behav.* **2**(2), 189–218 (1993)
- J.C. Latombe, *Robot Motion Planning* (Kluwer, 1990)
- J.C. Latombe, *Robot Motion Planning* (Kluwer Academic Publishers, Boston, MA, 1991)
- S.M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, 2006) Available at <http://planning.cs.uiuc.edu/>
- H. Li, F. Karray, O. Basir, I. Song, A framework for coordinated control of multiagent systems and its applications. *IEEE Trans. Syst. Man Cybern., Part A: Syst. Hum.* **38**(3), 534–548 (2008)
- C. Lim, R.P. Metzger, A. Rodriguez, Interactive modeling, simulation, animation and real-time control (MoSART) twin lift helicopter system environment, in *Proceedings of the American Control Conference*, San Diego, CA, USA, 1999, vol. 4, pp. 2747–2751
- M.J. Matarić, Designing emergent behaviors: from local interactions to collective intelligence, in *From Animals to Animats 2, 2nd International Conference on Simulation of Adaptive Behavior (SAB-92)*, ed. by J.-A. Meyer, H. Roitblat, S. Wilson (MIT, Cambridge, Honolulu, Hawaii, USA, 1992), pp. 432–441
- I. Maza, A. Ollero, Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, in *Distributed Autonomous Robotic Systems 6*, ed. by R. Alami, R. Chatila, H. Asama. Volume 6 of Distributed Autonomous Robotic Systems (Springer, Tokyo/New York, 2007), pp. 221–230
- I. Maza, K. Kondak, M. Bernard, A. Ollero, Multi-UAV cooperation and control for load transportation and deployment. *J. Intell. Robot. Syst.* **57**(1–4), 417–449 (2010)
- I. Maza, F. Caballero, J. Capitan, J.M. de Dios, A. Ollero, A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities. *J. Field Robot.* **28**(3), 303–328 (2011)
- L. Merino, Cooperative perception techniques for multiple unmanned aerial vehicles: applications to the cooperative detection, localization and monitoring of forest fires. PhD thesis, Dpto. Ingenieria de Sistemas y Automatica – University of Seville

- L. Merino, F. Caballero, J.M. de Dios, J. Ferruz, A. Ollero, A cooperative perception system for multiple UAVs: application to automatic detection of forest fires. *J. Field Robot.* **23**(3–4), 165–184 (2006)
- N. Michael, J. Fink, V. Kumar, Cooperative manipulation and transportation with aerial robots. *Auton. Robots* **30**(1), 73–86 (2011)
- M. Mittal, J.V.R. Prasad, D.P. Schrage, Nonlinear adaptive control of a twin lift helicopter system. *IEEE Control Syst. Mag.* **11**(3), 39–45 (1991)
- A. Ollero, I. Maza (eds.), *Multiple Heterogeneous Unmanned Aerial Vehicles*. Springer Tracts on Advanced Robotics (Springer, Berlin/New York, 2007)
- A. Ollero, L. Merino, Control and perception techniques for aerial robotics. *Annu. Rev. Control* **28**(2), 167–178 (2004)
- I. Palunko, P. Cruz, R. Fierro, Agile load transportation: safe and efficient load manipulation with aerial robots. *IEEE Robot. Autom. Mag.* **19**(3), 69–79 (2012)
- L. Parker, ALLIANCE: an architecture for fault-tolerant multi-robot cooperation. *IEEE Trans. Robot. Autom.* **14**(2), 220–240 (1998)
- L. Parker, Distributed intelligence: overview of the field and its application in multi-robot systems. *J. Phys. Agents* **2**(1), 5–14 (2008)
- T. Paul, T. Krogstad, J. Gravdahl, Modelling of UAV formation flight using 3D potential field. *Simul. Model. Pract. Theory* **16**(9), 1453–1462 (2008)
- H.K. Reynolds, A.A. Rodriguez, H_∞ control of a twin lift helicopter system, in *Proceedings of the 31st IEEE Conference on Decision and Control*, Tucson, AZ, USA, 1992, pp. 2442–2447
- T. Schmitt, R. Hanek, M. Beetz, S. Buck, B. Radig, Cooperative probabilistic state estimation for vision-based autonomous mobile robots. *IEEE Trans. Robot. Autom.* **18**, 670–684 (2002)
- C. Schumacher, S. Singh, Nonlinear control of multiple UAVs in close-coupled formation flight, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, USA, 2000, pp. 14–17
- A.J.C. Sharkey, Robots, insects and swarm intelligence. *Artif. Intell. Rev.* **26**(4), 255–268 (2006)
- G. Smith, The Contract Net Protocol: high-level communication and control in a distributed problem solver. *IEEE Trans. Comput.* **29**(12), 1104–1113 (1980)
- K. Sreenath, N. Michael, V. Kumar, Trajectory generation and control of a quadrotor with a cable-suspended load – a differentially-flat hybrid system, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013
- T.G. Sugar, V. Kumar, Control of cooperating mobile manipulators. *IEEE Trans. Robot. Autom.* **18**(1), 94–103 (2002)
- S. Sukkariéh, E. Nettleton, J.-H. Kim, M. Ridley, A. Goktogan, H. Durrant-Whyte, The ANSER project: data fusion across multiple uninhabited air vehicles. *Int. J. Robot. Res.* **22**(7–8), 505–539 (2003)
- S. Thrun, A probabilistic online mapping algorithm for teams of mobile robots. *Int. J. Robot. Res.* **20**(5), 335–363 (2001)
- P. Venkatasubramaniam, S. Adireddy, L. Tong, Sensor networks with mobile agents: optimal random access and coding. *IEEE J. Sel. Areas Commun.* (Special issue on Sens. Netw.) **22**(6), 1058–1068 (2004)
- A. Viguria, I. Maza, A. Ollero, Distributed service-based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions. *Adv. Robot.* **24**(1–2), 1–23 (2010)
- B. Yun, B. Chen, K. Lum, T. Lee, Design and implementation of a leader-follower cooperative control system for unmanned helicopters. *J. Control Theory Appl.* **8**(1), 61–68 (2010)
- S. Zelinski, T.J. Koo, S. Sastry, Hybrid system design for formations of autonomous vehicles, in *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, USA, 2003, vol. 1, pp. 1–6
- D. Zhang, G. Xie, J. Yu, L. Wang, Adaptive task assignment for multiple mobile robots via swarm intelligence approach. *Robot. Auton. Syst.* **55**(7), 572–588 (2007)

M. L. Cummings

Contents

39.1 Introduction 978
39.2 Operator Interaction in Centralized UAV Architectures 978
39.3 Operator Capacity in Centralized UAV Architectures 980
39.4 Operator Interaction in Decentralized UAV Architectures 986
39.5 Conclusion 990
References 991

Abstract

There has been significant recent research activity attempting to streamline Unmanned Aerial Vehicle (UAV) operations and reduce staffing in order to invert the current many-to-one ratio of operators to vehicles. Centralized multiple UAV architectures have been proposed where a single operator interacts with and oversees every UAV in the network. However, a centralized network requires significant operator cognitive resources. Decentralized multiple UAV networks are another, more complex possible architecture where an operator interacts with an automated mission and payload manager, which coordinates a set of tasks for a group of highly autonomous vehicles. While a single operator can maintain effective control of a relatively small network of centralized UAVs, decentralized architectures are more scalable, particularly in terms of operator workload, and more robust to single points of failure. However, in terms of operator workload, the ultimate success of either a centralized or decentralized UAV architecture is not how many vehicles are in the network per se but rather how many tasks the group of vehicles generates for the operator and how much autonomy is onboard these vehicles. Task-based control of UAV architectures with higher degrees

M.L. Cummings
Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: m.cummings@duke.edu

of autonomy (i.e., decentralized networks) can mitigate cognitive overload and reduce workload. Mutually exclusive boundaries for humans and computers in multiple UAV systems should not be the goal of designers for either centralized or decentralized architectures, but rather more effort needs to be spent in defining mutually supportive roles such that humans and computers complement one another.

39.1 Introduction

The use of unmanned aerial vehicles (UAVs), often referred to as drones, has recently revolutionized military operations worldwide and holds similar promise for commercial settings. The U.S. Air Force now has more UAVs than manned aircraft, small UAVs are now used worldwide by various first response and police units, and they can be found fighting forest fires, monitoring wildlife and possible poachers, in cargo missions and even in entertainment.

UAVs require human guidance to varying degrees and often through several operators. Most military and government UAVs require a crew of two to be fully operational. Conventional stick-and-rudder skills have been replaced by point-and-click control so that traditional pilots are no longer needed to control such systems. Onboard automation currently determines the most efficient control response, which is true in many commercial aircraft. While one operator supervises the actual flight activity of the UAV (the “pilot”), the other operator typically monitors the UAV’s sensors, such as a camera, and coordinates with the “pilot” so that he or she can maneuver the UAV for the best system response.

There has been significant recent research activity attempting to streamline UAV operations and reduce staffing in order to invert the current many-to-one ratio of operators to vehicles. This is important not just for military operations but also for future commercial operations where air traffic controllers will direct both manned and unmanned aircraft. This chapter will discuss the implications of this staffing inversion for multiple UAV control, particularly in terms of two UAV control architectures, centralized and decentralized. It should be noted that these two architectures are not mutually exclusive and that there really exists a continuum of architectures in between these two bookends. Moreover, while there are many aspects of control architectures that are critical to consider, this chapter focuses on the human implications of such control architectures.

39.2 Operator Interaction in Centralized UAV Architectures

The shift from stick-and-rudder to point-and-click control in UAVs represents a shift in the role of humans from the need for highly rehearsed skill sets to more knowledge-based reasoning inputs. For UAVs and for fly-by-wire military and commercial aircraft, pilots are less in direct manual control of systems but more involved in the higher levels of planning and decision making, particularly for

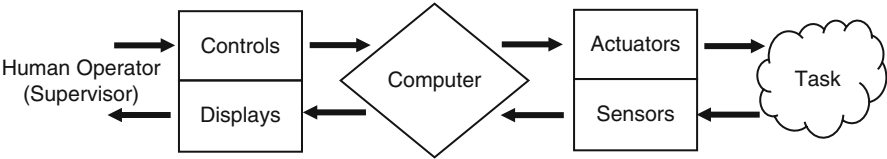


Fig. 39.1 Human supervisory control (Sheridan and Verplank 1978)

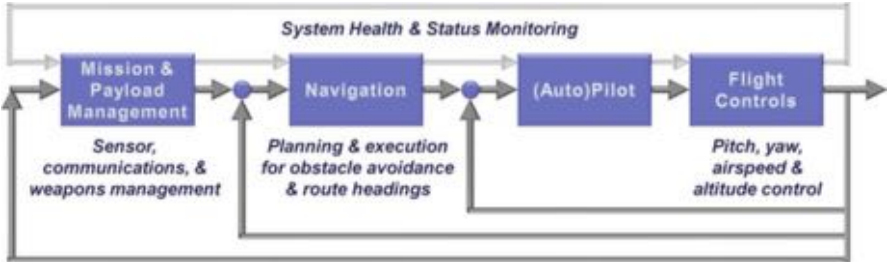


Fig. 39.2 Hierarchical control loops for a single UAV

remote operations. This shift in control from lower-level skill-based behaviors to higher-level knowledge-based behaviors is known as Human Supervisory Control (HSC). HSC is the process by which a human operator intermittently interacts with a computer, receiving feedback from and providing commands to a controlled process or task environment, which is connected to that computer (Sheridan and Verplank 1978) (Fig. 39.1).

In a centralized UAV control architecture, human supervisory control in UAV operation is hierarchical, as represented in Fig. 39.2. The innermost loop of Fig. 39.2 represents the basic guidance and motion control loop, which is the most critical loop that must obey physical laws of nature such as aerodynamic constraints for UAVs. In this loop, the autopilot optimizes local control (keeping the aircraft in stable flight), and while UAV pilots could theoretically take control in this loop, with inherent time latencies that can cause pilot-induced instabilities, this loop is generally left to the automation.

The second loop, the navigation loop, represents the actions that some agent, whether human or computer, must execute to meet mission constraints such as routes to waypoints, time on targets, and avoidance of threat areas and no-fly zones. In most current systems, humans enter GPS coordinates as waypoints, and then the system automatically flies to these waypoints. Only now are more advanced automated path planners that generate entire missions instead of serial waypoints starting to appear in operationally deployed UAV systems.

The outermost loop of Fig. 39.2 represents the highest levels of control, that of mission and payload management. In this loop, sensors must be monitored and decisions made based on the incoming information to meet overall mission requirements.

In the mission management loop, human operators provide the greatest benefit since their decisions require knowledge-based reasoning that includes judgment, experience, and abstract reasoning that in general cannot be performed by automation.

Finally, the system health and status monitoring loop on the top of Fig. 39.2 represents the continual supervision that must occur, either by a human or automation or both, to ensure all systems are operating within normal limits. This control loop is a highly intermittent loop in terms of the human, i.e., if the human is engaged in another task, with the highest priority given to the innermost loop, health and status monitoring becomes a distant, secondary task.

From the human-in-the-loop perspective, if the inner loops fail, then the higher (outer) loops will also fail. The dependency of higher loop control on the successful control of the lower loops drives human limitations in control of a single and, especially so, for multiple UAVs. If humans must interact in the guidance and motion control loop (e.g., manually fly a UAV), the cost is high because this effort requires significant cognitive resources. What little spare mental capacity is available must be divided between the navigation and mission management control loops. Violations of the priority scheme represented in Fig. 39.2 have led to numerous crashes (Williams 2004). When operators become cognitively saturated or do not correctly allocate their cognitive resources to the appropriate control loops in the correct priorities, they violate the control loops constraints, potentially causing catastrophic failure.

39.3 Operator Capacity in Centralized UAV Architectures

In centralized UAV systems supervised by a human, the primary consideration for system design is how many vehicles a single controller can effectively supervise. Since this supervisor has to interact with each vehicle individually, just how many vehicles can be effectively and safely controlled will primarily be driven by the amount of autonomy onboard the aircraft, which is subsumed across the four loops as shown in Fig. 39.2.

By increasing UAV autonomy, operator workload will theoretically be reduced as it could reduce the number of tasks for the operator, and it should reduce the level of interaction even at the highest levels of control in Fig. 39.2. For example, those UAVs that are flown in an autopilot mode relieve the operator from the manual flying tasks that require significant cognitive resources. This frees the operator to perform other critical tasks like mission planning and imagery analysis.

While there have been many studies that have attempted to experimentally derive the number of UAVs a single operator can control in a given setting (e.g., Ruff et al. 2002; Dixon et al. 2003; Cummings and Guerlain 2007), model-based approaches are generally more useful in determining not only an upper bound, but also provide insight into how much autonomy will be needed if a certain number of UAVs in a system is desired.

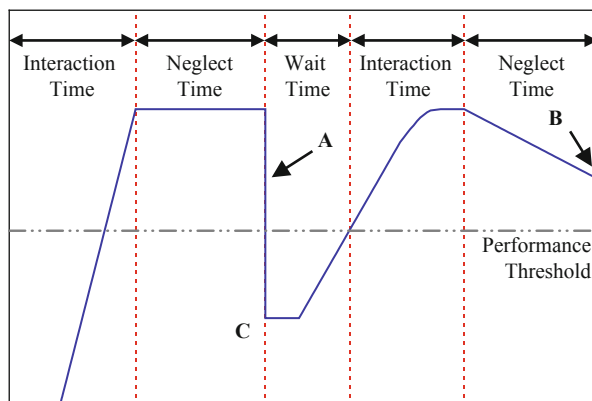


Fig. 39.3 The relationship between interaction, neglect, and wait times (Cummings and Mitchell 2008)

One such approach is the fan-out approach (Olsen and Wood 2004), which predicts the upper bound of the number of vehicles a single operator can control given the amount of autonomy in a vehicle as represented through its neglect time (NT) and the required human-computer interaction called interaction time (IT). The fan-out model was later modified to account for wait times that would inevitably be experienced by the system due to human inefficiencies. This adjustment provides a more realistic and lowered upper bound (Cummings and Mitchell 2008). For example, one UAV can operate in a period of NT, and during that period of NT, the operator can attend to other vehicles. However, if an operator fails to notice that a UAV needs assistance (such as needing a new goal once a waypoint has been achieved), or becomes engrossed in a mission plan for a UAV with a system malfunction, one or more UAVs can wait for the operator's attention, causing a delay for one or more vehicles.

Figure 39.3 represents how NT, IT, and wait times (WT) interrelate. Point A represents a discrete event that occurs after a period of neglect time, which causes the vehicle to require immediate operator assistance such as an engine loss. NTs may not be so clearly observable, as exemplified by Point B, which represents performance degradation causing vehicle performance to drop below the NT performance threshold, e.g., a slow degradation of an inertial navigation system. In both NT cases in centralized UAV architectures, once performance has dropped below an acceptable level requiring human interaction, the UAV must wait until the operator recognizes and solves the problem and so that the UAV can move to another NT state. Point C illustrates the system time delay if the problem is not addressed at the appropriate time.

Equation 39.1 represents the fan-out mathematical relationship where NT and IT are as defined above. However, it should be noted that IT should account for not just the time an operator inputs commands to a UAV, but also it should include delays where an operator has entered a command and waits for a system's response.

Wait times that add delays to the necessary ITs are accounted for in a separate term as defined by Eq. 39.2.

$$FO = \frac{NT}{IT + WT} + 1 \quad (39.1)$$

$$WT = \sum_{i=1}^X WTQ_i + \sum_{j=1}^Y WTSA_j \quad (39.2)$$

WT: Wait time

WTQ: Queuing wait time

WTSA: Wait time caused by a loss of situation awareness

X = Number of human-automation interaction queues that build

Y = Number of time periods in which a loss of SA causes wait time

In Eq. 39.2, WTQ, or wait time due to queue, results when multiple UAVs require attention, but the operator can only serially attend to them, effectively causing a queue to form for the operator's attention. For example, if an operator is controlling two UAVs on a search mission and both require the operator to insert waypoints near-simultaneously, the second UAV may have to loiter in place while the operator attends to the first. Assuming the operator can switch attention quickly once the first UAV is back to NT, the time the second UAV waits in the queue (WTQ) is effectively the IT for the first vehicle.

WTSA, or wait time due to a loss of situation awareness, is perhaps the most difficult wait time component to model because it represents how effectively an operator can manage his or her attention. Situation awareness (SA) is generally defined as having three levels, which are (1) the perception of the elements in the environment, (2) the comprehension of the current situation, and (3) the projection of future status (Endsley 1995). While SA can decrease under high workload due to competition for attentional resources (Andre and Wickens 1995), it can also decrease under low workload due to boredom and complacency (Rodgers et al. 2000). If an operator does not realize a UAV needs attention (and thus experiences a loss of SA), the time from the initial onset of the need for IT to actual operator recognition of the problem could range from seconds to minutes.

Thus, Eq. 39.2 categorizes system wait times as the summation of wait times that result from queues due to near-simultaneous arrival of events that require human intervention plus the wait times due to the operator loss of SA. Wait times increase overall IT and reduce the number of vehicles a single operator can supervise.

In terms of operator capacity, Eq. 39.2 demonstrates that as a UAV's degree of autonomy increases (expressed as an increase in NT), holding all other parameters equal means that a single operator could control more vehicles. Consequently, for a fixed NT (or degree of autonomy), operator capacity could be increased by making the ground control station interactions more streamlined (i.e., lower IT) or ensuring all the correct alarms are in place so that operators do not miss critical points of interventions (i.e., lower WTSA).

Table 39.1 Levels of autonomy

| LOA | Automation description |
|-----|--|
| I | The computer offers no assistance: human must take all decision and actions |
| II | The computer offers a complete set of decision/action alternatives |
| III | The computer offers a selection of decisions/actions |
| IV | The computer suggests a plan and executes that suggestion if the human approves (management by consent) |
| V | The computer suggests a plan and allows the human a restricted time to veto before automatic execution (management by exception) |
| VI | The human is not involved in the decision-making process; the computer decides and executes autonomously |

Table 39.2 Multiple UAV Study Comparison

| | Experiment | LOA | Max UV# |
|---|----------------------------------|--------|---------|
| 1 | Dixon et al. (2005) (baseline) | I | 1 |
| 2 | Dixon et al. (2005) (autopilot) | I | 2 |
| 3 | Dixon et al. (2005) (auto-alert) | IV | 2 |
| 4 | Ruff et al. (2002) | IV | 4 |
| 5 | Dunlap (2006) | IV | 4 |
| 6 | Cummings et al. (2008) | III–IV | 5 |
| 7 | Lewis et al. (2006) | IV–V | 8 |
| 8 | Cummings and Guerlain (2007) | IV | 12 |
| 9 | Hilburn et al. (1997) (ATC) | N/A | 11 |

A meta-analysis of several previous studies looking at various levels or degrees of automation/autonomy in centralized single operator control of multiple UAVs, particularly at the mission management level (the outermost loop in Fig. 39.2), demonstrates that as NT increases (meaning the degree of autonomy increases), the maximum number of unmanned vehicles a single operator can control increases (Cummings and Mitchell 2008) (Tables 39.1 and 39.2 and Fig. 39.4). The levels of autonomy (LOAs) in Table 39.1 are loosely modeled on Sheridan’s levels of automation framework (Sheridan and Verplank 1978). While there have been numerous other levels of automation and autonomy proposed for UAVs, as recently highlighted by a 2012 Department of Defense report (Defense Science Board 2012), such frameworks pose many problems. These LOAs used here only illustrate increasing degrees of autonomy and thus NT and are not meant to be normative.

As shown in Fig. 39.4, research has previously demonstrated that with very low levels of mission management automation, a single operator can supervise at best only two UAVs. However, given high neglect times enabled by higher degrees of autonomy (i.e., UAVs plan their own routes and obstacle avoidance, only seeking high-level mission plan approval), experimentally operators have

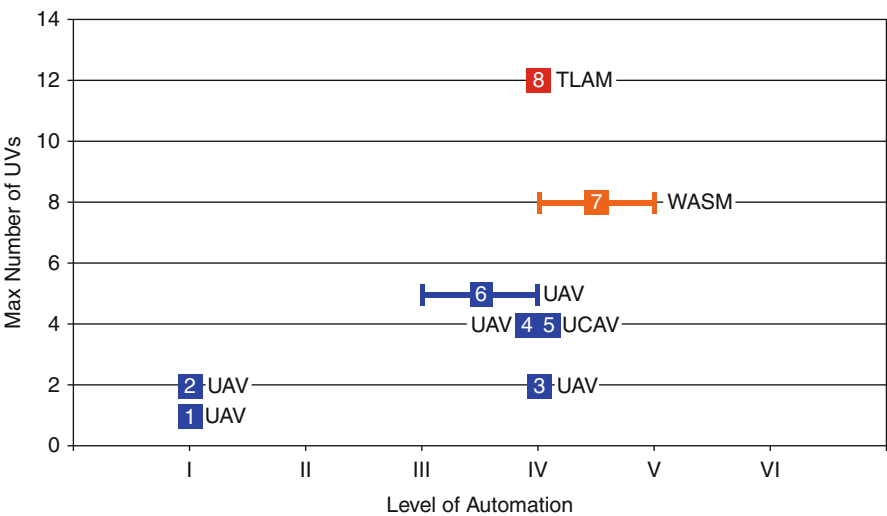


Fig. 39.4 Meta-analysis of previous experiments demonstrating increased operator capacity for increasing neglect time (as expressed by levels of automation)

successfully controlled up to 12 Tomahawk Land Attack Missiles (TLAM), which are highly automated missiles that navigate on their own. These vehicles are effectively one-way UAVs, controlled in much the same way as UAVs through GPS commands. WASMs, or wide area search munitions, are also similar weapons that are launched from another aircraft and then fly themselves until they find their assigned target. Curiously the number of WASMs and TLAMs, which are highly automated and operate in centralized control systems, a single operator can simultaneously control (8–12) is very similar to the number of airplanes a single air traffic controller can handle (~11, Hilburn et al. 1997). Arguably, manned aircraft have similar degrees of autonomy since the pilot onboard is expected to obey all the high-level commands of the controller.

One of the limitations common across the studies in Table 39.2 is the lack of measurable *system-level* performance metrics. In general for the studies in Table 39.2, the performance of the operators was deemed acceptable as a function of expert observation, which is a valid method for performance assessment (Endsley and Garland 2000) but is not generalizable across domains and only useful as a descriptive and not predictive metric. Thus, a system-level performance metric should capture both aspects of human and automation performance, which indicates an objective level of goodness and/or satisficing (Simon et al. 1986) (i.e., a “good enough” solution as opposed to optimal). Such system-level metrics are often referenced as key performance parameters (KPPs) (Joint Chiefs of Staff 2007).

Towards this end of developing more comprehensive KPPs for multiple UAV systems, a recent study demonstrated that the number of UAVs that a single operator can control in a centralized architecture is not just a function of the level of decision

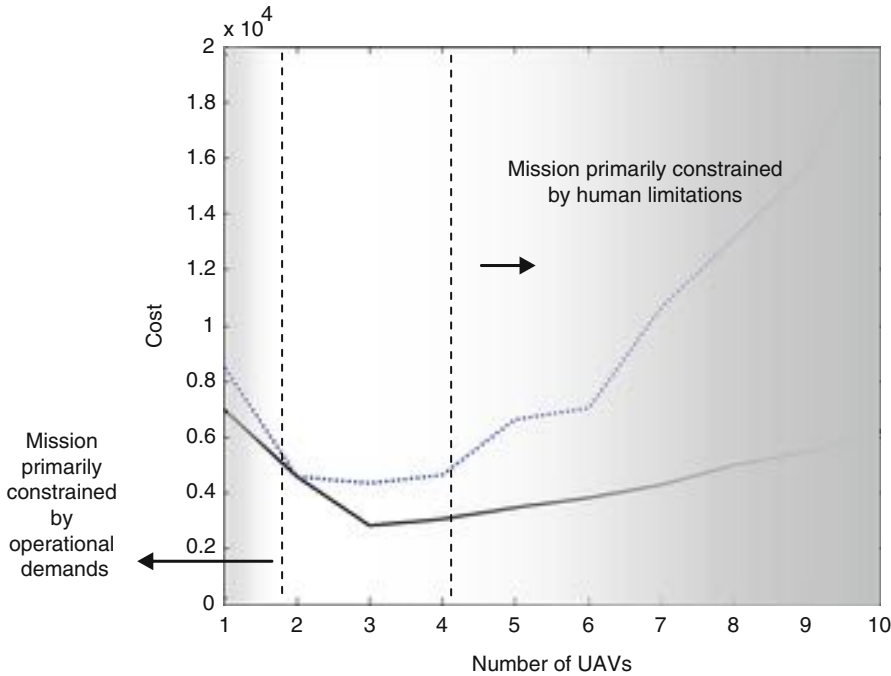


Fig. 39.5 Operator capacity as a function of mission constraints

support automation but is inextricably tied to both mission complexity and overall system performance (Cummings et al. 2007). Using human experimentation in a multiple UAV simulation test bed and a simulated annealing (SA) technique for heuristic-based optimization, operator performance was predicted to be significantly degraded beyond approximately five UAVs with approximately levels 3–4 of autonomy as defined in Table 39.1. The optimal range was predicted to be between 2 and 4 vehicles (Fig. 39.5). Interestingly, in a different single operator-multiple UAV study with an entirely different test bed but similar levels of autonomy and centralized architecture, the optimal number was experimentally determined to be ~ 4 UAVs (Cummings and Mitchell 2008).

The KPP in Fig. 39.5 is cost, which takes into account not just operational costs such as fuel but also the cost of missed targets and cost in terms of mission delays introduced by inefficient human interactions. The solid curve in Fig. 39.5 represents a theoretically perfect human operator, and the dotted line represents more realistic human performance that accounts for delays due to inefficient decision making, communication problems, cognitive load, etc. Thus, the performance of the system (the automation *and* the operator) can vary both as a function of the operator but also can vary due to the operational constraints such as number of targets and operational costs. This variation is why it is important to explicitly link system performance to operator capacity.

39.4 Operator Interaction in Decentralized UAV Architectures

While Fig. 39.2 demonstrates supervisory control at the single vehicle level, which for centralized multiple UAV control is simply replicated for each vehicle under control, Fig. 39.6 represents a notional system architecture that will be required for single operator control of multiple decentralized UAVs. In order to achieve this futuristic system, operators will need to interact with an overall automated mission and payload manager, which coordinates a set of tasks for a group of vehicles, instead of individually tasking each vehicle. This effectively represents a decentralized architecture, where operators convey high-level goals to an automated mission manager (such as requesting that an area be searched), which then allows the UAVs to coordinate across the group to determine how to assign particular tasks, which may be dynamic. In a decentralized architecture, navigation and motion control tasks are necessarily subsumed by automation.

The decentralized architecture provides a substantial benefit in that the operator and his or her ground control station do not become a single point of failure, i.e., if the operator has intermittent or loss of communications with the vehicles, the system can still function. For example, because the network of vehicles communicates with one another, if one vehicle breaks down, another can take its place. Another advantage is that the system is robust to lapses in operator situation awareness and delays since vehicles do not necessarily have to wait for commands. However, emergent UAV behavior in such systems can be complex and confusing for an operator, and if the system operates in a suboptimal fashion, it could be difficult for operators to correct problems unless they have the ability to understand and then execute the necessary commands to correct the system.

For operators supervising a decentralized system, the fan-out approach as depicted in Eq. 39.1 cannot be used to estimate operator capacity, since for centralized systems, the assumption is that the vehicles have their own independent NTs and ITs, which drives the overall number of vehicles that can be controlled. Since operators only provide high-level goals at the mission and payload management level, they do not have an IT for each vehicle but rather an IT for high-level interaction with the team. Similarly for NT, there is not distinct per vehicle NT, since they work together.

In control of a decentralized UAV network, the question of operator capacity is driven by how many *tasks* an operator can handle instead of how many *vehicles*. Under task-based, decentralized control, a human operator provides high-level control by approving which tasks should be completed by the team of vehicles without directly tasking a particular vehicle. Then the decentralized network of vehicles chooses how to allocate the approved tasks among themselves and can make tactical-level changes on their own, such as switching tasks.

In controlling a network of collaborative, decentralized UAVs, the operator could control, for example, 2, 20, 200, or even 2,000 UAVs, as long as the tasks generated by the group of UAVs were manageable by a single operator. Determining the task load manageable by a single operator can roughly be thought of as the number of

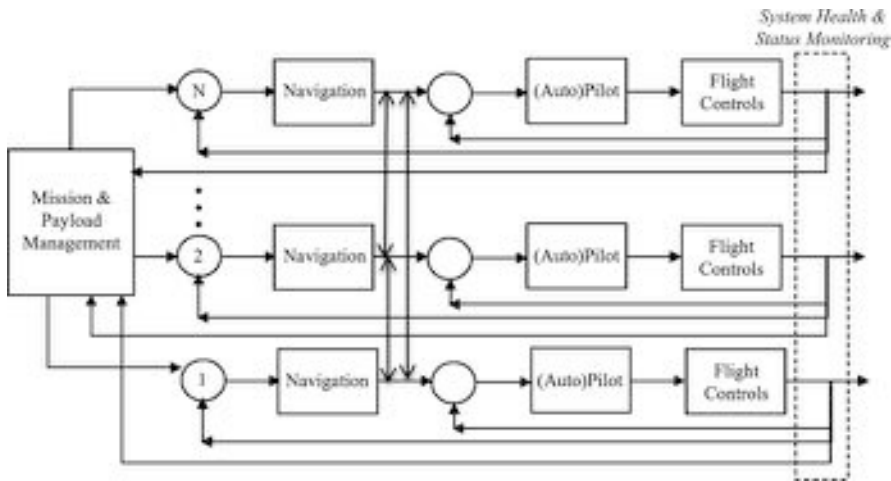


Fig. 39.6 Decentralized control for multiple UAVs

tasks that can be successfully accomplished over the course of the mission. While this number will, of course, vary widely across different missions and with different mixes of vehicles and people, there is one proxy metric that can be used across any decentralized UAV system for workload comparison, which is the concept of utilization.

Utilization refers to the “percent busy time” of an operator, i.e., given a time period, the percentage of time that person is busy. In supervisory control settings, this is generally meant as the time an operator is directed by external events to complete a task (e.g., replanning the path of a UAV because of an emergent target) or attending to internal tasks like responding to text messages. What is not included in this measurement is the time spent monitoring the system, i.e., just watching the displays and/or waiting for something to happen. The concept of utilization as a mental workload measure has been used in numerous studies examining supervisory controller performance (Schmidt 1978; Rouse 1983; Cummings and Guerlain 2007; Donmez et al. 2010). These studies generally support that when tasked beyond 70 % utilization, operators’ performances decline.

In terms of the previously discussed IT and NT terms, utilization can generally be thought of as $IT/(NT + IT)$ on the aggregate task level. It can be used to describe an operator’s response to task load in centralized UAV architectures as well as decentralized, but it is especially useful for decentralized system analysis given the operator’s interaction at the meta-level instead of the individual vehicle level, which reflects the architecture of Fig. 39.6.

In order to determine whether decentralized systems provide any utility in terms of reduced workload and improved performance as compared to centralized UAV architectures, a set of studies was compared that span increasing degrees of

autonomy and increasing task loads, summarized in Table 39.1. In the first experiment at the lowest degree of autonomy (Nehme et al. 2008), operators controlled 2–8 mostly centralized unmanned ground vehicles (UGVs), which resulted in four different experiment levels. For this experiment set, the individual vehicles relied on the human for goal setting but had some ability to share local navigation information with one another.

In the second experiment with three experimental levels (Nehme 2009), a single operator controlled various mixes of five multiple unmanned aerial and underwater vehicles, with slightly more autonomy in the sense that vehicles would not only path plan themselves, but if the operator did not assign an ultimate goal within a prespecified time, the vehicles would assign themselves to the nearest target. However, the operator could override any individual vehicle and redirect not only its path but its ultimate goal as well.

The last experiment with four increasing task load levels represented the highest degree of decentralization, in that operators could only specify a task list to a group of five unmanned aerial and unmanned surface ships. The vehicles negotiated among themselves through a consensus-based bundled algorithm which vehicle would be assigned to which task, and each vehicle determined its own route. The operator could only insert and reprioritize tasks but never direct an individual vehicle. Two different studies were included that used this test bed which focused on medium (Cummings et al. 2010) and high (Clare and Cummings 2011) levels of task loading.

In order to directly compare these different studies, the average tasks per minute were determined, which shows how many tasks in each experiment the operator was expected to complete over an average 1-min time interval. Performance scoring was aggregated into low, medium, and high categories since the performance metrics used in each of the sets of experiments could not be directly compared. Low means that for the specific study, that condition resulted in the worse performance and, respectively, for the high performance ranking. Lastly, the average utilization, which is the percentage of time the operator was busy performing tasks required by the system, was also listed.

Figure 39.7 illustrates that for each of the three sets of experiments, utilization increased with an increasing task load, which is expected and also an internal validity check. In addition, Fig. 39.7 also illustrates that as the degree of decentralization increases, (i.e., more autonomy across a network of vehicles and less direct control by a human operator), utilization decreases. Interestingly all three experiments had an experimental level of ~8 tasks/min and the most decentralized architecture allowed the average operator to work less by 11 % as compared to the more centralized architecture (and 3 % less than the somewhat centralized architecture (Nehme 2009)). And in no case did the centralized architectures produce lower utilizations for similar task loads.

In terms of performance, each of the observed data points in Table 39.3 are color coded in Fig. 39.7 to reflect the relative performance score, with red indicating the worst performance; yellow demonstrates moderate performance, and green represents the best performance. Recall that each of these scores is a relative ranking so caution is advised in interpretation. In general the lower utilizations

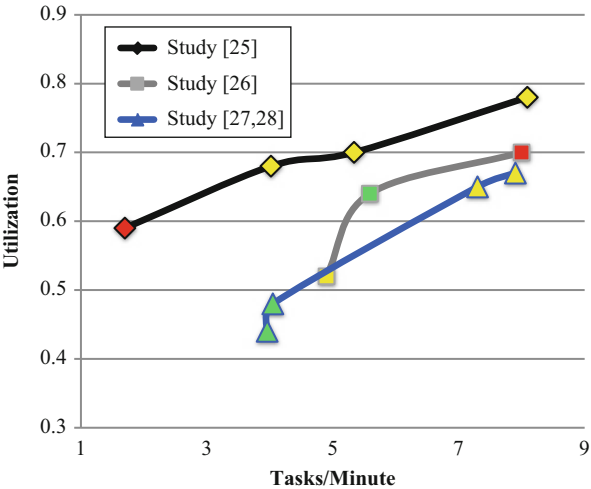


Fig. 39.7 Utilization for the average tasks/minute for the experiments in Table 39.3

Table 39.3 Task load, workload, and performance for three multiple UAV architectures with increasing autonomy

| Experiment | Architecture | Primary task load | Average tasks/min | Performance | Average utilization |
|--|--|------------------------|-------------------|-------------|---------------------|
| #1: Unmanned ground vehicles (Nehme et al. 2008) | Mostly centralized, with some navigation sharing | 2 vehicles | 1.70 | Low | 0.59 |
| | | 4 vehicles | 4.02 | Medium | 0.68 |
| | | 6 vehicles | 5.34 | Medium | 0.70 |
| | | 8 vehicles | 8.09 | Medium | 0.78 |
| #2: Unmanned aerial and underwater vehicles (Nehme 2009) | Somewhat centralized with some autonomous path planning and goal selection, 5 vehicles | Same-type vehicles | 4.90 | Medium | 0.52 |
| | | 2 different types | 5.59 | High | 0.64 |
| | | All different vehicles | 8.00 | Low | 0.70 |
| #3–4: Unmanned ground and aerial vehicles, medium (Cummings et al. 2010) and high (Clare and Cummings 2011) workload | Decentralized, 5 vehicles | 23.3 tasks | 3.96 | High | 0.44 |
| | | 30 tasks | 4.05 | High | 0.48 |
| | | 43.3 tasks | 7.30 | Medium | 0.65 |
| | | 50 tasks | 7.90 | Medium | 0.67 |

produced the best performances, with the caveat that little work has been done in terms of the possible negative impact of low task load on performance (in one exception, see Cummings et al. 2013). For all studies, performance suffered when

the 70 % utilization threshold was exceeded, with only the somewhat centralized study participants exhibited markedly worse performance.

For the somewhat decentralized (Nehme 2009) and more decentralized experiments (Cummings et al. 2010; Clare and Cummings 2011), there appears to be a nonlinear relationship in the 6 tasks/min and below regions, which is not evident in the centralized experiment (Nehme et al. 2008). More work is needed to determine why such nonlinear relationships exist, the nature of the critical point for the sharp rise, and how a system could be better designed to reduce the sharp increases in workload.

39.5 Conclusion

Taken together, the results from both the centralized and decentralized sets of experiments demonstrate that decentralized, task-based control of UAV architectures with higher degrees of autonomy can mitigate cognitive overload and reduce workload. While a single operator can maintain effective control of a relatively small network of centralized UAVs, decentralized architectures are more scalable, since adding additional agents also adds computational capability (assuming the tasks generated by the system do not linearly increase). Moreover, the decentralized UAV framework is robust to a single point of failure, since no single agent is globally planning for the fleet.

In terms of workload for a supervising operator, the ultimate success of either a centralized or decentralized UAV architecture is not how many vehicles are in the network per se but rather how many tasks the group of vehicles generates for the operator and how much autonomy is onboard these vehicles so that neglect time can be increased. And while this increasing autonomy in a decentralized network of UAVs can mean reduced workload for the operator, it also adds significant more complexity to the system, which not only means increased developmental costs than a centralized network but also one that is much harder to certify as safe.

Another caveat to the use of increased autonomy to mitigate workload across a UAV network is that such increased autonomy and increased neglect times can exacerbate a loss of operator situation awareness, as well as promote complacency and skill degradation (Parasuraman et al. 2000). Management-by-exception architectures, which occur when automation takes action based on some set of predetermined criteria and only gives operators a chance to veto the automation's decision, have been shown to improve operator performance (Cummings and Mitchell 2006). However, in such control schemes, operators are also more likely to exhibit automation bias, a decision bias that occurs when operators become over-reliant on the automation and do not check to ensure automated recommendations are correct (Mosier and Skitka 1996). Automation bias is a significant concern for command and control systems, so it will be critical to ensure that when higher levels of automation are used, especially at the management-by-exception level, this effect is minimized.

Lastly, while it is critical to consider the mental workload of a supervisor of multiple UAVs, the ability of the human to add value to the performance of a team of UAVs cannot be overlooked. In one study that examined whether human supervisors added value in a search and track task for a decentralized, highly autonomous network of UAVs, results in a controlled study show that a 30–50% increase in overall system performance, particularly in the search task, could be achieved by letting humans coach the automation (Cummings et al. 2012). Thus, instead of attempting to dictate mutually exclusive boundaries for human and computers in multiple UAV systems, either centralized or decentralized, more effort needs to be spent in trying to define mutually supportive roles such that humans and computers complement one another.

References

- A. Andre, C. Wickens, When users want what's not best for them. *Ergon. Des.* **3**, 10–14 (1995)
- A.S. Clare, M.L. Cummings, Task-based interfaces for decentralized multiple unmanned vehicle control, in *AUVSI Unmanned Systems North America*, Washington, D.C., 2011
- M.L. Cummings, S. Guerlain, Developing operator capacity estimates for supervisory control of autonomous vehicles. *Hum. Factors* **49**(1), 1–15 (2007)
- M.L. Cummings, P.J. Mitchell, Automated scheduling decision support for supervisory control of multiple UAVs. *AIAA J. Aerosp. Comput. Inf. Commun.* **3**(6), 294–308 (2006)
- M.L. Cummings, P.J. Mitchell, Predicting controller capacity in supervisory control of multiple UAVs. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **38**(2), 451–460 (2008)
- M.L. Cummings, C.E. Nehme, J. Crandall, Predicting operator capacity for supervisory control of multiple UAVs, in *Innovations in Intelligent Machines*, vol. 70, ed. by J.S. Chahl, L.C. Jain, A. Mizutani, M. Sato-Ilic (Springer, Berlin/New York, 2007)
- M.L. Cummings, A. Clare, C. Hart, The role of human-automation consensus in multiple unmanned vehicle scheduling. *Hum. Factors* **52**(1), 17–27 (2010)
- M.L. Cummings, J. How, A. Whitten, O. Toupet, The impact of human-automation collaboration in decentralized multiple unmanned vehicle control. *Proc. IEEE* **100**(3), 660–671 (2012)
- M.L. Cummings, C. Mastracchio, K.M. Thornburg, A. Mkrtchyan, Boredom and distraction in multiple unmanned vehicle supervisory control. *Interact. Comput.* **25**(1), 34–47 (2013)
- Defense Science Board, The role of autonomy in DoD systems. Department of Defense, 2012
- S.R. Dixon, C.D. Wickens, D. Chang, Comparing quantitative model predictions to experimental data in multiple-UAV flight control, in *Human Factors and Ergonomics Society 47th Annual Meeting*, Denver, 2003
- S. Dixon, C. Wickens, D. Chang, Mission control of multiple unmanned aerial vehicles: a workload analysis. *Hum. Factors* **47**, 479–487 (2005)
- B. Donmez, C. Nehme, M.L. Cummings, Modeling workload impact in multiple unmanned vehicle supervisory control. *IEEE Syst. Man Cybern. Part A Syst. Hum.* **99**, 1–11 (2010)
- R.D. Dunlap, The evolution of a distributed command and control architecture for semi-autonomous air vehicle operations, in *Moving Autonomy Forward Conference*, Grantham (Muretex, 2006)
- M.R. Endsley, Toward a theory of situation awareness in dynamic systems. *Hum. Factors* **37**(1), 32–64 (1995)
- M.R. Endsley, D.J. Garland, *Situation Awareness Analysis and Measurement* (Lawrence Erlbaum, Mahwah, 2000)
- B. Hilburn, P.G. Jorna, E.A. Byrne, R. Parasuraman, The effect of adaptive air traffic control (ATC) decision aiding on controller mental workload, in *Human-Automation Interaction:*

- Research and Practice*, ed. by M. Mouloua, J.M. Koonce (Lawrence Erlbaum, Mahwah, 1997), pp. 84–91
- Joint Chiefs of Staff, Chairman of the Joint Chiefs of Staff instruction 6212.01D. DoD, 2007
- M. Lewis, J. Polvichai, K. Sycara, P. Scerri, Scaling-up human control for large UAV teams, in *Human Factors of Remotely Operated Vehicles*, ed. by N. Cooke, H. Pringle, H. Pedersen, O. Connor (Elsevier, New York, 2006), pp. 237–250
- K.L. Mosier, L.J. Skitka, Human decision makers and automated decision aids: made for each other? in *Automation and Human Performance: Theory and Applications, Human Factors in Transportation*, ed. by R. Parasuraman, M. Mouloua (Lawrence Erlbaum, Mahwah, 1996), pp. 201–220
- C.E. Nehme, Modeling human supervisory control in heterogeneous unmanned vehicle systems. Doctor of philosophy, Massachusetts Institute of Technology, 2009
- C.E. Nehme, J. Crandall, M.L. Cummings, Using discrete-event simulation to model situational awareness of unmanned-vehicle operators, in *2008 Capstone Conference*, Norfolk, 2008
- D.R. Olsen, S.B. Wood, Fan-out: measuring human control of multiple robots, in *SIGCHI conference on Human factors in Computing Systems*, Vienna, 2004
- R. Parasuraman, T.B. Sheridan, C.D. Wickens, A model for types and levels of human interaction with automation. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **30**(3), 286–297 (2000)
- M.D. Rodgers, R.H. Mogford, B. Strauch, Post hoc assessment of situation awareness in air traffic control incidents and major aircraft accidents, in *Situation Awareness Analysis and Measurement*, ed. by M. Endsley, D.J. Garland (Lawrence Erlbaum, Mahwah, 2000), pp. 73–112
- W.B. Rouse, *Systems Engineering Models of Human-Machine Interaction* (North Holland, New York, 1983)
- H. Ruff, S. Narayanan, M.H. Draper, Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. *Presence* **11**(4), 335–351 (2002)
- D.K. Schmidt, A queuing analysis of the air traffic controller's workload. *IEEE Trans. Syst. Man Cybern.* **8**(6), 492–498 (1978)
- T.B. Sheridan, W. Verplank, Human and computer control of undersea teleoperators. Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT, Cambridge, 1978
- H.A. Simon, R. Hogarth, C.R. Piott, H. Raiffa, K.A. Schelling, R. Thaler, A. Tversky, S. Winter, Decision making and problem solving, in *Research Briefings 1986: Report of the Research Briefing Panel on Decision Making and Problem Solving* (National Academy Press, Washington D.C., 1986)
- K.W. Williams, A summary of unmanned aircraft accident/incident data: human factors implications. Federal Aviation Administration, Civil Aerospace Medical Institute, Oklahoma City, 2004

Section IX

UAV Health Management Issues

Kai Goebel and Michael J. Roemer

Kimon P. Valavanis and George J. Vachtsevanos

Reliability, availability, maintainability, and safety of UAVs, and more generally of other unmanned systems, are a critical concern for the OEM and user communities. The development and application of emerging Prognostics and Health Management and Condition-Based Maintenance (CBM) technologies to all UAV classes is recognized as a key enabler for their effective utility in multiple application domains.

UAV Health Management addresses fundamental technologies and their application to UAVs that relate to monitoring and sensing strategies for health management, data acquisition and processing/analysis onboard and off-board the vehicle, fault diagnosis and failure prognosis algorithms, as well as fault-tolerant control routines that enhance the UAV's useful life and mitigate potential catastrophic events. CBM methods and tools are also covered that take advantage of the prevailing paradigm shift from scheduled or breakdown maintenance towards maintenance practices that are based on the current condition of these assets.

► **Integrated Vehicle Health and Fault Contingency Management for UAVs** by Roemer and Tang presents various concepts for integrating real-time vehicle health assessment and fault contingency management technologies for UAVs. The presented integrated vehicle health management (IVHM) and automated contingency management (ACM) system architecture are shown to support real-time, onboard health state assessment and fault management so that UAVs can enjoy greater autonomy and survivability during anomalous operating conditions. Selected

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

real-time system identification and automated health assessment algorithms are presented that can readily identify the dynamics and performance limitations of degraded UAV systems. Additionally, a high-level mission adaptation approach is presented to estimate the safe flight operating envelope after the occurrence of faults. Reconfigurable fault-tolerant control techniques that directly utilize the identified UAV subsystem dynamic models have been developed and tested in simulation. Finally, proof-of-concept demonstrations are presented using NASA engine and aircraft dynamic models with simulated engine and actuator faults.

► **Automated Failure Effect Analysis for PHM of UAV** by Snooke describes how model-based simulation can be employed to automatically generate the system-level effects for comprehensive sets of component failures on systems within the aircraft. The results of the simulation can be used in several ways. They can be used to produce a system-level failure modes and effects analysis (FMEA) for aircraft systems. They can be used to identify the sensors necessary to discriminate remotely between different failures on the aircraft. Once a set of sensors have been chosen for placement on the vehicle, the simulation results can also be used to generate diagnostic and prognostic software for deployment on the vehicle. Using automated FMEA safety analysis software is more efficient than doing the same work without software and also provides a guaranteed level of performance. Using the results of this analysis can provide sensor selection and diagnostic capability while retaining some of the benefits of rule-based diagnostic systems. Alternative model-based techniques have been widely used to create diagnostic systems in a variety of domains, and these approaches are compared with the diagnostic capability provided by a failure effect-oriented technique from the perspective of the UAV application.

► **Prognostics Applied to Electric Propulsion UAV** by Goebel and Saha explores the technical underpinnings of how to perform prognostics and shows an implementation on the propulsion of an electric UAV. An accurate run-time battery life prediction algorithm is of critical importance to ensure the safe operation of the vehicle if one wants to maximize in-air time. Current reliability-based techniques turn out to be insufficient to manage the use of such batteries where loads vary frequently in uncertain environments. A particle filter is shown as the method of choice in performing state assessment and predicting future degradation. The method is then applied to the batteries that provide power to the propeller motors.

► **Actuator Fault Detection in UAVs** by Ducard is dedicated to actuator fault detection systems for UAVs, with two main requirements: real-time capability and modularity. After defining the terminology employed in this field, it first reviews some commonly used techniques in FDI systems, followed by presenting briefly the mathematical model of a UAV that serves as a basis for the design of two actuator FDI systems. The first method presents and illustrates the multiple-model approach, whereas the second method presents an FDI system, which is based on a single model. Both methods are enhanced by a mechanism that actively tests actuators in order to efficiently detect and isolate actuator faults and failures. Advantages and drawbacks of each method are stated issues of robustness and are discussed against model uncertainties and external perturbation. In addition, aspects of computational

load are addressed. The FDI systems applied to a realistic model of an unmanned aircraft and the performance of the methods are shown in simulation.

► **Experimental Validation of Fault Detection and Diagnosis for Unmanned Aerial Vehicles** by Chamseddine, Hadi Amoozgar, and Zhang investigates the problems of fault detection, diagnosis, and fault-tolerant control for UAVs. It first presents a detailed overview of existing experimental research focusing on fixed-wing as well as rotorcraft UAVs including single-rotor and multi-rotor helicopters. It then discusses three Kalman filters employed for actuator fault detection and diagnosis, namely, the unscented Kalman filter, the two-stage Kalman filter, and the adaptive two-stage Kalman filter. The three filters are experimentally applied to a quadrotor helicopter UAV test bed, and results are shown, compared, and discussed.

► **Fault Detection and Diagnosis for NASA GTM UAV with Dual Unscented Kalman Filter** by Zhang investigates a simultaneous state and parameter estimation-based fault detection and diagnosis (FDD) scheme applied to a realistic nonlinear six degree-of-freedom fixed-wing unmanned aerial vehicle (UAV) model, the NASA GTM (generic transport model). By introducing partial loss faults in actuators into the NASA GTM, a Dual Unscented Kalman Filter (DUKF) algorithm is applied for the purpose of online estimation of both flight states and fault parameters. A Bayesian rule is then used for detection and isolation decision-making. The developed FDD scheme is implemented in both the nonlinear GTM and the linear parameter-varying (LPV) representation of the nonlinear GTM. Simulation results show satisfactory results for detecting and diagnosing the control effectors faults.

► **Fault Diagnosis of Skew-Configured Inertial Sensor System for Unmanned Aerial Vehicles** by Yoon, Kim, Bae, Kim, and Kim presents fault detection and isolation scheme to handle three successive faults in the skew-configured inertial sensors of an unmanned aerial vehicle. The skew-configured inertial sensors are composed of the primary inertial measurement unit and the redundant secondary inertial measurement unit. Since small unmanned aerial vehicles are restricted by cost and payload space, the secondary small and low-cost inertial measurement unit is installed with a skewed angle in addition to the primary inertial measurement unit. In the hybrid fault detection and isolation scheme, a parity space method and an in-lane monitoring method are combined to increase system tolerance to the occurrence of multiple successive faults during flight. The first and second faults are detected and isolated by the parity space method. The third fault is detected by the parity space method and isolated by the in-lane monitoring method based on the discrete wavelet transform. Hardware-in-the loop tests and flight experiments with a fixed-wing unmanned aerial vehicle are performed to verify the performance of the proposed fault diagnosis scheme.

Michael J. Roemer and Liang Tang

Contents

| | | |
|--------|--|------|
| 41.1 | Introduction | 1000 |
| 41.2 | Integrated IVHM and ACM Overview | 1001 |
| 41.3 | Real-Time System Health Identification and System Performance Estimation | 1002 |
| 41.3.1 | Real-Time Strong Tracking System ID Algorithm | 1003 |
| 41.3.2 | Self-Tuning Kalman Filter (STKF) | 1004 |
| 41.3.3 | Engine Health Management | 1005 |
| 41.3.4 | Flight Control Actuator Health Management | 1007 |
| 41.3.5 | Dynamic Flight Envelope Assessment | 1010 |
| 41.4 | ACM System Optimization | 1012 |
| 41.4.1 | Model Predictive Control | 1014 |
| 41.5 | Flight Envelope Assessment and Fault-Tolerant Control Simulation Results | 1015 |
| 41.5.1 | Dynamic Flight Envelope Assessment | 1015 |
| 41.5.2 | Detection of Control Effectiveness Loss | 1016 |
| 41.5.3 | Model Predictive Control for Stuck Elevator | 1018 |
| 41.5.4 | Prognostics-Enhanced ACM System | 1019 |
| 41.6 | Conclusions and Future Work | 1023 |
| | References | 1024 |

Abstract

This chapter presents various concepts for integrating real-time vehicle health assessment and fault contingency management technologies for unmanned air vehicles. The presented integrated vehicle health management (IVHM) and automated contingency management (ACM) system architecture is shown to support real-time, onboard health state assessment and fault management so that UAVs can enjoy greater autonomy and survivability during anomalous operating conditions. Selected real-time system identification and automated

M.J. Roemer (✉) • L. Tang
Impact Technologies, LLC, Rochester, NY, USA
e-mail: mike.roemer@impact-tek.com

health assessment algorithms are presented that can readily identify the dynamics and performance limitations of degraded UAV systems. Additionally, a high-level mission adaptation approach is presented to estimate the safe flight operating envelope after the occurrence of faults. Reconfigurable fault-tolerant control techniques that directly utilize the identified UAV subsystem dynamic models have been developed and tested in simulation. Finally, proof-of-concept demonstrations are presented using NASA engine and aircraft dynamic models with simulated engine and actuator faults. Simulation results and remarks on future work are also presented.

41.1 Introduction

At the forefront of technology in today's military, unmanned systems are highly desired by combatant commanders (COCOMs) for their versatility and persistence. Through their expanding roles in areas such as surveillance; signals intelligence (SIGINT); precision target designation; mine detection; and chemical, biological, radiological, and nuclear (CBRN) reconnaissance, unmanned systems have made key contributions to the Global War on Terror. Unmanned system technology has gone from a prolonged period of limited acceptance and utilization spanning a couple of decades starting with the early drones, to proving and expanding their value in Desert Storm and Kosovo, to a multifaceted, highly desired resource with the recent conflicts in Iraq and Afghanistan. As of October 2008, coalition unmanned aircraft systems (UAS) (exclusive of hand-launched systems) have flown almost 500,000 flight hours in support of Operations Enduring Freedom and Iraqi Freedom; unmanned ground vehicles (UGVs) have conducted over 30,000 missions, detecting and/or neutralizing over 15,000 improvised explosive devices (IEDs); and unmanned maritime systems (UMSs) have provided security to ports [Office of the Secretary of Defense Unmanned Systems Roadmap (2009–2034)].

UAVs and their supporting systems and personnel are seeing a dramatic increase in the operational tempo they support compared to what was originally planned or designed. In addition, modern unmanned technology is still very young, measuring the maturity of any particular platform in years and hundreds of thousands of flight hours for the most mature and heavily utilized, rather than decades and tens of millions of flight hours for many manned platforms. Add to this the fact that many platforms in high demand entered low-rate production while still under advanced concept technology demonstration (ACTD) programs and had not been fully vetted through the traditional process of going to full production. With the military under continued strain to do "more with less" (people, money, etc.), one begins to understand why early data from a 2003 OSD UAV reliability study reports that the proportions of human error versus mechanical, system, and environmentally induced mishaps are nearly reversed between UAVs and the aggregate of manned aircraft, i.e., human error is the primary cause of roughly 85 % of manned mishaps, but only 17 % of unmanned ones. The report also states the cumulative mishap rate (i.e., class A accidents per 100,000 h of flight) of 32 for predator, 334 for

pioneer, and 55 for hunter (16 since the major reliability improvements in 1996). In comparison to manned aviation mishap rates, general aviation aircraft suffer about 1 mishap per 100,000h, regional/commuter airliners about a 10th that rate, and larger airliners about a 100th that rate [Unmanned Aerial Vehicle Reliability Study, Feb, 2003, Office of the Secretary of Defense].

Clearly, there is a significant opportunity for both integrated vehicle health management (IVHM) and automated contingency management (ACM) technologies to be deployed, which will help UAVs reach the reliability and safety statistics attained by manned systems. The combination of these features will also help provide the opportunity to reconfigure the appropriate UAV control systems in the event of failures with the aim of increasing the survivability of the aircraft. Enhanced supervisory control methodologies and associated algorithms have been developed for optimizing the operations and performance of such systems due to changing environmental conditions and system degradation (Vachtsevanos et al. 2005, 2006; Bodson 1995; Byington et al. 2004; Volponi and Wood 2005; Litt et al. 2004; Tang et al. 2005). However, in order to provide a truly comprehensive level of integrated fault isolation and accommodation for potentially hundreds of fault scenarios that could be encountered in flight requires a dynamic, model-based fault contingency management concept that can make decisions “on the fly,” under fault conditions that were never expected or accounted for previously. Hence, a dynamically reconfigurable control architecture that can support on-line fault contingency management based on faults being detected in real time is the key enabling issue to be addressed.

This chapter presents the development of a hierarchical architecture and enabling techniques for a real-time aircraft fault contingency management system with the following features:

1. A hierarchical architecture that integrates real-time system health identification, control reconfiguration, and high-level reasoning.
2. The dynamics and performance limitation of the degraded/damaged system are estimated on-line in real time.
3. Adaptive reconfigurable control will be utilized to stabilize and recover the system using the system dynamic model identified on-line in real time.

NASA’s Modular Aero-Propulsion System Simulation (MAPSS) engine model and Generic Transport Model (GTM) have been utilized in proof-of-concept simulation studies, and preliminary results are presented.

41.2 Integrated IVHM and ACM Overview

As shown in Fig. 41.1, the core functionality of the integrated vehicle health management (IVHM) and automated contingency management (ACM) system is to use the real-time health information from detected/isolated faults to evaluate a system’s capability for performing its future operations. The ACM system will perform the contingency analysis including the assessment of controller capabilities and core operational capabilities based on reduced system functionality.

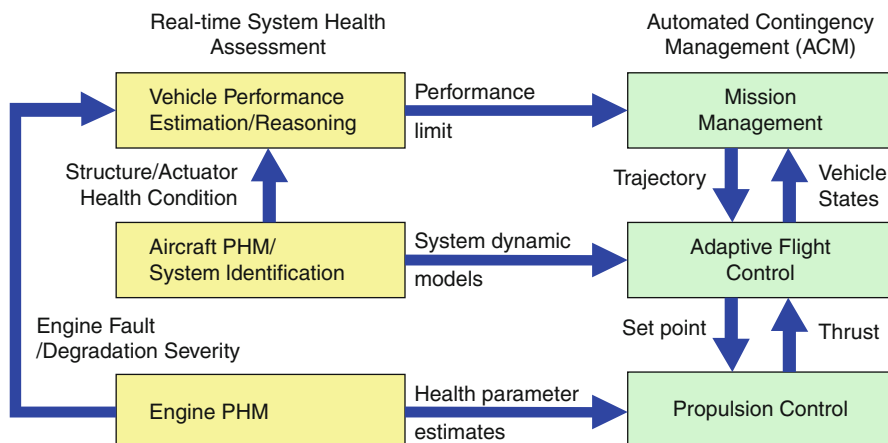


Fig. 41.1 A conceptual system overview

The analysis results will form the basis for the new constraints of the system's operation. If the current mission operation plan is incapable of being satisfied, then operation replanning and optimization becomes necessary under the new constraints in order to respond both intelligently and efficiently to the changes of system's health status or environment.

Within this architecture, UAV subsystem faults are detected by prognosis and health management (PHM) modules in real time. At the core of the real-time system health assessment modules is a real-time on-line system identification technique that can rapidly track time-varying parameters. The fault accommodation functionalities are provided in three levels. At the lowest (subsystem) level, fault-tolerant control strategies are utilized to accommodate the fault in the subsystem. A propulsion example is utilized in Fig. 41.1. However, it can also be a fault-tolerant controller for an EMA flight actuator at the subsystem level. At the flight control level, an adaptive control architecture is utilized to compute a time-varying model of the plant dynamics. Using the identified dynamics, the controller calculates the commands required to achieve the desired system responses as specified by a set of flying-qualities models. At the mission level, results from the system health assessment modules are utilized to estimate achievable system performance. The adaptive mission planner can then adjust/optimize mission commands based on the estimated performance of the degraded system.

41.3 Real-Time System Health Identification and System Performance Estimation

In order to adapt to the changing operational environment, fault conditions, or performance degradation, it is very important that the IVHM system possesses the

capability to identify/update various models in real time so that the appropriate control can be applied. This chapter will present two system/health identification techniques that have been developed and evaluated in simulation studies using NASA engine and aircraft models. These include (1) a real-time strong tracking system identification (SID) algorithm for adaptive flight control and (2) a self-tuning Kalman filter algorithm for engine health assessment (Tang et al. 2007a). Based on the health assessment and newly identified UAV dynamics, the operational flight envelop in real time can be assessed to ensure safe flight regimes are utilized under compromised conditions.

41.3.1 Real-Time Strong Tracking System ID Algorithm

The discrete state-space model or equations of motion for a UAV can often be stated (in discrete format) as

$$y(n) = \theta(n)^T \phi(n) + v(n) \quad (41.1)$$

where $y(n)$ is a measurable system state or output, $\theta(n)$ is a vector representing the parameters to be tracked/estimated, $\phi(n)$ is the regressor vector, $v(n)$ represents the modeling error, and n is a time step index. For example, if the parameters (C and D) of a linear output equation ($y = Cx + Du$) are being tracked with this algorithm, then $\theta(n)$ would be a vector considering of a row of the C matrix and a row of the D matrix and $\phi(n)$ would be a vector of the states and inputs, i.e., $(x^T; u^T)^T$. The same simple linear formulation can be applied to estimate aerodynamic coefficients of a nonlinear equation of motion when it can be linearized or the nonlinear terms can be wrapped into the regressor vector. The least square algorithm drives $\theta(n)$ toward its true value, θ^* , by a modified least squares algorithm that seeks to minimize an augmented cost function (Bodson 1995):

$$J(\theta(n)) = \frac{1}{2} \sum_{k=n-N+1}^n \lambda^{n-k} \|y(k) - \theta(n)^T \phi(k)\|^2 + \alpha \|\theta(n) - \theta(n-1)\|^2 \quad (41.2)$$

where λ is the time-varying forgetting factor, which will be modified adaptively according to a fault detection module. α is the weighting coefficient that adjusts the influence of the derivation of the current estimate from the previous estimate.

The augmented cost function includes one additional term that restricts the movement of the estimate, $\theta(n)$, in the temporal dimension. Similarly, another term can be added to restrict the movement of $\theta(n)$ in the spatial dimension, e.g., $\beta \|\theta(n) - \hat{\theta}\|^2$, where $\hat{\theta}$ is an a priori estimation of $\theta(n)$. This formulation will further help when $\theta(n)$ does not deviate too much from $\hat{\theta}$ and result in a more

robust estimate in the presence of noisy measurements. However, at the same time, extra computational burden is introduced.

As an improvement for fast detection and tracking in real time, a change detection scheme and a time-varying forgetting factor can be integrated with the above algorithm. When an abrupt change in prediction residuals is detected, a smaller forgetting factor is utilized to discount “old” data and place more weight on the latest measurements. This mechanism helps to improve system response when an abrupt fault occurs.

41.3.2 Self-Tuning Kalman Filter (STKF)

In order to make the onboard subsystem models adaptive to potential performance variations due to performance degradation or anomalous conditions, an STKF can be designed to implement the capability to adjust its performance through the tuning of health parameters which are introduced to the system dynamics as “virtual states.” This technique has been applied to aircraft engine fault diagnosis in recent years (Volponi and Wood 2005). For example, the health parameters for a typical gas turbine engine can be defined as

$$X_c = (\eta_{\text{Fan}} w_{\text{Fan}}, \eta_{\text{LPC}}, w_{\text{LPC}}, \eta_{\text{HPC}}, w_{\text{HPC}}, \eta_{\text{LPT}}, w_{\text{LPT}}, \eta_{\text{HPT}}, w_{\text{HPT}}) \quad (41.3)$$

where X_c is the augmented virtual states including the efficiency and flow capacity of fan, low- and high-pressure compressors, and low- and high-pressure turbines. The tuning parameters are embedded in the Kalman filter design. If sensor outputs deviate from nominal condition values due to component degradation and/or faults, the Kalman filter will attribute the cause of sensor output deviations to the tuning parameters, so that the residuals of state/output estimates will remain small. At the same time, the values of the tuning/health parameters will be a good estimate of engine component health condition as long as the underlining engine model is reasonably accurate.

The accuracy of the underlining subsystem model significantly affects the performance of the fault detection, isolation, and estimation (FDIE) algorithm presented in the previous section. Discrepancies between model outputs and sensor measurements caused by modeling mismatch can be mistakenly considered as a fault effect leading to a false alarm. Therefore, a technical challenge for this type of model-based approach is building an accurate, real-time subsystem model that is robust to unmodeled system dynamics, engine-to-engine variation, and nominal deterioration over time. To solve this problem, empirical modeling is often introduced to capture the modeling mismatch using neural networks (NNs) and improve the accuracy of the physics-based subsystem model. The resulting hybrid model consisting of both a physics-based baseline model and a neural network-based empirical model is often utilized in real-time engine health management applications as illustrated in Fig. 41.2.

Fig. 41.2 Hybrid engine modeling for online real-time engine health management

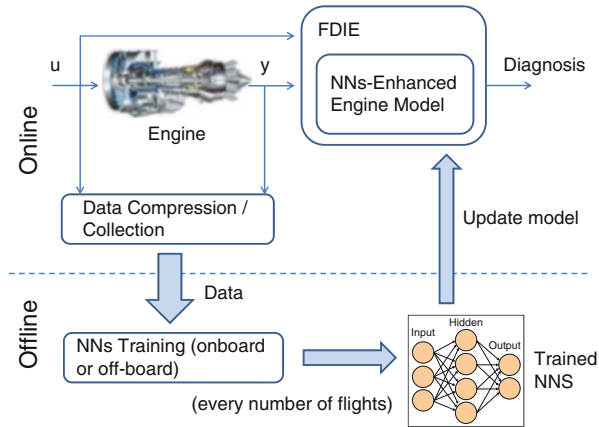
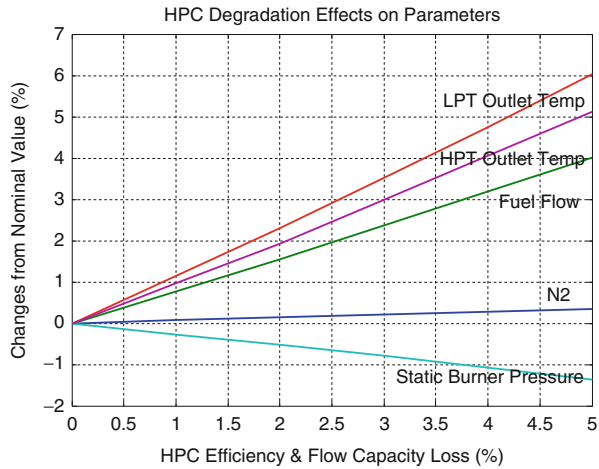


Fig. 41.3 Effects of high pressure compressor degradation on engine parameters



41.3.3 Engine Health Management

Let us further examine the previous engine model tuning example in terms of modeling the potential degradation and how it can be used within the IVHM system architecture. Engine performance monitoring exploits the availability of monitored gas path parameters, including spool speeds, temperatures, pressures, and flow rates at key points in the engine. Degradations in the engine’s performance will commonly manifest themselves as gradual shifts away from expected values corresponding to the current operating state of the engine, as illustrated in Figs. 41.3 and 41.4 (based on simulation results on C-MPASS engine model running at sea level static maximum power). It is recommended that the engine performance monitoring module implement a model-based approach based upon detecting and classifying these shifts as they occur. This technology assesses signal

Fig. 41.4 Effects of low pressure compressor degradation on engine parameters

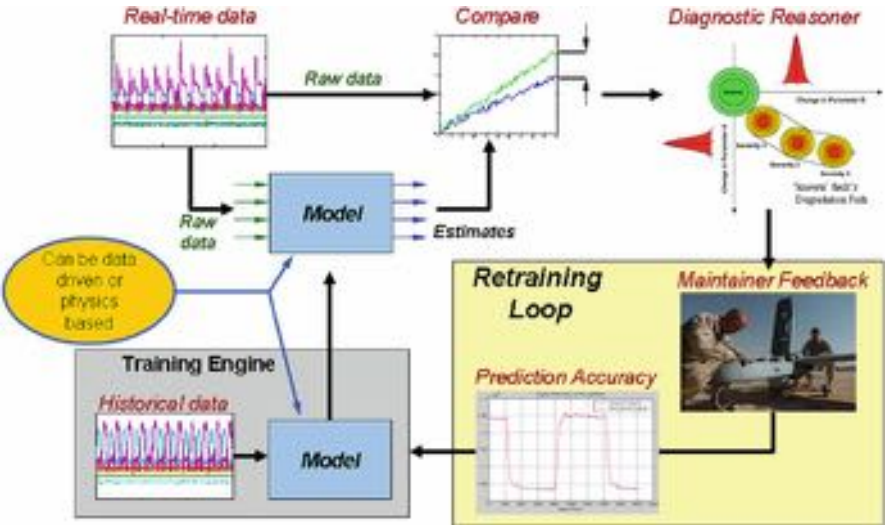
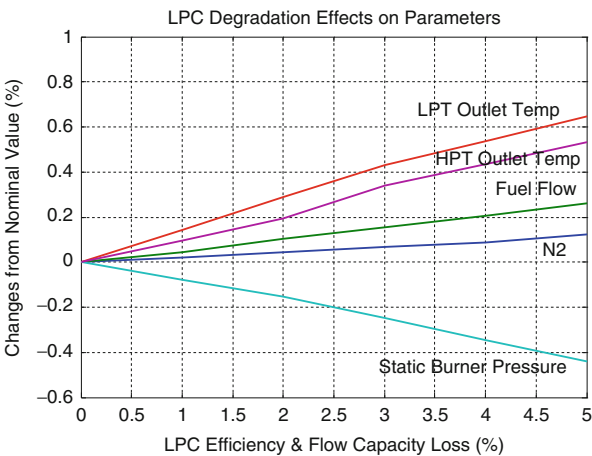


Fig. 41.5 Model based anomaly detection, diagnostics, and trending

health and system performance by employing a combination of signal processing, statistics, and data-driven modeling techniques. The process is illustrated in Fig. 41.5. Measured engine parameter data are input to the model, which provides expected values as output. These expected values are then evaluated relative to the original measured data to determine the residual value. These residual values used together form a pattern which can then compared to the characteristic degradations like those depicted in Figs. 41.3 and 41.4. Subsequent reasoning can then be applied to isolate the source of the performance degradation to a specific section of the engine. The techniques described here have been

implemented successfully in various gas turbine engine applications. Consideration of mechanical vibrations in the engine's moving parts augments the engine performance evaluation to form a more complete propulsion monitoring system. Such techniques can be applied to drive train components specific to the engine such as rolling-element bearings. In addition, engine orders ($1\times$, $2\times$, and higher harmonics of running speed) can be tracked to provide indications of a variety of faults.

Critical component prognostics techniques based on fatigue accumulation models for specific components and engine configurations can provide an estimate of the remaining life before maintenance or overhaul is required. These models are typically a function of specific usage, mission profiles, speeds, and loads and can be adjusted by diagnostic indicators of the existence of faults. If such models are unavailable, then a much simpler approach can be implemented which only tracks total accumulated cycles (TAC) to provide a rudimentary approach to assessing engine life consumption. This approach, applied to gas turbines, monitors very generalized engine speed movements to gauge the severity of thermal transients and assess usage accumulation.

41.3.4 Flight Control Actuator Health Management

Linear actuators are used extensively in UAVs for everything from manipulation of flight control surfaces to weapon release systems. One commonly applied type of actuator is the electromechanical actuator (EMA), which uses an electric motor and gear set to produce linear or rotary motion. EMAs have some advantages over equivalent hydraulic actuators; less weight, size, and complexity, characteristics that allow them to gain acceptance for aerospace applications.

Prognostics and health management for EMAs typically employ two types of approaches to yield an overall assessment of EMA health. A model-based approach utilizing only command and response information from the EMA's controller employs a core group of diagnostic features, including local gear stiffness, frictional damping, and torque constant to determine EMA system health. In this implementation, the model-based approach illustrated in Fig. 41.5 utilizes command signals as input and exercises a model of the actuator system varying the values for the three diagnostic features until a suitable match is found, measured by a minimization of the residual between the modeled response and the measured response. The set of the three diagnostic features then serve as the axes of a three-dimensional failure space wherein different failure modes propagate through the failure space on different, unique paths. A second technique can utilize vibration data collected from an accelerometer to monitor the characteristics of the rolling elements in the actuator. This type of approach utilizes traditional vibration-based features to determine overall actuator health. Results from the two approaches can then be evaluated by a higher-level reasoning structure as described in the next section.

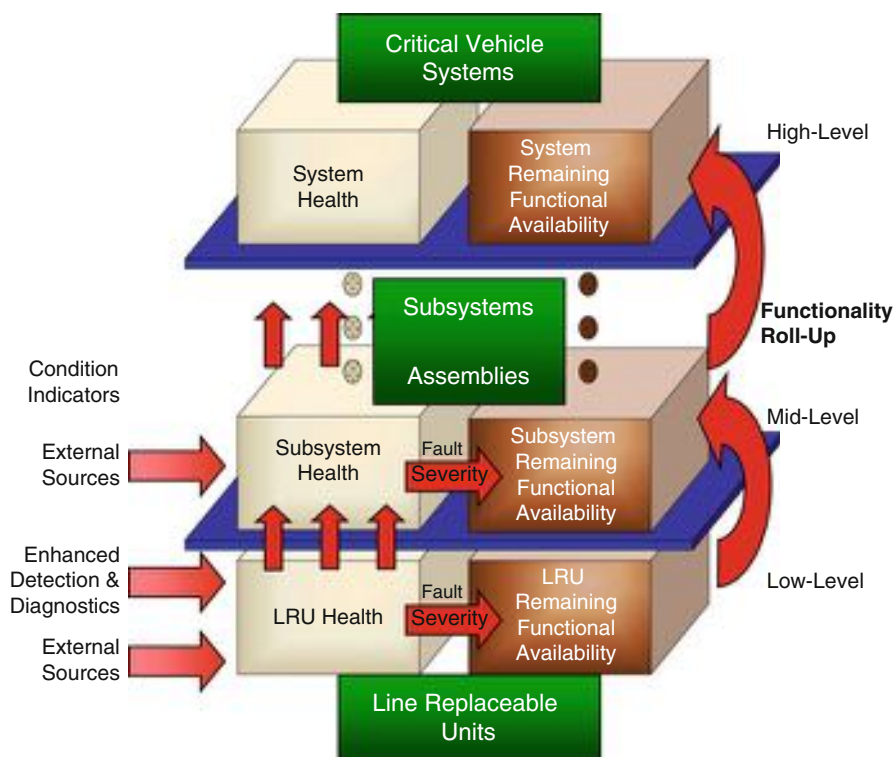


Fig. 41.6 Hierarchical reasoning

41.3.4.1 Vehicle- and Subsystem-Level Reasoning

One of the key aspects distinguishing an IVHM approach from a basic health and usage monitoring capability is an enhanced reasoning capability utilized to provide a vehicle-wide assessment of the implications of any detected failure mechanisms on the overall capability of the vehicle. Reasoning, within an IVHM system, processes condition or health indicators obtained from the subsystem-specific health modules to determine the current health state of the vehicle and its constituent subsystems. This goes beyond the basic diagnoses of observed indications by attempting to quantify the remaining functionality, which has been reduced by the indicted failure mode(s). The reasoning engine employs a hierarchical architecture, as illustrated in Fig. 41.6, to not only identify failure modes of line replaceable components (LRC) but also determine the functional impact in terms of remaining functional/operational availability at the subsystem and vehicle levels. When analyses result in an ambiguous outcome, the reasoner works to isolate the root cause or at least reduce the ambiguity group. The reasoner also attempts to provide estimates of the severity of the underlying failure and the remaining useful life of the LRC where prognostic capabilities are available. The reasoned results output by

the IVHM system provide real-time health and remaining functionality information useful to operations personnel for decision support.

At the lowest level, diagnostic reasoning seeks to classify latent failure mode indications from raw sensor data or diagnostic feature data processed by the subsystem-specific modules. The mid-level of the reasoning architecture is employed to determine the overall functional availability of the constituent subsystems, i.e., what are the implications of the detected failure modes on the functional availability of the subsystem? The vehicle- or system-level reasoning, the highest level of onboard reasoning, shares this task of determining and quantifying functional availability, but from a vehicle- or system-wide perspective. At this highest level, the functional availability assessments, or condition indicators, from all underlying subsystems are utilized to determine the capability of the vehicle to continue operations as expected or needed.

41.3.4.2 Prognostics

Performing onboard prognostics for a given set of LRUs or subsystems within the UAV IVHM system architecture is desirable but often times a challenge due to the necessary information and models needed to perform the task in real time. One such technique that has been shown to be useful for performing on-line prognostics for critical UAV subsystems is called the particle filter. Particle filtering is an emerging and powerful methodology for sequential signal processing based on the concepts of Bayesian theory and sequential importance sampling (SIS) and is suitable for the embedded applications onboard the IVHM system's embedded element. Particle filtering can be used for a wide range of applications in science and engineering and is very suitable for nonlinear systems or in the presence of non-Gaussian process/observation noise. For example, the technique has been successfully used in the diagnosis and prognosis of a helicopter gearbox fault, gas turbines, and electrochemical batteries at Impact. The fault diagnosis procedure fuses and utilizes the information present in a feature vector (observations) with the objective of determining the operational condition (state) of a system and the causes for deviations from desired behavioral patterns. From a nonlinear Bayesian state estimation standpoint, this task is accomplished by the use of a particle filter-based module built upon the nonlinear dynamic state model.

One particular advantage of this particle filtering approach is the ability to characterize the evolution in time of the above-mentioned nonlinear model through modification of the probability masses associated with each particle, as new feature information is received. Furthermore, the output of the fault diagnosis module, defined as the current expectation of each Boolean state, provides a recursively updated estimation of the probability for each fault condition considered in the analysis. The probability density function (PDF) estimates for the system's continuous-valued states provide swift transition to failure prognosis algorithms, which is a primary advantage for a particle filter-based diagnosis framework. Since prognosis intends to project the current condition of the indicator in the absence of future measurements, it necessarily entails large-grain uncertainty. These facts suggest a prognosis scheme based on recursive Bayesian estimation techniques, combining

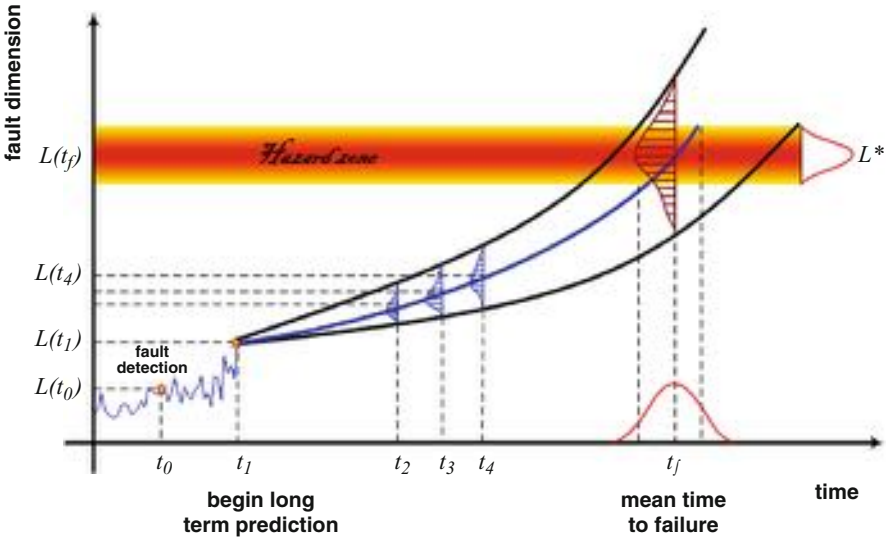


Fig. 41.7 Illustration of prognostics

both the information from fault growth models and on-line data obtained from sensors monitoring key fault parameters (observations or features).

The particle filtering-based approach for prognosis bases long-term predictions for the failure evolution on both an accurate estimation of the current state and a model describing the fault progression. This procedure tends to reduce the uncertainty associated with long-term predictions by using both the current state PDF estimation and a record of corrections made to previous computed predictions. Thus, the particle filtering algorithm serves the dual purpose of managing uncertainty and providing a means to continuously adapt the fault progression model. The probability of failure at any future time instant is estimated by combining both the weights of predicted trajectories and specifications for the hazard zone, as shown in Fig. 41.7: 6. The resulting Remaining Useful Life PDF provides the basis for the generation of confidence intervals and expectations for a prognosis.

41.3.5 Dynamic Flight Envelope Assessment

The flight envelope of a UAV or any air vehicle represents the ranges of altitudes and speeds over which it effectively operates. This envelope may be drawn as a series of constraining curves on the altitude-Mach number plane. These constraints may arise owing to aerodynamic or propulsion considerations (e.g., the characteristics of the flight surfaces wings, and ailerons) or the power plant characteristics. The flight envelope (as determined for any particular aircraft) is usually an approximation and may be represented as a “doghouse plot” as shown in Fig. 41.8.

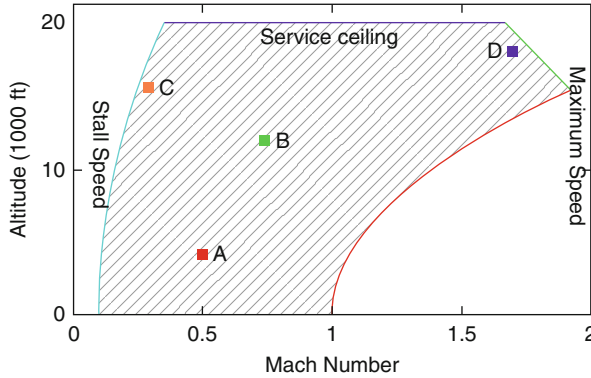


Fig. 41.8 Representative flight envelope

The flight envelope in Fig. 41.8 is bounded by four curves. Points which are well within the envelope (such as “A” and “B”) represent “comfortable” operating points, where the aircraft has considerable excess capacity for maneuvers. Points close to the edges of the envelope (“C” and “D”) represent operation of the aircraft close to its limits, and thus, there is limited capacity for maneuvers at these points.

Faults in UAV subsystems may affect the flight envelope significantly. Anomalies may occur in a variety of forms such as abrupt changes in aerodynamics, failed actuators, loss of propulsion, and damaged structures or control surfaces. In order to maintain vehicle stability and safe flight under adverse conditions, detection of the aircraft current state and prediction of achievable performance must be accomplished through onboard systems, in real time, and communicated to the flight control system, mission management system, and pilot. This onboard system must allow decision making to reduce risk while operating in the presence of uncertain information from various subsystems.

For example, severe engine faults may reduce the available thrust, which affects the maximum speed and the service ceiling curves. Similarly, damages in the control surfaces may cause loss of lift or increased drag. This would lead to higher stall speeds, while also affecting the service ceiling and maximum speed. Thus, the flight envelope would shrink due to engine or flight control surface faults.

The maximum speed attainable by the aircraft depends on the maximum engine thrust, as well as the wing area and the minimum attainable drag coefficient. This may be expressed as

$$\begin{aligned}\tau_{\max} &= \frac{1}{2} \rho(H) C_{d_{\min}} A_w V_{\max}^2 \\ V_{\max} &= \sqrt{\frac{2\tau_{\max}}{\rho(H) C_{d_{\min}} A_w}}\end{aligned}\quad (41.4)$$

where τ_{\max} is the maximum thrust, ρ is the density of air (as a function of altitude H), $C_{d_{\min}}$ is the minimum drag coefficient, A_w is the wing area, and V_{\max} is the

maximum possible air speed. Thus, loss of thrust due to engine degradation as well as increased drag owing to wing damage may both reduce the maximum speed, which shrinks the flight envelope to the right.

Similarly, the stall speed may be calculated as

$$W = \frac{1}{2} \rho(H) C_{L_{\max}} A_w V_s^2$$

$$V_s = \sqrt{\frac{2W}{\rho(H) C_{L_{\max}} A_w}} \quad (41.5)$$

where W is the weight of the aircraft, $C_{L_{\max}}$ is the maximum lift coefficient, and V_s is the stall speed. Reduced lift (owing to wing faults) could thus cause the stall speed to increase, which shrinks the flight envelope to the left.

Due to the experimentation and modeling capabilities available to the author, dynamic flight envelope estimation was initially investigated for severe engine faults. The reduction in the flight envelope due to an engine fault was simulated using the NASA MAPSS engine model (Parker and Melcher 2004). Specific engine faults were seeded into the MAPSS model which causes significant efficiency and flow capability loss in several engine components. The corresponding changes in the flight envelope were then estimated in real time by estimating the maximum thrust that the engine could produce under a particular level of degradation. As illustrated in Fig. 41.9, the maximum thrust estimate was provided by an artificial neural network which was trained off-line using the data from high-fidelity engine model simulations with engine controllers in the loop.

41.4 ACM System Optimization

Most ACM strategies for large-scale dynamical systems, such as an aircraft, rely on heuristic information about a reduced set of severe, frequent, and testable fault modes, a reasonable number of active controllers, and a mapping between the fault modes and the control reconfiguration routines. Such a strategy has the ability to adaptively switch from one controller to another, if control limits are reached, and by this switching action, critical mission objectives can be realized. This chapter presents a new approach in which an optimization problem is dynamically formulated and solved on-line to solve the optimal contingency strategy constrained by the available performance and resource. A typical cost model is expressed in terms of such elements as on-time execution of critical components, time to complete the mission, tracking error, fuel consumption, etc. Costs are computed in real time and may change dynamically.

Analytically, the objective of the ACM system is to optimize the utility of the vehicle with impaired capability to accomplish an assigned mission. The ACM system can be formulated as an optimization problem in two levels:

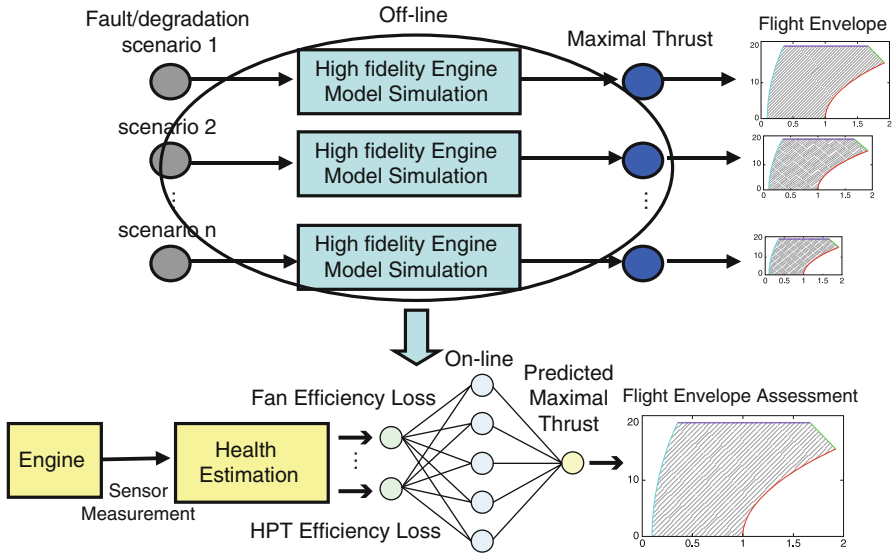


Fig. 41.9 Neural network-based dynamic flight envelope estimation for severe engine faults

High (mission)-level planner:

$$J(M) = \max_M U(P_e, P_r, M, M_{\text{com}}) \quad (41.6)$$

Lower (control reconfiguration) level:

$$J(R) = \max_R P_e(F_m, P_r, R, M) \quad (41.7)$$

where U is a cost function that quantifies the usefulness of the vehicle to accomplish its mission. U is a function of the available prognostic information (P_r), the system's closed loop performance (P_e), and the mission objectives (M and M_{com}). P_e is a function of fault mode F_m , future prediction P_r , as well as any restructuring/reconfiguration R applied to the system and current mission objective M . F_m is a vector of indicators (0 or 1) that characterizes the fault modes detected on the aircraft; R is a vector of indicators that characterizes all restructuring applied to the system. M_{com} describes the mission assigned to the aircraft. M allows the fault-tolerant control architecture, specifically the mission adaptation and resource management components, to modify the parameters of the assigned mission and redistribute available resources based on the vehicle's current performance, P_e . At the high level, mission adaptation and resource redistribution (M) allow the control architecture to pursue relaxed mission objectives in order to achieve greater vehicle usefulness U . At the lower level, the objective is to optimize vehicle performance P_e while satisfying the mission constraints, through

restructuring and reconfiguration, R . Practically, the above optimization problems have to be solved while adhering to various constraints including system dynamics and resource limitations.

While this generic problem formulation applies to all of the ACM levels (mission management, adaptive flight control, and propulsion control) in Fig. 41.1, the focus of this chapter is adaptive flight control. Fault-tolerant propulsion controls and mission reconfiguration are discussed in the previous publications (Tang et al. 2005, 2007b).

41.4.1 Model Predictive Control

Model predictive control (MPC) is a particularly effective method for control reconfiguration due to its ability to handle constraints and changing model dynamics systematically. MPC relies on an internal model of the system, which can be identified on-line in real time using the strong tracking system identification algorithms described in previous section.

System failures can be handled naturally in an MPC framework via changes in the input constraints and internal model. For example, actuator limit and rate constraints can be written as:

$$\begin{aligned} u_i^l &\leq u_i(t) \leq u_i^h \\ \Delta u_i^l &\leq \dot{u}_i(t) \leq \Delta u_i^h \end{aligned} \quad (41.8)$$

for actuator inputs u_1 through u_m . If actuator i becomes jammed at position \tilde{u}_i , the MPC controller can be made to accommodate the changes by simply changing the constraints on input i to

$$\begin{aligned} \tilde{u}_i &\leq u_i(t) \leq \tilde{u}_i \\ 0 &\leq \dot{u}_i(t) \leq 0 \end{aligned} \quad (41.9)$$

Structural failures can also be handled in a natural fashion by changing the system dynamics model used to make prediction in either an adaptive fashion, a multi-model switching scheme or in this case, by using a real-time system identification module to provide the estimated degraded system dynamics.

MPC fault-tolerant control can be cast as a constrained model-following problem as shown in Fig. 41.10. The controller comprises three main components: the block “real-time system ID” which performs identification of the fault’s effects on system dynamics, a “reference model” which uses pilot commands to generate a reference trajectory for the aircraft’s state vector, and the MPC controller whose objective is to track the reference trajectory, using the output of “real-time system ID” to update its internal model, constraints, etc. The pilot gives commands to the reference model, and the goal of the controller is to cause the aircraft to track the resultant trajectory. At each time step, the MPC controller chooses an input sequence which minimizes the difference between the predicted future trajectory, given by the reference model

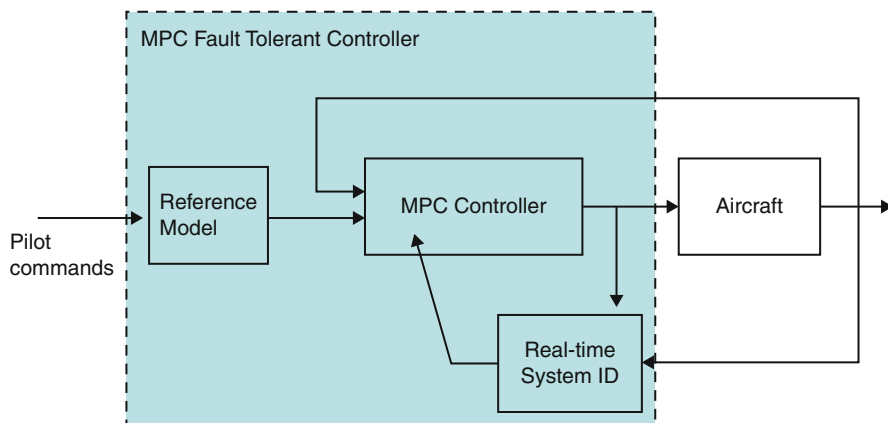


Fig. 41.10 MPC fault tolerant control architecture

under the assumption that the pilot's inputs are constant over the prediction horizon, and the predicted trajectory of the aircraft.

41.5 Flight Envelope Assessment and Fault-Tolerant Control Simulation Results

The NASA MAPSS engine model (Parker and Melcher 2004) was utilized to simulate engine faults for the evaluation of the fault detection and resulting dynamic flight envelope assessment techniques previously described. The self-tuning Kalman filter was used to identify and estimate the severity of the faults in terms of efficiency loss and/or flow capacity loss. At the flight control level, the NASA GTM aircraft model (Bailey et al. 2005) was utilized, and various actuator faults including control effectiveness loss and stuck actuators were simulated. Some select simulation results are presented below.

41.5.1 Dynamic Flight Envelope Assessment

In order to quantify the flight envelope, it was assumed that the thrust produced by the MAPSS model was utilized by an aircraft with certain predetermined characteristics. The characteristics of the aircraft are listed in Table 41.1. Within the simulation, pressure and temperature were varied as shown in Fig. 41.11, with the pressure assumed to drop roughly exponentially with altitude, while the temperature drops linearly with altitude up to a point, beyond which the temperature stays constant.

Table 41.1 Aircraft characteristics

| Parameter | Explanation | Value | Units |
|------------|---------------------------------------|--------|-------|
| A_w | Surface area of wings | 27.88 | m^2 |
| C_{Lmax} | Maximum lift coefficient before stall | 1.6 | None |
| C_{dmin} | Minimum drag coefficient | 0.01 | None |
| Φ | Equivalence ratio at sea level | 0.7 | None |
| M | Total mass of aircraft | 10,780 | Kg |
| T_{SL} | Sea level thrust | 130 | kN |

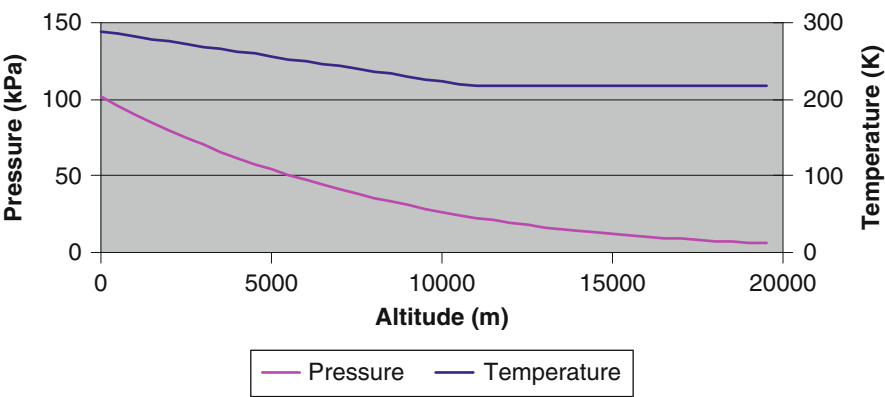


Fig. 41.11 Variation of pressure and temperature with altitude

Figure 41.12 shows a plot of the instantaneous flight envelope for a given level of engine fault. The NASA MAPSS software was modified with a new plot showing the estimated flight envelope. The result in Fig.41.13 shows how the flight envelope shrinks as a function of the engine fault severity level (in this case, the HPT efficiency loss).

41.5.2 Detection of Control Effectiveness Loss

With the strong tracking recursive identification algorithm, the A and B matrix of the linearized aircraft state-space model are identified in real time. Various scenarios including elevator/rudder/aileron/throttle effectiveness loss and aircraft weight loss have been simulated and evaluated with the fault-tolerant controller.

Figure 41.14 shows the identified control effectiveness change by the strong tracking recursive identification algorithm and the comparison with the result from classic recursive least square method. The true value is changed from 1.98 to 0.198

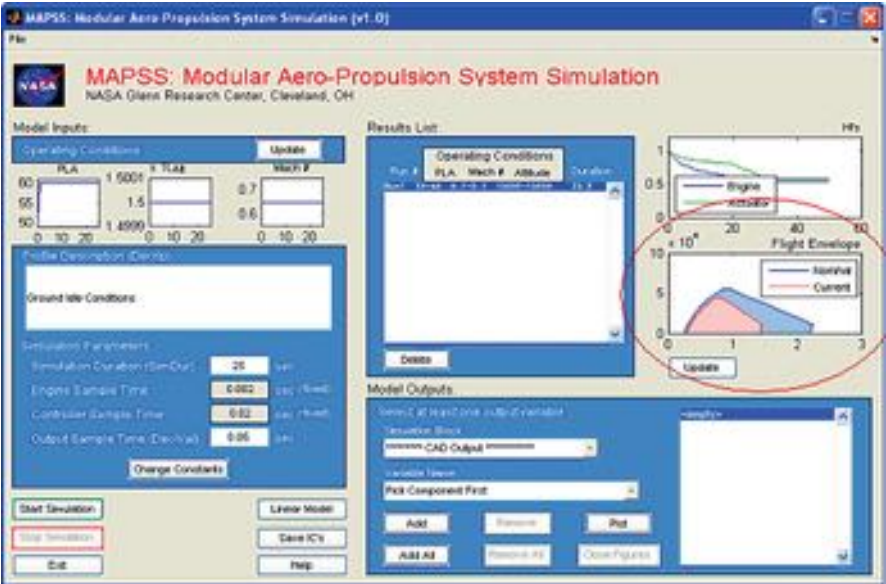


Fig. 41.12 Dynamic flight envelope assessment results added to NASA MAPSS GUI

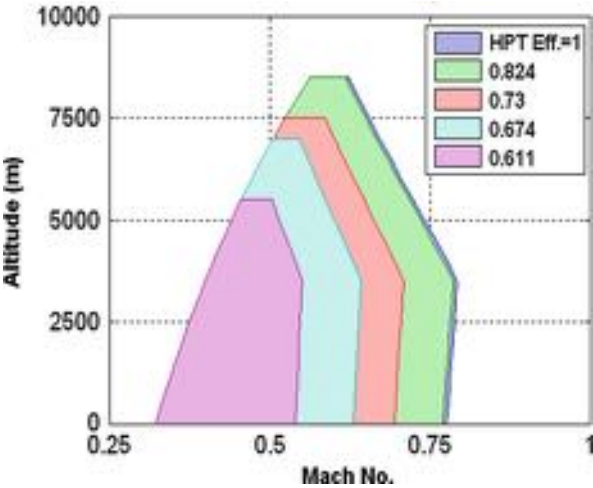


Fig. 41.13 Flight envelope reduction due to severe engine fault added to NASA MAPSS GUI

at $t = 20$ s (i.e., a 90 % control effectiveness loss). The result shows that the tracking performance of the strong tracking recursive identification algorithm (blue solid curve) outperforms the classic recursive least square method algorithm (red dashed curve) in tracking a sudden change in control effectiveness of the elevator.

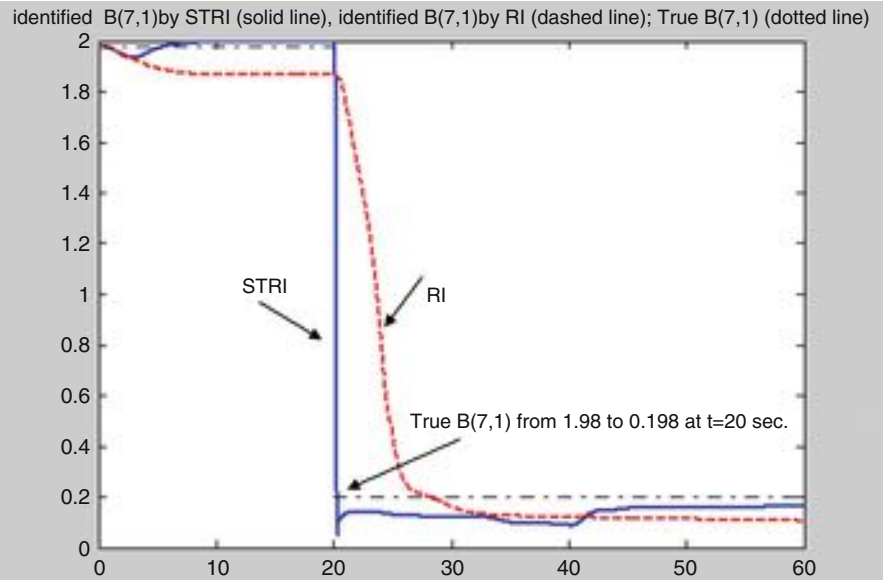


Fig. 41.14 Detection of control effectiveness loss

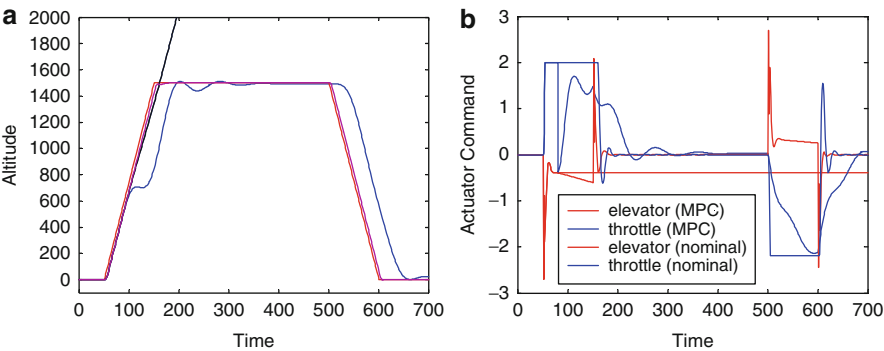


Fig. 41.15 MPC simulation results with stuck elevator actuator ($t = 75$). (a) Altitude response and (b) controls

41.5.3 Model Predictive Control for Stuck Elevator

An adaptive model predictive controller that utilizes the system dynamic model identified in real time was developed and tested with a stuck elevator fault scenario. A classic altitude hold controller was developed which utilizes throttle to control airspeed and elevator for altitude tracking. As shown in Fig. 41.15, after the occurrence of a stuck elevator fault at time 75, the aircraft would keep climbing up

without the fault-tolerant controller because the elevator is stuck at a climbing-up position (the black curve in Fig. 41.15a). With the adaptive MPC controller, throttle is used to control longitudinal dynamics, and the aircraft was able to accomplish the mission (the red curve in Fig. 41.15a) with degraded performance (the dashed blue curve in Fig. 41.15a). The elevator and throttle controls and the comparison with nominal controls are shown in Fig. 41.15b.

41.5.4 Prognostics-Enhanced ACM System

The safety of the unmanned vehicle involved in long-duration missions is dependent on not only the current health state (diagnostics) but also near-future health states (prognostics) as well. While a diagnostics-driven ACM system can react to and compensate for faults and performance degradation after they are detected, it cannot hope to overcome the fact that it will always be a reactive paradigm. As a result, the control strategies may in fact not be optimal over a longer period of time. By incorporating near-term prognostic information, undesirable future states of the system may be avoided or deferred through a suitable change in the control strategy while achieving required (or relaxed) mission objectives. In case a failure in certain system component is inevitable, a prognostics-enhanced ACM system can possibly minimize the effect or propagation of the failure by appropriate control reconfiguration and isolation strategies.

To illustrate the concept, one can consider the case of emergency landing of a distressed aircraft with a rudder failure (floating rudder with total loss of control authority). The loss of control is taken up by the engines operating in a thrust-enhanced, differential throttle mode. However, traditional engine time constants are too slow to be used in this situation. A fast-response engine controller is needed to reduce the time constants by pushing the engine operation more aggressively. As a result, engine stall margin and engine life can be significantly reduced which may lead to catastrophic engine failure. Therefore, competing requirements on aircraft performance (demanding aggressive engine operation) and engine safety (demanding mild engine operation) must be considered within the ACM framework.

The ACM strategy implemented is a supervisory safety verification controller (SSVC) which is a middle-level controller that determines low-level control configuration (switching between slow-response and fast-response engine controllers) at a certain decision interval. This level of control specifies parameters that the low-level controllers must assume in order to reallocate the controls once a fault has been diagnosed. It must be noticed that the low-level controllers (either slow or fast response) are responsible for the stability and performance of the engine while the SSVC optimizes the engine operation to achieve required aircraft performance while avoiding any violations of engine safety limits. Since the SSVC only operates at a near real-time decision interval (typically every couple of seconds), more computationally demanding optimization algorithms can potentially be applied at this level of controls.

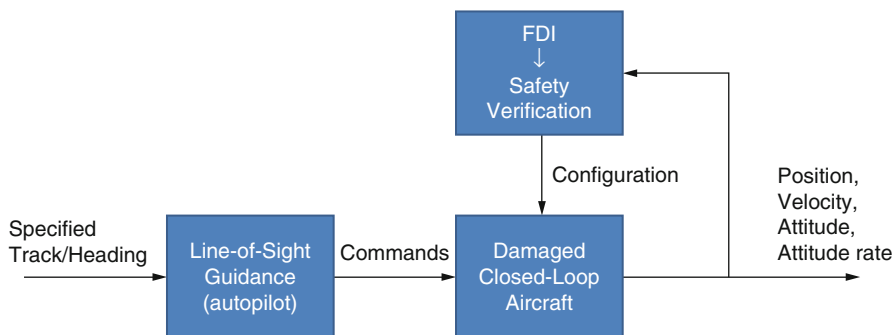


Fig. 41.16 Diagram of the safety verification controller using diagnostics and prognostics

Figure 41.16 illustrates the overall structure of a supervisory safety verification controller. The aircraft states including diagnostic and prognostic information are fed into an FDI/safety verification module which, in turn, computes the optimal control configuration given the fault and mission profile. The adopted configuration dictates which low-level controls are used and how the control is allocated (i.e., whether a conservative or aggressive policy should be adopted at any given time). A cross-track guidance system is adopted in lieu of a pilot as a baseline to provide commands to the flight controller (DeCastro et al. 2011).

For this emergency landing case study with rudder failure, the UAV's contingency envelope can be defined by three criteria:

1. Performance criteria: The maximum allowable error from the reference trajectory is 1,500 feet.
2. Operability limits: Engine stall margin is maintained above zero to avoid immediate loss of thrust and potential engine failure.
3. Prognostic (life) limits: The prognostic criteria penalize any trajectory or control policy that results in a large probability of accumulated engine failure by the end of the mission.

When considering engine prognostic algorithms, various types of prognostic algorithms may be used to track engine damage, including particle filter, Kalman filter, or data-driven algorithms. In order to use these algorithms in the supervisory control, the essential behavior of the algorithm is abstracted to a finite-state machine (FSM). The automaton discretizes the trajectory of the engine damage index and assigns transition probabilities to each node as a discrete-time Markov chain. Figure 41.17 shows how the probability of crossing the failure threshold is mapped to an FSM. The unsafe set corresponds to the failed state of the engine. It is important to note that the abstraction allows the prognostic model to be captured in the ACM optimization in a simplified way without losing essential information.

To demonstrate the overall reconfiguration concept, simulations are executed with the GTM simulation using only fast and slow engine control reconfiguration options. In the following scenarios, a new control decision is made every 10 s.

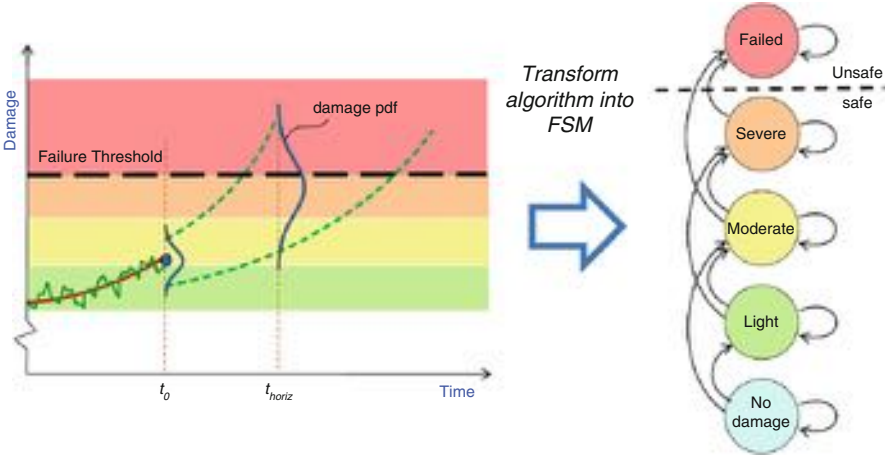


Fig. 41.17 Transforming a prognostic algorithm into a finite-state machine

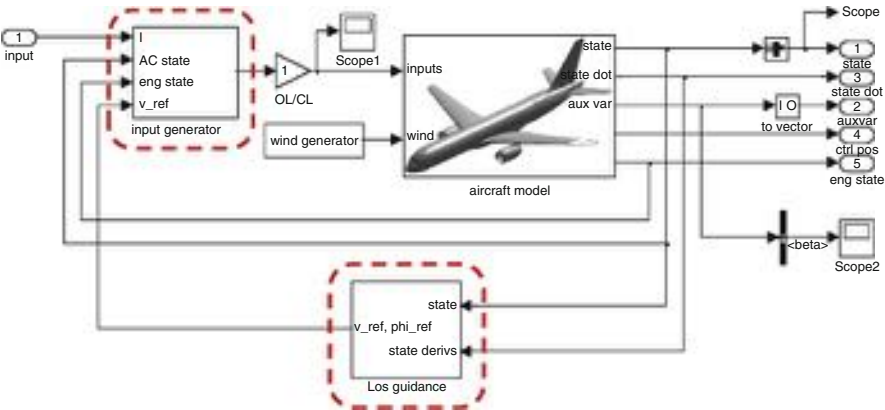


Fig. 41.18 Simulink diagram of the NASA full-scale GTM with safety verification control capability

The additive uncertainties to simulate effects such as wind gusts were modeled as Gaussian noise on each of the aircraft’s states. A Simulink model developed for this study is shown in Fig. 41.18.

In this simulated scenario, the GTM is making a series of coordinated turns to reduce the cross-track error to a specified parallel track to prepare for landing. The cross-track trajectory is shown in Fig. 41.19. For the 200-s simulation, the reconfiguration policy along with the trajectories on the aircraft’s lateral position and the stall margin between engines are shown in Fig. 41.20. The policy switching is dominated initially by the stall margin requirement by starting in the slow configuration. After 80 s, the control attempts to meet the cross-tracking performance

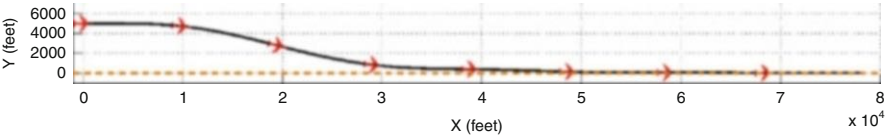


Fig. 41.19 UAV cross-track trajectory

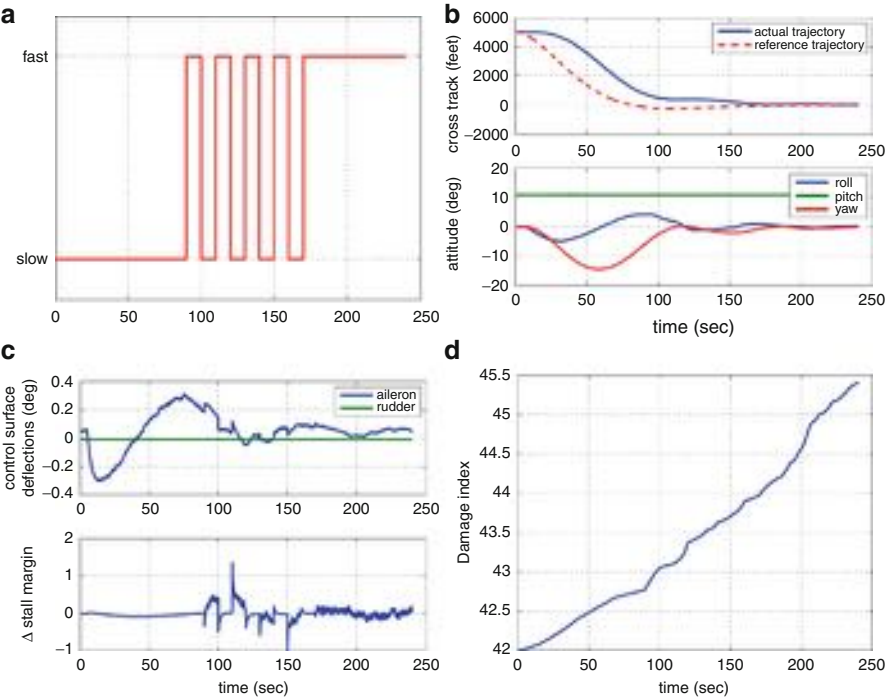


Fig. 41.20 Optimal configuration and states for SSVC with consideration of prognostics. (a) Optimal control sequence, (b) aircraft states, (c) aircraft control inputs and engine stall margin, and (d) engine damage index

requirement by switching to the fast configuration. The ACM strategy switches at every decision instant in order to improve the remaining useful life of the engine, thereby reducing the probability of failure before the end of the mission. Although the controls are not configured to mitigate bumpy transfer, the switching does not introduce oscillations in the overall response of the aircraft. Both the cross-track performance error and stall margin remain below thresholds throughout the horizon. The figure also shows the role of the prognostic model on the ensuing damage accumulated on the engines during the maneuver. The hypothetical hybrid abstraction of the prognostic algorithm is applied, assuming that the system is

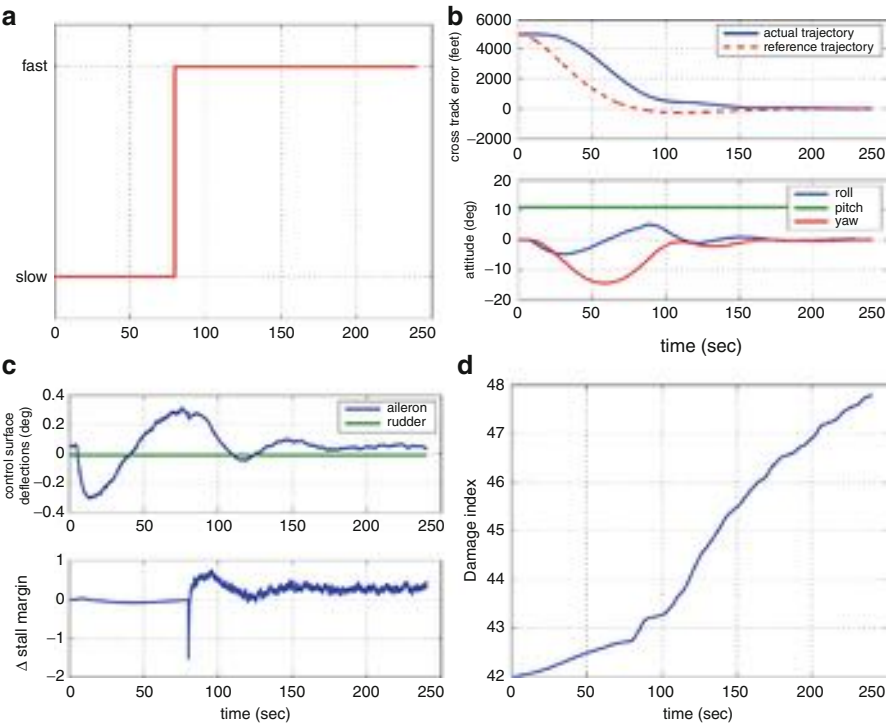


Fig. 41.21 Optimal configuration and states for SSVC without consideration of prognostics. (a) Optimal control sequence, (b) aircraft states, (c) aircraft control inputs and engine stall margin, and (d) engine damage index

initially in the moderate damage state. When prognostic information is included in the optimization, the damage index reaches 45.5 at the end of the mission.

As a comparison, the UAV cross-track control results without engine prognostics in the loop are shown in Fig. 41.21. Similar to the previous result, the ACM policy switching is dominated initially by the stall margin requirement by starting in the slow configuration. After 80 s, the control attempts to meet the cross-tracking performance requirement by switching to the fast configuration and remains in this control configuration for the duration of this simulation. As a result, the engine damage index has grown to about 48 (as compared to 45.5 in the previous simulation) at the end of the simulation. This is a clear evidence of an engine safety threat when engine life (prognostics) has been ignored in the ACM optimization.

41.6 Conclusions and Future Work

This chapter presented selected technologies for implementing an integrated aircraft health assessment and fault contingency management system. An innovative hierar-

chical fault contingency management architecture that integrates real-time system health management functionality, high-level reasoning, control reconfiguration, and automated contingency management modules has been developed. Real-time system identification and health assessment algorithms have been implemented to identify the dynamics and performance limitation of the degraded system. An intelligent approach has been developed to estimate safe flight envelope after the occurrence of engine faults. MPC-based reconfigurable fault-tolerant control techniques have been developed and tested in simulation to accommodate flight actuator faults using the system dynamic model identified in real time. Simulation studies using the NASA MAPSS engine model and GTM aircraft model have successfully demonstrated the feasibility of the presented concept and architecture. This chapter further explored a more advanced topic on incorporating prognostic information in a proactive ACM system. Simulation studies on a GTM aircraft model have been conducted and demonstrated the benefits of using near-term prognostics in control reconfiguration. This capability upgrades the conventional diagnostics-based reactive fault-tolerant strategies to a prognostics-enhanced proactive fault accommodation paradigm.

To further develop the ACM concept and eventually integrate it into real-world autonomous systems, several important issues need to be addressed. First, the diagnostic and prognostic information fed into the ACM system have to be reliable, accurate, and available in real time. This is certainly a challenging requirement on the design of onboard PHM system. Prognosis uncertainty management is another important issue that affects the performance of the ACM system significantly. In practice, accurate and precise prognostics has proven rather difficult to accomplish; thus, uncertainty management and reduction techniques have to be implemented before the prognostics can be optimally utilized by the ACM system. Furthermore, in order for ACM systems to be used in safety-critical aerospace applications, they must be proven to be highly safe and reliable. Rigorous methods for ACM system verification and validation (V&V) must be developed to ensure that ACM+P system software failures will not occur, to ensure the system functions as required, to eliminate unintended functionality, and to demonstrate that certification requirements can be satisfied.

References

- R.M. Bailey, R.W. Hostetler, K.N. Barnes, C.M. Belcastro, C.M. Belcastro, Experimental validation: subscale aircraft ground facilities and integrated test capability, in *AIAA Guidance, Navigation, and Control Conference*, Washington, DC, 2005
- M. Bodson, An adaptive algorithm with information-dependent data forgetting, in *Proceedings of the American Control Conference*, Seattle, WA, 1995, pp. 3485–3489
- C.S. Byington, M. Watson, D. Edwards, P. Stoelting, A model-based approach to prognostics and health management for flight control actuators, in *Proceedings of the IEEE Aerospace Conference*, Big Sky MN, paper 1047, 2004
- J.A. DeCastro, L. Tang, B. Zhang, G.J. Vachtsevanos, A safety verification approach to fault-tolerant aircraft supervisory control, in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Portland, OR, 8–11 Aug 2011

- J.S. Litt, D.L. Simon, S. Garg, et al., A survey of intelligent control and health management technologies for aircraft propulsion systems. *J. Aerosp. Comput. Inf. Commun.* **1**(12), 543–563 (2004)
- K.I. Parker, K.J. Melcher, The modular aero-propulsion system simulation (MAPSS) users' guide, NASA TM-2004-212968, 2004
- L. Tang, G.J. Kacprzynski, M.J. Roemer, G. Vachtsevanos, A. Patterson-Hine, Automated contingency management design for advanced propulsion systems, in *Infotech@Aerospace*, Arlington, VA, 26–29 Sept 2005
- L. Tang, M. Roemer, G. Kacprzynski, J. Ge, Dynamic decision support and automated fault accommodation for jet engines, in *IEEE Aerospace Conference*, Big Sky, MT, 3–10 Mar 2007 (2007a)
- L. Tang, G. Kacprzynski, K. Goebel, J. Reimann, M. Orchard, A. Saxena, B. Saha, Prognostics in the control loop, in *The 2007 AAAI Fall Symposium on Artificial Intelligence for Prognostics*, Arlington, VA, 9–11 Nov 2007 (2007b)
- G. Vachtsevanos, L. Tang, G. Drozeski, L. Gutierrez, From mission planning to flight control Of unmanned aerial vehicles: strategies and implementation tools. *Ann. Rev. Control* **29**, 101–115 (2005)
- G. Vachtsevanos, F.L. Lewis, M. Roemer, A. Hess, B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems* (Wiley, Hoboken, 2006)
- A. Volponi, B. Wood, Engine health management for aircraft propulsion systems, in *First International Forum on Integrated System Health Engineering and Management in Aerospace*, Napa, CA, 7–10 Nov 2005

Neal A. Snooke

Contents

42.1 Introduction 1028

 42.1.1 Prognostics 1028

42.2 Model-Based Reasoning 1029

42.3 Failure Modes and Effects Analysis 1031

42.4 Functional Description 1034

 42.4.1 Generating FMEA Results 1035

42.5 Sensor Selection and Diagnosability Analysis 1038

 42.5.1 Generating Symptoms from an FMEA 1041

 42.5.2 Associating Measurement to Functions 1043

 42.5.3 Fault Exoneration 1044

 42.5.4 FMEA Coverage and Symptom Generation 1044

42.6 Diagnosability Assessment 1047

42.7 Conclusion 1050

References 1050

Abstract

This chapter describes how model-based simulation can be employed to automatically generate the system-level effects for comprehensive sets of component failures on systems within the aircraft. The results of the simulation can be used in several ways. They can be used to produce a system-level failure modes and effects analysis (FMEA) for aircraft systems. They can be used to identify the sensors necessary to discriminate remotely between different failures on the aircraft. Once a set of sensors have been chosen for placement on the vehicle, the simulation results can also be used to generate diagnostic and prognostic software for deployment on the vehicle.

N.A. Snooke (✉)
Department of Computer Science, Aberystwyth University, Llandinam Extension, Aberystwyth,
Ceredigion, UK
e-mail: nns@aber.ac.uk

Using automated FMEA safety analysis software is more efficient than doing the same work without software and also provides a guaranteed level of performance. Using the results of this analysis can provide sensor selection and diagnostic capability while retaining some of the benefits of rule-based diagnostic systems. Alternative model-based techniques have been widely used to create diagnostic systems in a variety of domains, and these approaches are compared with the diagnostic capability provided by a failure effect-oriented technique from the perspective of the UAV application.

42.1 Introduction

Prognostics and health management (PHM) comprises the technology and systems needed to enable UAVs to monitor their own state. Based on that monitoring, the PHM systems need to make assessments of mission readiness that can be utilized by higher-level planning in the decision modeling area. One recent UK Project specifies the goal that for use in unsegregated airspace, PHM systems should be capable of replicating the fault detection, assessment, and decision-making ability of pilots. The current generation of computer systems falls far short of the high-level reasoning and decision-making capabilities of an experienced pilot; however, this limitation can be mitigated by the use of sophisticated and high-coverage lower-level fault detection and reconfiguration capabilities.

This chapter will show how model-based reasoning and simulation can be used to automate the process of producing a comprehensive failure modes and effects analysis and how this information can then be used to assist in the assessment of diagnostic capability and sensor selection. The techniques of this chapter were combined and integrated during the first phase of a UK government-funded program: Autonomous Systems Technology Related Airborne Evaluation and Assessment (Astraea 2009). The aim of the ASTRAEA program is to enable the routine use of UAS (Unmanned Aircraft Systems) in all classes of airspace without the need for restrictive or specialized conditions of operation. This is achieved through the coordinated development and demonstration of key technologies and operating procedures required to open up the airspace to UAS.

42.1.1 Prognostics

Prognostic capability provides a forecast of some future occurrence that may impact on a system. This general concept is interpreted in different ways dependent on the level of abstraction being used. To component engineers, the aim is usually to predict component failure or wear out before failure or degradation occurs that will negatively impact on system behavior. Competing objectives for reduced maintenance schedules and cost lead to activities such as condition-based maintenance. Prognostics at this level are inherently difficult and require detailed numerical monitoring and usage data. In some domains such as mechanical systems symptoms

such as increased vibration can provide prognostic indicators; however, in other areas such as electrical components, there are often very few if any indications prior to catastrophic component failure. In some systems data such as usage pattern logging can be used in models that predict degradation and assist in condition-based maintenance. The specialist modeling and limited applicability of these systems are outside of the scope of this chapter.

At the systems or mission level, prognosis involves the ability to reason about the impact of the loss of low-level functionality on higher-level objectives. For example, the detection of a fuel leak may require a change of mission objective (abort, emergency landing, etc.), and the prognostic task is to determine available scenarios and select appropriate action. The prognostics and health management (PHM) area is responsible for developing the technology and systems needed to enable UAVs to monitor their own state. Based on that monitoring, the PHM systems need to make assessments of mission readiness that can be utilized by higher-level planning in the decision modeling area. These tasks are usually considered as the vehicle health management systems and are addressed in other chapters of this book. The input information to these prognostic tasks is the result of system or subsystem design and diagnosis, and this will be the focus of this chapter.

42.2 Model-Based Reasoning

Model-based systems and qualitative reasoning (MBS&QR) as a visible subfield of artificial intelligence can be traced back some 20 years to the publication of seminal collection of papers in the field (Bobrow 1984; Hamscher et al. 1992), although there was of course earlier work in this area (de Kleer 1977; Brown et al. 1982).

Model-based reasoning encapsulates a number of key concepts identified in (Price et al. 2006b; Peischl and Wotawa 2003) as follows:

- Separation of system model from problem solving. Model-based systems are based on a separation of the problem-solving algorithm from the model of the domain. Once a library of appropriate component models has been established, only a structural description of the respective device or system (e.g., obtained from design data) is required to automatically generate a system model and, based on it, problem-solving software dedicated to this device or system.
- Compositionality. Since systems are assembled from standard components and the behavior (and misbehavior in the case of a fault) of the system emerges from the behavior of these components, establishing a model library is feasible and entails collecting models of (correct and faulty) behavior of such standard components. This is important: this kind of model-based reasoning cannot be performed if the overall behavior of the system cannot be composed from the behaviour of the components and the way in which they are linked. Where there is compositionality of models, a high degree of reuse is possible, as well as prediction of what will happen in unexpected circumstances such as failure situations.

- Existence of different abstract problem solvers. The combination of generic domain independent problem solvers and compositional models of a domain can produce model-based systems that are able to reason efficiently about a product using relatively easy to obtain and objective design data.
- Modeling at different levels of abstraction. In diagnosis, for example, modeling different kinds of problems may well involve modeling different phenomena and at different levels of detail. However, while there may be a need for quantitative and or semiquantitative models, qualitative models provide a nice solution for representing phenomena at a higher level of abstraction.

Model-based reasoning (MBR) appears in standard AI texts (Luger 2002) and has been widely used in several domains for a variety of tasks as documented in the review by Price et al. (2006a). In particular, the efficiency and coverage provided by qualitative models have many advantages in the early stages of design and in the natural abstraction of explanations provided by such simulations and analysis. The efficiency of MBR is particularly suited to the task of failure modes and effects analysis due to the high level of system state coverage required combined with the large number of failure modes to be considered. An FMEA must present failure modes and associated risk together with the possible causes of the failure mode. The abstraction provided by QR (Kuipers 1986) makes it possible to identify failure modes and effects using language that is intuitive to an engineer. For example, an explanation that a *low-pressure* failure mode has the effect of *reduced engine output power or shutdown* and is caused by a *pipe blocked* fault may be derived directly from a qualitative model. Detailed numerical models are often difficult to obtain, particularly early in the design process when it is still feasible to change the design and sensing requirements. This consideration is of particular importance if the objective is to design diagnosis capability and mitigation into the system rather than retrofitting it to a completed design.

The major limitation of qualitative techniques is that in some circumstances the simulation can predict alternative behaviors. For example, a qualitative electrical circuit containing a bridge network may predict current flow in either direction across the bridge resistance because the flow depends on the ratio of the surrounding resistance values. Similarly, for systems that use a qualitative representation of time, there may be alternative behavior predictions dependent upon the ordering of events within the same qualitative time period. This chapter considers a modeling strategy that ensures qualitative ambiguity does not become a limiting factor and that it is related to physical system characteristics that have real behavioral significance. Given these constraints qualitative ambiguity actually provides useful additional information by identification of critical parameters within the system and can be addressed by adding additional constraints to the model (effectively making explicit a design specification) or appealing to selected numerical simulation for particular situations.

The following sections will consider how model-based reasoning can be used to automatically generate a comprehensive FMEA. This report contains a great deal of diagnostic information and may also be used to create workshop diagnosis or to automatically generate a diagnostic system.

42.3 Failure Modes and Effects Analysis

Manual FMEA is generally carried out by identification of failure modes and subsequent derivation of system effects and potential root causes, such as component failure, by an engineering team. Automated FMEA generally approaches the problem from the bottom up by performing system simulation for each potential fault and interpreting the resulting behaviors in terms of system functionality (Price 1998).

The main concept is that a description of the overall behavior of a system can be constructed by knowing the structure of a system and the behavior of each of its components. In order to determine the behavior of the system when a failure occurs, it is only necessary to replace the component that has failed with a version of the component that reproduces the faulty behavior, and the model of the system will then reproduce the effects of the failure for the whole system.

Figure 42.1 shows the different levels of reasoning needed to perform model-based simulation of a system containing electrical and/or fluid flow components. The system description is usually provided at the component level – the system is described in terms of its components and their states and connections. This can be mapped onto the resistive state of all components, and the state of the overall system can be determined qualitatively (CIRQ). Flow and effort variables are global in nature and cannot be derived by propagating local component input/output behavior. At the qualitative level, a generalized solver (Lee 1999) can be used to determine global current and voltage in the electrical domain or pressure and

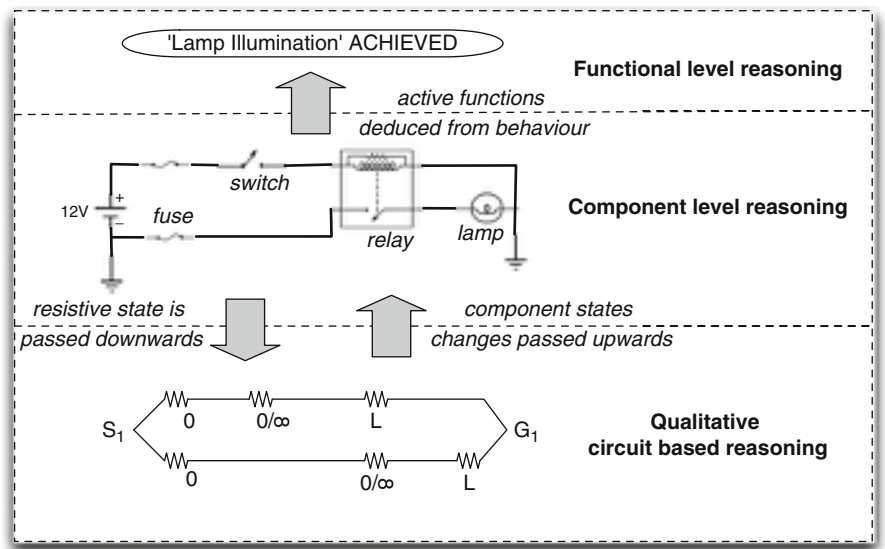


Fig. 42.1 Levels of electrical simulation

flow in a hydraulic or fluid transfer system. Alternative approaches such as bond graphs (Borutzky 2010; Mukherjee and Karmakar 1999) can also be used to provide similar information, although this may require more modeling effort and careful consideration of the causal flow through the circuit. For the purpose of FMEA, and the generation of rule-based diagnostics, and as a natural consequence of qualitative analysis, a state-based modeling approach is often adequate at the component level and used in preference to a complex dynamic numerical model for most cases. For states or subsystems where a qualitative or state-based model cannot provide the required behavioral detail or is unable to answer the required questions, numerical models can be used.

Changes to the effort and flow variables may result in changes in component state (e.g., if current is now flowing through a relay, then it may close, changing the state of the relay component, necessitating further qualitative simulation). Eventually, the state of the system will either stabilize or oscillate. Local component behavior such as the filling of a tank or the logic of an electronic control unit (ECU) is modeled at the component level often using finite state machines (FSM) (Harel and Politi 1998). The FSM representation is widely used to represent abstracted system or component behavior, being used to represent systems as diverse as communications protocols (SDL) and software behaviour (UML) (UML 2011). The results of the interchange between the component level reasoning and the qualitative electrical reasoning can be abstracted to the functional level and returned in terms of which functions of the system have occurred.

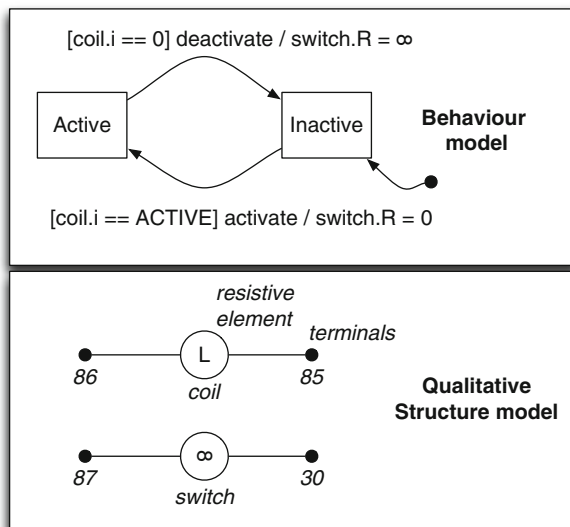
Figure 42.1 illustrates the relationship between the three levels for a very simple system, but these techniques work for electrical/electronic systems with several thousand components and produce useful FMEA results that pinpoint significant potential failures early in the design process.

To summarize the information required:

- Component models or a component library. Each model contains structural information such as a resistance network between the terminals of the component and also behavioral information such as a state chart that controls changes to the structural elements.
- A system schematic that connects individual component instances.
- A description of the system functions and which behaviors provide these functions.

A choice must be made as to the level of granularity required for the qualitative variables. A minimal but surprisingly powerful scheme for qualitative resistance is $\{0, \text{load}, \infty\}$. In this scheme, an open switch has ∞ resistance and a closed switch has zero resistance, and a power-consuming device is represented as a load. From this information the simulation can derive that a fractured wire will cause a motor to stop or that a bad connector (load) will lower the voltage across a motor. This is enough information to derive functional effects of an FMEA and for diagnosability analysis. There are situations where a finer-grained model is useful, for example, when dealing with event durations using a qualitative scale such as $\{\text{day}, \text{hour}, \text{second}, \text{mS}, \text{uS}\}$ is useful to avoid ambiguity. For example, if two components have state-changing behavior, the potential event ordering ambiguity between a tank

Fig. 42.2 Example simple relay component model



emptying and a relay switching is resolved. The tank model may specify that at nominal flow a partly filled tank will take “hours” to empty whereas the relay will switch in “mS.” Care must be taken with the semantics of such a scheme since qualitatively it is specifying that *any number* of sequential events in one qualitative timeslot will finish before any events in the next timeslot. Therefore, the qualitative model must not be too fine grained, and each level must represent a qualitatively significant and distinct region of behavior.

A major benefit of the qualitative modeling scheme is that the models are highly reusable since none of the numerical parameters are present. The model library will therefore contain a single model to represent a whole class of actual device part numbers. For example, a single model covers many “normally open single pole” relays. As an example Fig. 42.2 shows a structural and behavioral model for a basic relay. Such models are easy to produce and can be used in early stage design before detailed parameterization of the system is possible allowing early failure mode analysis and diagnosability characterization to be carried out.

The failure behaviors for each component failure mode are provided by a modification of the nominal component structure or behavior model. For example, a blocked pipe will simply change the resistance of the pipe to “infinite.” These faults are created for each type of component and are stored in the component library as part of the component description. Faults are then automatically inserted for each instance of the component present in a system. This section has described how nominal and failure behavior can be generated; however, an FMEA report is required to summarize the abnormal behavior of each failure in terms of the impact on the system functions. To automate this requires the system functions to be provided in a machine-readable form. The next section describes briefly

one approach to this known as the functional interpretation language (Bell et al. 2007) that was developed specifically for behavioral interpretation and abstraction.

42.4 Functional Description

Functional reasoning is a term that includes a number of knowledge representation techniques aimed at capturing the purpose or intent of an engineered system. Functional modeling has been in use for a number of years, and there are a variety of methods (see Far and Halim Elamy 2005; van Wie et al. 2005 for a comparative summary), both for deriving the behavior of a system from knowledge of its structure and component function and also for interpreting system behavior. Interpretation and verification tasks benefit particularly well from inclusion of relatively small amounts of functional information. Functional modeling is an important aspect of techniques such as MADe (PHM Technology) where a standardized taxonomy of functions is used together with a structured framework for modeling system functional dependencies.

MADe is used to model systems in order to identify and report on potential functional, monitoring, and maintenance issues in system design. This enables failure analysis to be conducted concurrently with system design using fuzzy cognitive mapping and bond graph simulations.

In the case of FMEA, a simple functional model is able to interpret many hundreds of qualitative variable values into the state of a set of system functions. One approach considers functions as having four states as shown in Table 42.1. The function state is determined from the state presence or absence of the function triggers and expected effects.

An example functional model fragment is shown in Fig. 42.3 for an aircraft fuel transfer function. The functions are linked to the simulation via the triggers and effects produced from the simulation. In this example “TVL.TR” are valve positions and “OC.WT” are fuel tanks used in the trigger and effect expressions, respectively. This functional interpretation language allows hierarchies of functions to be formally specified and relates the function to its purpose within a deployment environment (larger system) and identifies the behavior (triggers and effects) that determines the state of the function. In this example the behavior is linked to a qualitative simulation that returns values such as “FORWARD” and “REVERSE” for flow direction and “FUEL” or “AIR” as the substance present in pipes. This

Table 42.1 Function states

| Trigger | Effect | Function state |
|----------------|----------------|---------------------------------|
| False (absent) | False (absent) | Inoperative |
| False (absent) | True (present) | Spontaneous (unexpected effect) |
| True (present) | False (absent) | Failed |
| True (present) | True (present) | Achieved |


```

FUNCTION wing_transfer_to_left {
    ACHIEVES transfer_fuel_from_right_to_left_wing
    BY switch_on_transfer_pump AND set_transfer_valves_R_to_L
    TRIGGERS transfer_into_left_tank}

PURPOSE transfer_fuel_from_right_to_left_wing {
    DESCRIPTION 'transfer fuel from right wing tank to left wing tank'
    FAILURE_CONSEQUENCE 'cannot balance aircraft heavy on right'
    SEVERITY 5 DETECTABILITY 4}

TRIGGER TVL_TR_LH.position == 'crossover' AND TVL_TR_RH.position == 'normal'
    IMPLEMENTS set_transfer_valves_R_to_L

TRIGGER CP_TR.Control == 'on'
    IMPLEMENTS switch_on_transfer_pump

EFFECT OC_WT_LH.transfer.FLOWDIRECTION == 'REVERSE'
    AND OC_WT_LH.transfer.FLOW == 'NORMAL'
    AND 'FUEL' IN OC_WT_LH.transfer_port.SUBSTANCE
    IMPLEMENTS transfer_into_left_tank
    UNEXPECTED_CONSEQUENCE
        'aircraft imbalance - fuel transferred to left wing tank when not expected'
    SEVERITY 6 DETECTABILITY 8

```

Fig. 42.3 FIL language-based functional model description fragment

functional language is tailored for FMEA and captures risk associated with function failure as severity and detectability using the interpretation required by the engineer.

42.4.1 Generating FMEA Results

FMEA unlike fault tree analysis (FTA) (Vesely 2011) normally considers the effects of all possible individual component faults. While it is possible to consider multiple faults in an automated FMEA analysis, the effort involved precludes this in a manual analysis. In an automated FMEA, the number of concurrent faults considered is limited only by the simulation time available since it grows exponentially with the number of faults. One approach is to limit multiple fault analysis to combinations of faults with combined component failure likelihood above some threshold. This will have the effect of considering only combinations of less reliable components. The increase in output is generally not as great as the increase in simulation time since many combined faults will produce effects no worse than the individual failures or the combined effects of individual failures and thus add nothing extra to the FMEA output. It is only combinations that lead to effects that are not part of the effects of the individual faults that require additional reporting.

An FMEA report can be automatically generated by comparing the results from a simulation of nominal system behavior with the results of a simulation of a version of the system for each possible component fault. The differences between the two simulations comprise the effect of the component failure (e.g., a function occurs at

some point in the simulation of nominal system behavior, but fails to occur in the same state when there is a specific component failure). More details of this technique are given in Price et al. (2006a).

The system needs to be simulated in its different operating modes in order to identify the possible effects of a failure. That means that it is necessary to decide how the system should be exercised during simulation in order to cover all operating modes. Normally an engineer will provide a set of inputs that cause the system to enter all of its operating modes and configurations. For example, in an aircraft fuel system such as the example in Fig. 42.4, the valve positions may be changed to encounter normal fuel feed, fuel transfer operations, and fuel cross feed configurations. The set of inputs chosen is referred to as the scenario. On occasion an engineer may decide to only deal with a subset of system operation, and this will limit the effects that will be observed to those observable during the selected operations.

For simplicity the fuel supply system fragment in Fig. 42.5 will be used to illustrate the FMEA generation. A fuel tank is used to supply an engine via several pipes using a single pump. A flow sensor, a pressure sensor, and an isolation valve are present in addition to a three-way valve that is not used in this system fragment.

The FMEA generation system operates in the following way. The system is simulated for the chosen scenario with no failures present, and all observable values are recorded for each step of the simulation. The system model is reconfigured for each component failure. Each step of a failure simulation is compared with the non-failure simulation and the differences are recorded.

The automatically generated FMEA provides a description of the observable effects of each fault. These effects are consistent, providing all potential effects in qualitative terms. Figure 42.6 shows a fragment of an FMEA generated from the example in Fig. 42.5. Due to the qualitative nature of the analysis, the results are naturally presented as qualitative differences (such as value higher than expected).

The FMEA results are ordered by significance and are inspected by engineers to identify how significant failures can be mitigated, either by decreasing the effect of the failure (e.g., by providing backup systems) or improving the detection of the failure (e.g., by adding sensors) or reducing the occurrence of the failure (e.g., by using more robust components). Risk priority numbers (RPN) can be generated for each failure item if required for a criticality analysis and comprise the product of severity, detectability, and occurrence. The likelihood of the failure is stored as part of the component failure model, and the detectability and severity are associated with the failure of function and are therefore part of the functional model as can be seen in Fig. 42.3.

The FMEA report provides a consistent report that covers all component failures of specific types, and this consistency and coverage provides a basis for generating diagnostics that will be explored in the next section.

The selection of the scenario is an area where the engineer can have an influence on the FMEA. The aim is to exercise all of the system functions that are of interest for the faults to be considered. If the behavior associated with a function is not exercised, any fault in the components that are involved in producing the behavior

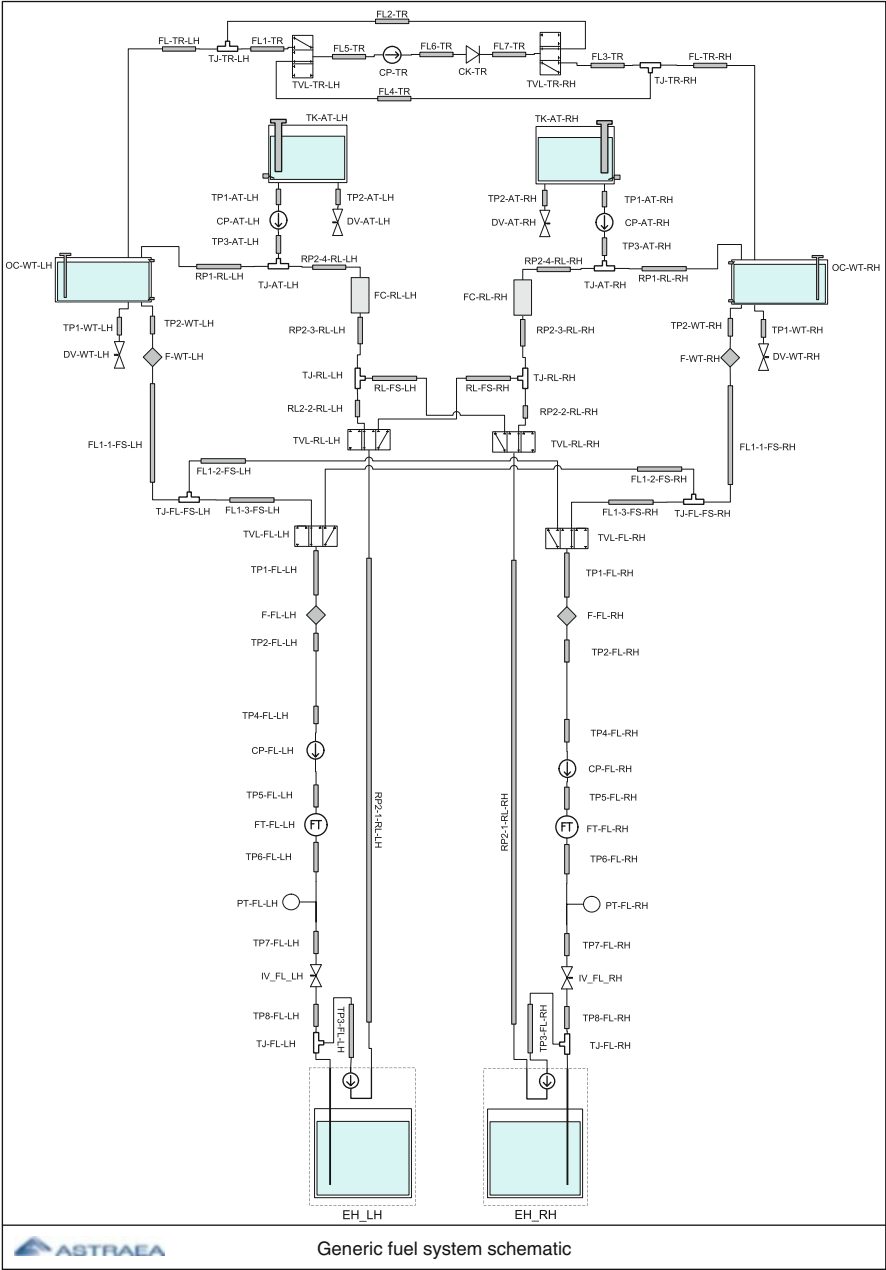


Fig. 42.4 Fuel system example

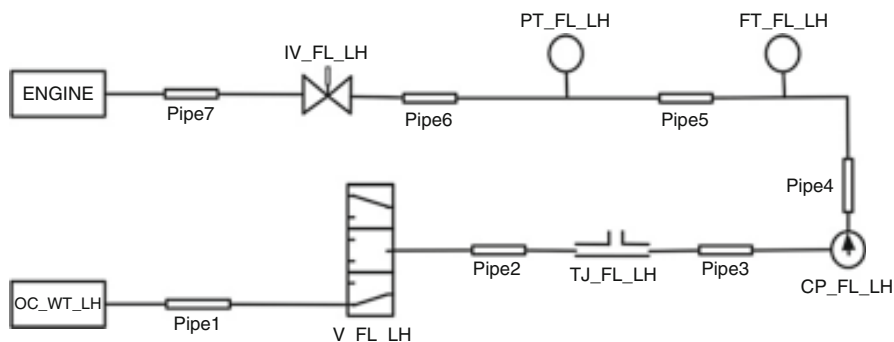


Fig. 42.5 Example fuel system fragment

will not produce effects, and the report will not be able to include that component fault as a cause of the function failing. For most systems it is only necessary to exercise each function individually since the worst effects of component faults are already present. It is possible that specific combinations of function activations or function operating modes could cause faults to have an effect that is worse (higher risk function failures) than the individual function activations, and in these cases the relevant function states need to be included in the FMEA. Including additional function combinations in the scenario has no disadvantage other than increased execution time, since the results will only need to include those that produce additional effects. One strategy sometimes used is to exercise individual functions and then “all functions” together if this is possible or largest physically possible subset if not.

Systems that contain backup and/or redundant functions need to be identified in the functional model with the relevant triggering conditions and the FMEA and functional model configured accordingly; the FMEA will then note that the backup function has been activated when a relevant fault occurs. A selective multiple fault analyses would be needed to determine the effects of faults in both the primary and nonredundant systems. Functional models such as (Bell 2006) identify fundamental meta-function types such as warning/telltale, fault mitigating, interlocking, and recharging, providing strong indications that the scenario and components involving these functions need particular consideration.

42.5 Sensor Selection and Diagnosability Analysis

Diagnosis is a fertile application area for artificial intelligence methodologies. Techniques range from rule-based expert systems and case-based reasoning to black-box approaches such as neural networks and have been applied to a wide variety of applications, for example, Travé-Massuyès and Milne (1997). This section will focus on model-based techniques because unlike black-box techniques

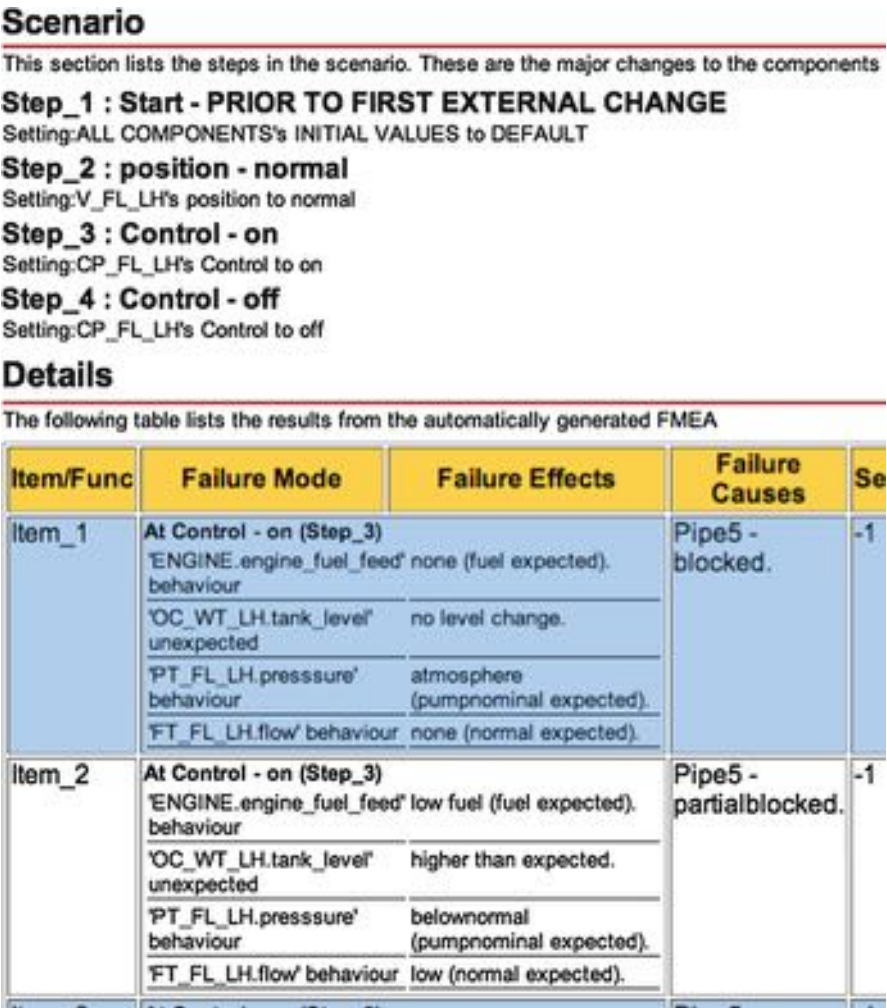


Fig. 42.6 FMEA fragment, for example, system

they can explain how a result was obtained and unlike rule-based expert systems or case-based techniques, model-based method do not require large amounts of historical data, training sets, or rules to be obtained from engineers. The traditional consistency-based diagnosis approach is outlined in Fig. 42.7 and is flexible and generic with widely used tools available (Forbus 1990; Forbus and de Kleer 1993; Kuipers 1994; Struss and Price 2003); however, these approaches do require that the model executed in real time, probably onboard the aircraft, to enable the discrepancy in actual and predicted behavior to be used to detect faults.

This approach does not necessarily require fault models for components: however, this can lead to the possibility of the production of physically impossible

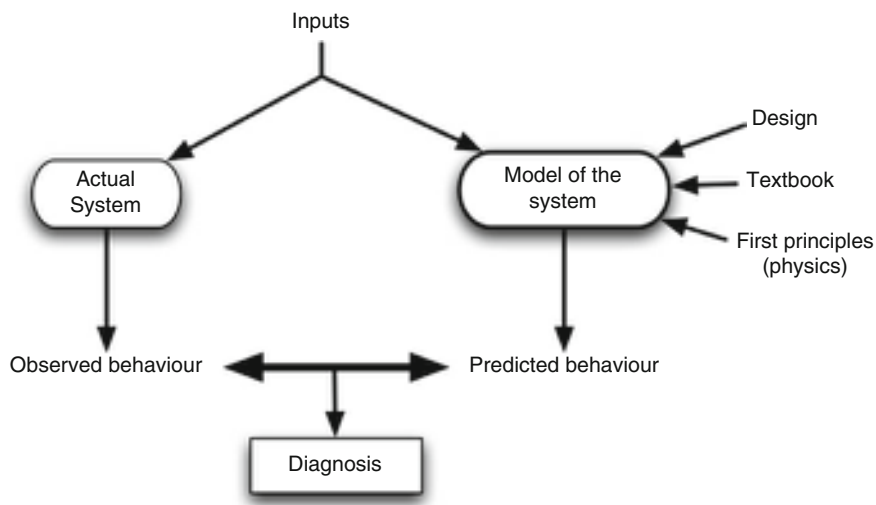


Fig. 42.7 Model-based diagnosis

diagnoses; for example, the classic example of three lamps in parallel with a battery, in the case where two lamps fail as well as a correct diagnoses, the possibility that the battery has failed and the lamp has failed (produces light when no power is applied) is produced. The kinds of problems stem from only having models of correct behavior, since the diagnostic system has no knowledge of how a component can fail. Qualitative models significantly reduce the amount of work required to specify fault models of components with most faults being modeled as simple changes to structural or behavioral elements. Automated FMEA relies on having fault models available for all component failure modes of interest.

Onboard model execution and certification concerns have led to a different approach to model-based diagnosis based on automated FMEA results that uses off-line models and simulation to produce sets of diagnostic rules that can be used as the basis of an onboard system. The rule set can then be used to perform diagnosability analysis. Automated FMEA data provides a great deal of diagnostic data. A comprehensive set of possible component faults is included, and the resulting effects for wide coverage of the system functional states will be available. Furthermore, and unlike a manually generated FMEA, an automated FMEA is produced by simulation and will provide accurate, consistent, and detailed effects for each fault. The results can therefore be processed automatically to produce diagnostic rules (symptoms) for each fault. In addition, the automated FMEA has been validated by an engineer providing higher confidence that the failure modes and causes are reasonable.

Once a set of diagnostic rules is produced, it can be tuned and combined with other techniques such as Bayesian networks (BN) (Ben-Gal 2007) to produce an efficient onboard system that is able to calculate the probability a fault will exist given a set of symptoms and thus provide ranked sets of potential faults. A BN (Fig. 42.8) is able to accommodate an incomplete set of diagnostic rules

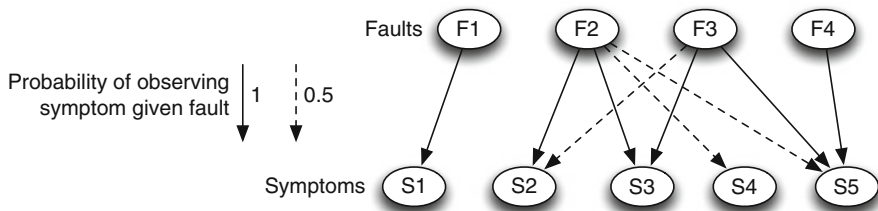


Fig. 42.8 Bayesian diagnostic network

or uncertainty in the observations, thus making the system less sensitive to the numerical thresholds set for qualitative values and also to the states produced the FMEA scenario. Due to its statistical nature, a BN can operate with whatever symptom and fault data is available, for example, symptoms produced by hand from manual FMEA or symptoms that are incomplete or contain contradictory evidence. Other than thresholds, the network uses three basic numerical measures:

Symptom leak – probability that the symptom will be observed even though there are no faults. By default this is set to 0 for symptoms generated from an automated FMEA symptoms to indicate the symptom will not be observed unless a relevant fault exists. It may be decided that certain observations are more likely to be spurious than others due to sensor or system characteristics.

Prior – probability that the fault has occurred prior to any symptoms. This is the fault occurrence number in FMEA (or 0.01 default)

Fault – probability that the symptom will be observed given the fault. By default this is set to 1 for the auto FMEA symptoms to indicate the symptom will be observed if the fault exists. Other values may be used for some symptom or fault categories because the effects predicted by the FMEA are less or more significant than can be determined qualitatively. For example, leaks may not always produce a *measurable* pressure drop, even though theoretically any leak would produce some drop in pressure. Engineering experience can also be captured and added into the model over time. For example, if a valve was found to make a characteristic sound (which can be detected), then this symptom can be added and conditional probabilities added based on how often the sound was reported for each fault.

When one or more symptoms are observed, these values are propagated through the network to produce an ordered list of possible faults. A confidence value is associated with each predicted fault allowing decisions to be made within the higher-level control elements of the system.

42.5.1 Generating Symptoms from an FMEA

The automated FMEA contains a lot of diagnostic information in the form of component fault – effect relationships. In addition these relationships are validated

once an engineer has considered the FMEA. For convenience the symptoms are categorized into three types, although algorithms that generate the symptoms generate whichever type is necessary based on the number of observations needed and to ensure symptoms are able to exonerate faults if required:

- *Single value*, for example, *pressure transducer value = out of range low*. In this situation a symptom is a simple value that does not occur during normal operation and directly indicates one or more faults.
- *Multiple value*, for example, *level = empty AND low level switch = on*. Two values may be linked to indicate an inconsistency in system operation caused by a fault. Typically, this is because two values measure physically related properties and should be consistent in the absence of a fault.
- *Conditional IF* (*pump = on*), *pressure monitor = high* or *IF (Valve SOV4 commanded OPEN), Open Sensor is not responding OPEN*. Conditional symptoms may be used when a value is only indicative of a fault in specific operating modes or configuration. For example, a low-pressure reading might only be significant when a pumping operation is in progress. If no pumping is carried out, then the symptom does not apply (pressure will be expected to be low). The condition specifies if the observation is valid. For the diagnostic net (discussed in following paragraph), the distinction between multiple value and conditional symptoms is important because the symptom observations are not entered to the net for an invalid symptom.

During diagnosis symptoms with a satisfied condition are valid, and valid symptoms may either indicate a fault if the overall symptom is satisfied or exonerate a fault if the overall symptom is false. Invalid symptoms are not applicable in the state or operating mode being considered.

The FMEA report provides differences of observable values, for example, *when PIPE5 is blocked, PT_FL_LH pressure is below normal when normal was expected*. This type of information can be used to produce the kind of fault-symptom information needed by the online diagnostic system. The symptoms should be as simple as possible while still providing a definite indication of the associated fault. The FMEA produces output in the form of the difference between nominal and failure operation at each step of a scenario, exercising the system. An example may be *After switch X closed and switch Y closed, Obs A=L when Obs 1=H expected*. The FMEA information does not directly provide a symptom because it is usually not known explicitly what is expected. The symptoms must therefore be based only on identification of abnormal observations. The FMEA item above plus additional simulation data generated during the FMEA production is used to produce a symptom of the form: *condition = {(X, closed), (Y, closed), (Obs B, Active)}; observation = {(Obs B, L)}; faults = {Pipe1 Blocked, Pipe2 blocked}*. To produce symptoms, additional nominal observations are often required such as the value of *Obs B* in the example.

Valid symptoms must satisfy two constraints. Firstly, a symptom should only be present when the associated fault is present. Secondly, it is desirable to detect as many faults as possible in as many operating states as possible. The first constraint requires that any potential symptom be identified such that it is not satisfied in any

nominal operating mode; however, it can be satisfied for other faults if general symptoms are required. There are efficient set theory algorithms for performing this kind of search; one specific example used for this application can be found in Snooke and Price (2012).

The second constraint implies symptoms should be general and therefore contain the fewest measurements, such that the first constraint is satisfied. Simply using symptoms that include every available observable value produced from the component failure simulations will create a very limited diagnostic system because faults will only be diagnosable in the states contained in the FMEA scenario. The simulations carried out to produce an automated FMEA provide a *subset* of the possible failure and nominal system states. To produce symptoms that diagnose faults over the majority of system states requires generalization of the effects of faults. Such generalization is achieved by reducing the number of observations in the symptom, in particular by excluding those that are not relevant to the fault being diagnosed.

42.5.2 Associating Measurement to Functions

The functional model (also used to interpret the FMEA) is the enabling concept that indirectly provides abstract-guiding knowledge-facilitating extrapolation of symptoms to system states that were not present in the FMEA. Measurements are associated with system functions based on the achievement and failure of functions across the entire FMEA. Given a comprehensive set of failures, this provides a simulation-derived description of which functions each component/measurement affects and is used to indicate a measurement has some behavioral or structural “causal” relationship to a function based on evidence of failure effect from the FMEA. Often these will simply be different value measurements from the same sensor; however, this is not assumed. By associating these abnormal measurements with functions that simultaneously fail, a mapping of function-associated measurements is created. Even measurements structurally adjacent to external triggers will be affected by faults in the connecting components. For example, *switch.position == on* is a trigger; however, the current flow through the *switch.contact* will be affected by faults such as the contact being stuck or wiring faults in the switch circuit and will be associated with any functions that are affected by any of the switches failure modes. This produces a set of measurements that agree with engineering expectation of all the components used within the implementation each function. However, even if this is not clear, there must be a structural or behavioral relationship between the function and associated measurements, as required for measurement selection.

A measurement-function mapping in turn allows an assessment of the relevance of measurements associated with a fault; measurements associated with functions that are never affected during the exercising of the fault (which should include at least all individual system functions) can be considered irrelevant and are removed from the set of measurements to be used as candidates for any symptom that will

diagnose the fault. This prevents symptoms being generated that rely on spurious correlations between measurements and faults due to the incomplete coverage of the system state space during the FMEA.

42.5.3 Fault Exoneration

Diagnostic symptoms may be required to be negated to perform fault exoneration – a symptom expression evaluating to false needs to indicate fault absence. For example, consider a lamp with a plausible symptom:

lamp == inactive AND switch == on implies faults {lamp blown, switch dirty, lamp wire fractured}

If the switch is off, then *switch == on* is false resulting in a false symptom expression, but this does not imply the lamp is OK – it could be blown. Manually crafted symptoms are often conditional so that symptom absence will exonerate associated faults. This is because engineers consider the implication of not seeing the symptom as well as its presence. In some applications such as a BN-based diagnostic system may require that symptoms exonerate faults. This also allows better fault detection by allowing evidence from nominally operating parts of a system to contradict evidence from very general symptoms, hence reducing the set of possible faults. That is, a symptom expression must not exonerate a fault for an observation where the fault exists and to achieve this requires additional checks in the symptom generation algorithm. Fault exoneration requires that the algorithms include additional features to ensure that observations are included in the conditional part of the symptom as necessary to ensure the symptom is only valid in operating states such that the symptom can be negated.

42.5.4 FMEA Coverage and Symptom Generation

The coverage provided by the FMEA can have an influence on the selection measurement strategy required to generate symptoms. For example, consider Fig. 42.9. Assume the functions A and B were not exercised simultaneously during the FMEA. There will be an entry in the FMEA for fault *w2 fracture* when the switch *sw1* is closed providing the effect *function A failed (function B is inoperative)*. One of the symptoms generated from this simulation requires the state of *w1* and *sw1*. Wire *w1* is allowed because it is associated with *Function A* (as are all the other wires on the path through *sw1* and *l1*). The symptom generation search algorithm determines that state of the switch is included in the conditional part of the symptom because if the switch is not closed, electrical activity in *w1* can say nothing about the fault in *w2* (it depends on the state of *Function B*). The symptom correctly predicts the fault *w2 fracture* when *function A* only is active but falsely exonerates the fault when both functions are triggered because *w1* is active when the condition is true but the symptom is false. Using a BN the probabilities could be adjusted to reflect this;

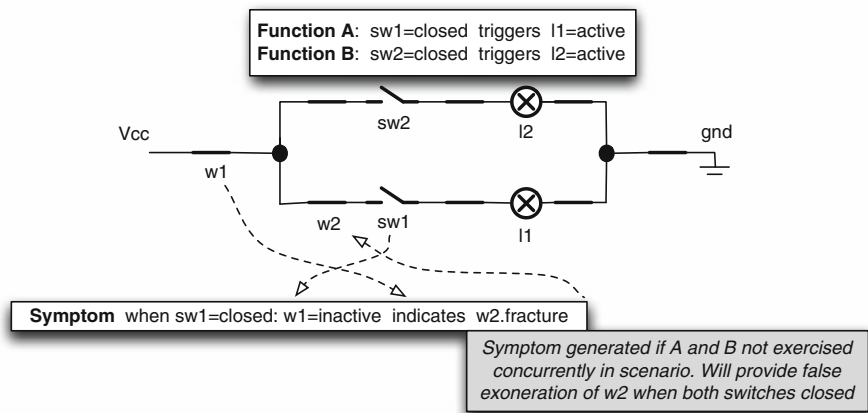


Fig. 42.9 Exoneration of faults

however, more accurate consideration of the operating modes of the system will lead to more accurate diagnoses (higher confidences in the predicted fault). One approach to achieving this is to exercise both of the functions simultaneously in the FMEA scenario, forcing the symptom generation to include additional measurements in the symptom, for example, the position of switch *sw2*. Thus by extending the scenario to close both switches and exercise both functions simultaneously will produce the symptom *when sw1 closed and sw2 open: w1 inactive indicates w2 fracture* to be generated since A is failed and B is active, allowing *w1* to be used.

There may be many possible permutations of functions and particularly function states available making exercising of all in the FMEA an unattractive proposition. In addition it is necessary to detect any spurious symptoms, although this can be done by further simulation. An alternative is to disallow measurements associated with several functions if any of the functions are inactive or unexpected. For the *w2* fault, this will remove *w1* as a candidate measurement for the symptom since *w1* is associated with the inoperative *function B*. Thus, *w1* will not be used in any symptoms generated for Fig. 42.9 unless both switches are closed in the FMEA scenario. The symptom *when sw1 closed: I1 inactive indicates w2 fracture* and similar for the other wires on that branch will be generated. Notice that both these revised symptoms apply to more specific operating conditions than the original and are only valid (usable) when function B is not activated. The symptom can only exonerate *w2* when function B is not activated (*sw 2* open) and function A is activated (*sw1* closed) and *w1* is not inactive. Measurements of *w1* are less useful than measurements of *I1* in diagnosing *w2 fractured*.

The measurement selection technique described addresses the incomplete set of low-level operating states inherent in the FMEA and ensures that irrelevant measurements are not included in symptoms, and hence that symptoms are extrapolated

Fig. 42.10 Example symptoms

| Symptoms | | |
|----------|---|------|
| Id | Description | Leak |
| S1 | PT_FL_LH is "atmosphere" | 0.0 |
| S2 | ENGINE.engine_fuel_feed is "low fuel" | 0.0 |
| S3 | OC_WT_LH.tank_level is "higher than expected" | 0.0 |
| S4 | PT_FL_LH is "belownormal" | 0.0 |
| S5 | FT_FL_LH is "low" | 0.0 |
| S6 | OC_WT_LH.tank_level is "lower than expected" | 0.0 |
| S7 | FT_FL_LH is "high" | 0.0 |
| S8 | ENGINE.engine_fuel_feed is "air" | 0.0 |
| S9 | PT_FL_LH is "pumpnominalabovenormal" | 0.0 |
| S10 | PT_FL_LH is "ambiguous(short)" | 0.0 |
| S11 | Condition: CP_FL_LH is "on" FT_FL_LH is "zero" | 0.0 |
| S12 | Condition: CP_FL_LH is "on" OC_WT_LH.tank_level is "no level change" | 0.0 |
| S13 | Condition: CP_FL_LH is "on" ENGINE.engine_fuel_feed is "zero" | 0.0 |
| S14 | Condition: CP_FL_LH is "on" PT_FL_LH is "zero" | 0.0 |

to be as general as possible. Conversely the decision as to which measurements are relevant is derived from the entire set of failure and non-failure states encountered during the FMEA. Given a reasonably comprehensive FMEA that exercises all system functions and includes faults for the majority of components, symptoms can be produced that are general enough to cover a great deal of the system state that is not explicitly included in the FMEA without producing spurious symptoms (Snooke and Price 2012). Analysis of the actual coverage of the derived symptom set is covered in a later section on diagnosability assessment. Since a diagnostic system based on the symptoms will use positive and negative evidence of each failure to derive a final likelihood of each fault, a “complete” set of symptoms is not necessary, and probabilistic techniques such as BN allow for a degree of uncertainty in the symptoms.

Figure 42.10 shows an example set of symptoms generated from the simple example system shown in Fig. 42.5. Most of these symptoms are single abnormal values; however, notice that several conditional symptoms have been generated (S11–S14) to allow values that occur when the system is inactive to become symptoms when the operating conditions change. For more complex systems, the combinations of values and conditions required become greater and provide symptoms that detect unusual effects and are particularly useful when a system has many operating modes with symptoms specific to each.

Once the set of symptoms have been generated, the FMEA is used to produce a matrix of fault-symptom pairs. An example fragment is shown in Fig. 42.11,

| | | | | |
|-----|----------------|-----|---|------|
| F23 | Pipe1 leak | S8 | ENGINE.engine_fuel_feed is "air" | 0.99 |
| F24 | Pipe6 blocked | S9 | PT_FL_LH is "pumpnominalabovenormal" | 0.99 |
| F25 | Pipe7 blocked | S9 | PT_FL_LH is "pumpnominalabovenormal" | 0.99 |
| F12 | Pipe6 fracture | S10 | PT_FL_LH is "ambiguous(short)" | 0.99 |
| F14 | Pipe7 fracture | S10 | PT_FL_LH is "ambiguous(short)" | 0.99 |
| F1 | Pipe5 blocked | S11 | Condition: CP_FL_LH is "on" FT_FL_LH is "zero" | 0.99 |
| | | | Condition: CP_FL_LH is "on" | |

Fig. 42.11 Fragment of an example fault-symptom mapping

example fault-symptom mapping, showing fault number, fault name, symptom number, symptom name, and (default) confidence value. In a real system, there are likely to be one or more faults associated with a symptom, and each fault may be associated with one or more symptoms.

42.6 Diagnosability Assessment

Ideally, any system fault could be diagnosed to the individual component that has failed; however, during design, there is a trade-off between the amount of sensing possible for a system and the diagnostic capability. It is useful to be able to investigate the relationship between the number and placement of sensors and the resultant ability to detect faults and subsequently isolate faults to a specific component or line replaceable unit (LRU). One of the benefits of model-based simulation is easy access to many system parameters, allowing analysis of those that might provide the required diagnosability at a given cost. Alternatively the question might be reversed to analyze how many sensors would be required to be able to provide a given level of diagnosis. On the ASTRAEA project, two types of analysis were performed; the first provides an ordered list of sensors that can provide the maximum number of diagnosed faults, and the second provides an ordered list prioritized by the ability to isolate a fault. In practice a whole variety of compromises must be made, and these “what if” analyses are used to inform higher-level engineering decisions.

Symptom ranking is carried out using a recursive procedure starting with no included symptoms or detected faults and consists of the following steps:

1. From the remaining symptoms that have not been considered, find the symptom(s) that provide the maximum number of additional faults, that is, the number of faults detected by the symptom that are not already located by previously considered symptoms. These are termed the “next best symptoms”
2. Add each of the “next best symptoms” in turn to the included symptoms and also add any failures it detects to the faults detected list.
3. If the symptom has not already been found and included in the overall results, then include it in the results in the correct place (according to the number of

symptoms) and carry out the procedure again excluding the current symptom from the available symptoms.

This will generate sets of symptoms for each number of observations that can diagnose the most faults. For further information and examples of the software tools used, see (Snooke 2009; Snooke and Price 2012). Where several symptoms each diagnose the same number of faults, they are all outputted. For the example circuit in Fig. 42.1, the following summary is produced:

```
Ordered symptom information
1 combinations of 1 SYMPTOMS indicate 11 FAILURES (1 partitions)
1 combinations of 2 SYMPTOMS indicate 18 FAILURES (2 partitions)
1 combinations of 3 SYMPTOMS indicate 24 FAILURES (3 partitions)
1 combinations of 4 SYMPTOMS indicate 28 FAILURES (5 partitions)
8 combinations of 5 SYMPTOMS indicate 28 FAILURES (6--7 partitions)
28 combinations of 6 SYMPTOMS indicate 28 FAILURES (7--9 partitions)
56 combinations of 7 SYMPTOMS indicate 28 FAILURES (8--10 partitions)
70 combinations of 8 SYMPTOMS indicate 28 FAILURES (8--11 partitions)
56 combinations of 9 SYMPTOMS indicate 28 FAILURES (9--11 partitions)
28 combinations of 10 SYMPTOMS indicate 28 FAILURES (10--12 partitions)
8 combinations of 11 SYMPTOMS indicate 28 FAILURES (11--12 partitions)
```

The first of these provides:

```
ENGINE.engine{\_}fuel{\_}feed = none AND CP{\_}FL{\_}LH.Control = on
INDICATES FAULTS:
Pipe5.blocked; Pipe5.fracture; Pipe2.blocked; Pipe3.blocked; Pipe1.
blocked;
Pipe6.blocked; Pipe6.fracture; Pipe7.blocked; Pipe7.fracture; Pipe4.
blocked;
Pipe4.fracture
```

The second entry considers the best pair of sensors:

```
ENGINE.engine{\_}fuel{\_}feed = none AND CP{\_}FL{\_}LH.Control = on
INDICATES FAULTS:
Pipe5.blocked; Pipe5.fracture; Pipe2.blocked; Pipe3.blocked; Pipe1.
blocked;
Pipe6.blocked; Pipe6.fracture; Pipe7.blocked; Pipe7.fracture; Pipe4.
blocked;
Pipe4.fracture

ENGINE.engine{\_}fuel{\_}feed = low fuel
OC{\_}WT{\_}LH.tank{\_}level = higher than expected
FT{\_}FL{\_}LH.flow = low
INDICATES FAULTS:
Pipe5.partialblocked; Pipe2.partialblocked; Pipe3.partialblocked;
Pipe1.partialblocked; Pipe6.partialblocked; Pipe7.partialblocked;
Pipe4.partialblocked
```

In this case, there are three symptoms that all indicate the same set of failures and any one could be used. Sometimes there may be several sets of (nonequivalent) symptoms able to discriminate the same number of faults. In the example above, this occurs for 5 symptoms, where 8 different combinations of symptoms indicate 28 faults (though not necessarily the same 28 faults). They may have different

fault-isolating capability dividing the faults into six or seven partitions provided by different combinations of the chosen symptom set. These partitions each contain a set of one or more faults that are indistinguishable using the selected set of observations used by the selected symptoms. From the above table, it is clear that 4 symptoms are adequate to identify all 28 faults present in the FMEA of the system:

```
ENGINE.engine{\_}fuel{\_}feed = none AND CP{\_}FL{\_}LH.Control = on
INDICATES FAULTS: Pipe5.blocked; Pipe2.blocked; Pipe3.blocked;
Pipe1.blocked; Pipe6.blocked; Pipe7.blocked; Pipe4.blocked

ENGINE.engine{\_}fuel{\_}feed = low fuel
OC{\_}WT{\_}LH.tank{\_}level = higher than expected
FT{\_}FL{\_}LH.flow = low
INDICATES FAULTS: Pipe5.partialblocked; Pipe2.partialblocked; Pipe3.
partialblocked; Pipe1.partialblocked; Pipe6.partialblocked;
Pipe7.partialblocked; Pipe4.partialblocked

ENGINE.engine{\_}fuel{\_}feed = air
INDICATES FAULTS:
Pipe2.fracture; Pipe2.leak; Pipe3.fracture; Pipe3.leak; Pipe1.
fracture;
Pipe1.leak

ENGINE.engine{\_}fuel{\_}feed = none AND CP{\_}FL{\_}LH.Control = on
OC{\_}WT{\_}LH.tank{\_}level = lower than expected
INDICATES FAULTS: Pipe5.fracture; Pipe6.fracture; Pipe7.fracture;
Pipe4.fracture
OC{\_}WT{\_}LH.tank{\_}level = lower than expected
INDICATES FAULTS: Pipe5.leak; Pipe6.leak; Pipe7.leak; Pipe4.leak
```

This analysis demonstrates that the fuel feed detected by the engine is the most important available diagnostic indicator, followed by an incorrect fuel level in the tank. The flow and pressure sensors are not even necessary to detect a full set of faults. However, the engineer could decide that the engine fuel feed is not a feasible observation by using knowledge outside the scope of the modeling and simulation; removing it and running the analysis again, the flow meter then becomes an important sensor.

It is also possible to maximize the fault isolation capability using a modified version of the symptom-ranking algorithm that maximizes the number of fault partitions instead of simply the number of faults. Selecting the most useful measurements leads to 7 symptoms that can identify all 28 faults and divides them into 10 separate sets of (indistinguishable) faults. Maximizing for fault isolation leads to 12 partitions of faults using various combinations of the following 12 symptoms:

```
(Pipe2.partialblocked Pipe3.partialblocked Pipe1.partialblocked
Pipe6.partialblocked Pipe7.partialblocked) (Pipe5.partialblocked
Pipe4.partialblocked) (Pipe4.leak) (Pipe5.leak Pipe6.leak Pipe7.leak)
(Pipe2.leak Pipe3.leak Pipe1.leak) (Pipe2.fracture Pipe3.fracture
Pipe1.fracture) (Pipe5.blocked Pipe4.blocked) (Pipe6.blocked Pipe7.
blocked)
(Pipe6.fracture Pipe7.fracture) (Pipe5.fracture) (Pipe2.blocked
Pipe3.blocked Pipe1.blocked) (Pipe4.fracture)
```

This in fact includes all the symptoms in Fig. 42.4 because S2, S3, and S5 turn out to be identical as mentioned above.

The above paragraphs assume that all faults are equally important in terms of diagnosis. In practice, there are several additional considerations. Often there is a limit to the granularity required by fault isolation due to the presence of LRUs. There is no requirement to be able to isolate a fault beyond a single LRU.

Observations may also fall into categories including basic system sensing that must be available for nominal operation and groups of additional observations that only make sense to provide as single units (sensors). Including these additional factors in the sensor selection allows further structuring of the symptoms, observations, and measures used to select symptom permutations.

42.7 Conclusion

This chapter is concerned with methods and technologies needed to be able to routinely fly UAVs safely in commercial airspace. The techniques described in this chapter contribute to this goal in several ways:

- They automate the generation of failure effects for an FMEA report, providing consistent results for a complete set of component failure modes. It can be guaranteed that results are produced for all component failure modes that are modeled.
- The generated FMEA results can be arranged as failure-symptom pairs with the same consistency and can be linked to specific symptoms that are observable by an online system.
- The failure-symptom pairs have been integrated into a larger diagnostic system, where other failure-symptom pairs will have been produced by other methods.
- Further analysis can indicate the most effective points within the system being diagnosed to place sensors, assisting in the decision of where sensors should be placed when designing the system.

A number of existing modeling and analysis techniques can be integrated to provide models and design analysis that allow diagnosability and diagnosis to be designed in rather than bolted on, and this can form the basis of a lightweight verifiable onboard diagnostic system. Both of these aspects are essential when comprehensive system health information is required to enable mission level objectives to be decided in an automated environment.

References

- Astraea (2009), <http://www.astraea.aero/>. Accessed 20 Nov 2011
- Bell, Interpretation of simulation for model based design analysis of engineered systems. Ph.D. Thesis, University of Wales Aberystwyth, 2006. <http://cadair.aber.ac.uk/dspace/handle/2160/177>
- J. Bell, N. Snooke, C.J. Price, A language for functional interpretation of model based simulation. *Adv. Eng. Inform.* **21**(4), 398–409 (2007)

- I. Ben-Gal, in *Encyclopedia of Statistics in Quality and Reliability*, ed. by F. Ruggeri, R.S Kennett, F.W Faltin (Wiley, Chichester, 2007). ISBN 978-0-470-01861-3
- D. Bobrow (ed.), Special issue on qualitative reasoning about physical systems. *Artif. Intell.* **24**, 1–5 (1984)
- W. Borutzky, *Bond Graph Methodology* (Springer, New York, 2010). ISBN 978-1-84882-881-0
- J.S. Brown, R. Burton, J. de Kleer, Pedagogical and knowledge engineering techniques in SOPHIE I, II and III, in *Intelligent Tutoring Systems*, ed. by D. Sleeman, J.S. Brown (Academic, New York, 1982), pp. 227–282
- J. de Kleer, Multiple representations of knowledge in a mechanics problem-solver, in *Proceedings of the IJCAI-77* (Morgan Kaufmann, Los Altos, 1977), pp. 299–304
- B.H. Far, A. Halim Elamy, Functional reasoning theories: problems and perspectives. *Artif. Intell. Eng. Des. Anal. Manuf.* **19**(2), 75–88 (2005). Publisher Cambridge University Press, New York. ISSN: 0890–0604
- K. Forbus, The qualitative process engine, in *Readings in Qualitative Reasoning About Physical Systems*, ed. by D. Weld, J. de Kleer (Morgan Kaufmann, San Mateo, 1990), pp. 220–235
- K.D. Forbus, J. de Kleer, *Building Problem Solvers* (MIT, Cambridge, 1993). ISBN 978-0-262-06157-5
- W.C. Hamscher, J. de Kleer, L. Console (eds.) *Readings in Model-Based Diagnosis*. (Morgan Kaufmann, San Mateo, 1992)
- D. Harel, M. Politi, *Modeling Reactive Systems with Statecharts: The STATEMATE Approach* (McGraw-Hill, New York, 1998). ISBN 978-0070262058. Out of print. Downloadable from <http://www.wisdom.weizmann.ac.il/~harel/books.html>
- B.J. Kuipers, Qualitative simulation. *Artif. Intell.* **29**, 289–338 (1986)
- B. Kuipers, *Qualitative Reasoning – Modelling and Simulation with Incomplete Knowledge* (MIT, Cambridge, 1994). ISBN 978-0-262-11190-4
- M.H. Lee, Qualitative circuit models in failure analysis reasoning. *Artif. Intell.* **111**, 239–276 (1999)
- G.F. Luger, *Artificial Intelligence – Structures and Strategies for Complex Problem Solving*, 4th edn. (Addison Wesley, Reading, 2002). ISBN 0-201-64866-0
- A. Mukherjee, R. Karmakar, *Modeling and Simulation of Engineering Systems Through Bond-graphs* (CRC Press LLC/N.W. Corporate, Boca Raton, 1999, 2000). ISBN 978-0849309823
- B. Peischl, F. Wotawa, Model-based diagnosis or reasoning from first principles. *IEEE Intell. Syst.* **18**, 32–37 (2003)
- PHM Technology, <http://www.phmtechnology.com/>. Accessed 18 Nov 2011
- C.J. Price, Function directed electrical design analysis. *Artif. Intell. Eng.* **12**(4), 445–456 (1998)
- C.J. Price, N.A. Snooke, S.D. Lewis, A layered approach to automated electrical safety analysis in automotive environments. *Comput. Ind.* **57**, 451–461 (2006a)
- C. Price, L. Travé-Massuyès, R. Milne, L. Ironi, K. Forbus, B. Bredeweg, M. Lee, P. Struss, N. Snooke, P. Lucas, M. Cavazza, G. Coghill, Qualitative futures. *Knowl. Eng. Rev.* **21**(4), 317–334 (2006b). Cambridge University Press
- N. Snooke, An automated failure modes and effects analysis based visual matrix approach to sensor selection and diagnosability assessment, in *Proceedings of the Prognostics and Health Management Conference (PHM09)*, San Diego, CA, Sept 2009
- N. Snooke, C.J. Price, Automated FMEA Based Diagnostic Symptom Generation, *Proc. Advanced Engineering Informatics*, AEI 2012, **26**, 870–888. doi 10.1016/j.aei.2012.07.001
- N. Snooke, C.J. Price, An effective practical model-based detectability tool, *Proceedings of the DX-11*, Murnau, Germany, 2011, pp. 180–187
- P. Struss, C.J. Price, Model-based systems in the automotive industry. *AI Mag.* **24**(4), 17–34 (2003)
- The Unified Modelling Language, <http://www.uml.org/>. Accessed 29 Nov 2011
- L. Travé-Massuyès, R. Milne, TIGERTM: gas turbines condition monitoring using qualitative model based diagnosis. *IEEE Expert Intel. Syst. Appl.* **12**(3), 21–31 (1997)
- M. van Wie, C.R. Bryant, M.R. Bohm, D.A. McAdams, R.B. Stone, A model of function-based representations. *Artif. Intell. Eng. Des. Anal. Manuf.* **19**, 89–111 (2005)
- W. Vesely, *Fault Tree Handbook with Aerospace Applications*, NASA, <http://www.hq.nasa.gov/office/codeq/doctree/ftbh.pdf>. Accessed 3 Sept 2011

Kai Goebel and Bhaskar Saha

Contents

43.1 Introduction 1054

43.1.1 Prognostics 1054

43.2 Modeling 1055

43.2.1 Nominal Behavior 1056

43.2.2 Damage Modeling 1056

43.3 Prognostics Algorithms 1057

43.3.1 Model-Based Algorithms 1057

43.3.2 Particle Filter 1057

43.3.3 Data-Driven Algorithms 1059

43.4 Case Study: Prognostics for Batteries Used in Electric UAV 1060

43.4.1 Battery Characteristics 1060

43.4.2 Model Adaptation 1062

43.4.3 Battery Model 1065

43.4.4 UAV Application 1065

43.4.5 Implementation Results 1067

43.5 Conclusion 1069

References 1069

Abstract

Health management plays an important role in operations of UAV. If there is equipment malfunction on critical components, safe operation of the UAV might possibly be compromised. A technology with particular promise in this arena is equipment prognostics. This technology provides a state assessment of the health

K. Goebel (✉)
NASA Ames Research Center, Moffett Field, CA, USA
e-mail: kai.goebel@nasa.gov

B. Saha
Palo Alto Research Center, Palo Alto, CA, USA
e-mail: bhaskar.saha@parc.com

of components of interest, and if a degraded state has been found, it estimates how long it will take before the equipment will reach a failure threshold, conditional on assumptions about future operating conditions and future environmental conditions. This chapter explores the technical underpinnings of how to perform prognostics and shows an implementation on the propulsion of an electric UAV. An accurate run-time battery life prediction algorithm is of critical importance to ensure the safe operation of the vehicle if one wants to maximize in-air time. Current reliability-based techniques turn out to be insufficient to manage the use of such batteries where loads vary frequently in uncertain environments. A Particle Filter is shown as the method of choice in performing state assessment and predicting future degradation. The method is then applied to the batteries that provide power to the propeller motors.

43.1 Introduction

Prognostics Health Management (PHM) is an engineering discipline that aims at maintaining nominal system behavior and function and assuring mission safety and effectiveness under off-nominal conditions. PHM encompasses a set of wide-ranging subdisciplines that address the design, development, operation, and life cycle management of subsystems, vehicles, and other operational systems. Starting with simple time-temperature recorder for the engine hot section on the F-8 aircraft (during deployment in Vietnam), and later the A-7 aircraft engine health monitoring program of the early 1980s, PHM principles found their way in various aircraft. Analysis of vibration and acoustic emissions data from rotorcraft drivetrains has led to breakthroughs in predicting impending failures of complex mechanical systems, resulting in the development of relative mature Health and Usage Monitoring Systems (HUMS) for rotorcraft (Revor and Bechhoefer 2004). Service providers like GE, PW, and Rolls-Royce have employed PHM principles to remotely monitor jet engines around the clock to detect early signs of damage as part of guaranteed uptime service agreements.

43.1.1 Prognostics

Prognostics is a core element of PHM. It is a younger member in the family of health management techniques but has recently received considerable attention due to its game-changing potential. Prognostics is the science of determining the remaining useful life of a component or subsystem given the current degree of wear or damage, the component's load history, and anticipated load and environmental conditions. A quantification of the degree of a component's wear or damage and the estimate of end of life gives decision makers important information about the health of a system. This information can be used on UAV for risk reduction in go/no-go decision, cost reduction through the scheduling of maintenance as needed, and improved asset availability. Prognostics employs techniques that are

often based on a detailed analysis of historical data or an analysis of the fault modes and the modeling of the physics of both the component itself and the attributes that characterize the fault. For the latter, the idea is to model the progression of damage which includes the effects of damage accelerators or stressors (such as load or environmental conditions). Next, algorithms that estimate the remaining life use estimation techniques that propagate the anticipated degradation into the future and provide as output the point where the component does no longer meet its desired functionality. The algorithms use these physics-based models as well as measurements from the system as input. Output of the algorithms is the remaining life estimate. An alternative to physics-based models are data-driven techniques. These synthesize a behavioral representation from a large number of example run-to-failure trajectories via machine learning methods.

Prognostics can be developed for almost any critical component as long as one has either some knowledge about the underlying physics or a sufficient amount of run-to-failure data exists. The efficacy of prognostics has been demonstrated for a wide range of diverse components ranging from mechanical components to electrochemical components to electronics. Specific application areas range from rotating machinery (Marble and Tow 2006) to batteries (Saha et al. 2007), from printed circuit boards (Gu et al. 2007) to solid rocket motors (Luchinsky et al. 2008). In the U.S. military, two significant weapon platforms were designed with a prognostics capability as an integral element of the overall system architecture: the Joint Strike Fighter Program (Hess et al. 2004) and the Future Combat Systems Program (Barton 2007). Prognostic technology is also finding its way into future NASA launch vehicles and spacecraft (Osipov et al. 2007) as well as UAV (Valenti et al. 2007). As the technology matures further, it is expected that prognostics will play an important role in the design and operation of commercial systems such as passenger aircraft, automobiles, ships, the energy infrastructure, and consumer electronics.

43.2 Modeling

Underlying any prediction is a model that describes how the component of interest behaves under nominal conditions and how it will evolve as it experiences wear or a fault condition. To that end, one needs to capture that process in the form of a mathematical model. The model may be derived from laws of physics, captured by empirical relations, or learned from data. Models can also use a combination of these approaches. For example, parts of a component that are well understood may be constructed using physics-based models, with unknown physical parameters learned from data using appropriate system identification techniques.

Modeling of physics can be accomplished at different levels, for example, micro- and macro-levels. At the microlevel, physical models are embodied by a set of dynamic equations that define relationships, at a given time or load cycle, between damage (or degradation) of a component and environmental and operational conditions under which the component is operated. The microlevel models are often referred to as damage propagation model, for example, Yu and Harris's fatigue life

model for ball bearings, which relates the fatigue life of a bearing to the induced stress (Yu and Harris 2001), Paris and Erdogan's crack growth model (Paris and Erdogan 1963) and stochastic defect-propagation model (Li et al. 2000) are other examples of microlevel models. Since measurements of critical damage properties (such as stress or strain of a mechanical component) are rarely available, sensed system parameters have to be used to infer the stress/strain values. Microlevel models need to account in the uncertainty management the assumptions and simplifications, which may pose significant limitations of that approach.

Macro-level models are characterized by a somewhat abridged representation that makes simplifying assumptions, thus reducing the complexity of the model (typically at the expense of accuracy). An example is a lumped parameter model which assumes that the attributes of the component have idealized behavior and the nonideal characteristics are characterized with equivalent elements that suffice for a first-order approximation. When such a system is designed well, it will often (but certainly not always) result in satisfactory results, depending on performance requirements. It should be noted that for complex systems (e.g., a gas turbine engine), even a macro-level model may be rather time-consuming and labor intensive. The resulting simplifications may need to be accounted for via explicit uncertainty management.

43.2.1 Nominal Behavior

The system model describes the characteristics of the system under nominal conditions. Ideally, such a model should be able to factor in the effects of operational and environmental conditions as well as any other conditions that cause different system response under nominal conditions. The model should also be able to adapt to changes of the system that are not considered abnormal. To that end, the system model could learn system behavior from examples, for instance, using machine learning techniques or it could integrate domain expertise and be implemented using rules.

43.2.2 Damage Modeling

A damage propagation model describes how the damage is expected to grow in the future. It should, similar to the system model, account for operational and environmental conditions as well as any other conditions that have an impact on the damage. While one often thinks of damage as a monotonically increasing phenomenon, it is possible for the domain in which damage is evaluated to have non-monotonic attributes. These could be either intrinsic attributes (e.g., recovery effects in batteries or power semiconductors) or extrinsic effects such as partial repair actions. Depending on the fault mode, damage propagation may exhibit different symptoms and it may be necessary to consider dedicated damage propagation models for different fault modes.

43.3 Prognostics Algorithms

The role of the prognostic algorithm is applying the damage propagation model into the future. It needs to ensure that it properly considers the effects of environmental and operational conditions, healing phenomena, as well as how to account for the different sources of uncertainty. Depending on the implementation, the damage propagation model and the prognostic algorithm may not be separable. For the general case, the damage propagation model and the prognostic algorithm will be treated as separate.

43.3.1 Model-Based Algorithms

In order to make a prediction, the current health state of the system must first be known. Determining the current state of the system health is generally known as health state estimation. Given the current state of health, the model-based prediction algorithm propagates this estimate forward in time using damage propagation model equations up to the end-of-life (EOL) threshold. As mentioned earlier, the evolution of the state depends on future stressors. Therefore, these needs to be stated to the degree possible. For many applications, some knowledge about future stressors exists where tasks are scheduled or repeated. In other cases, one has only statistical information about future stressors. The accuracy and precision of the predictions depend on both the quality of the model and the uncertainty of future inputs. The prediction algorithm of choice depends on the type of model used, on what information is needed to describe the EOL (which could be in the form of a distribution), and the amount of computation that may be performed.

This section takes a look at Particle Filters, one of the most prevalent algorithms for model-based prediction.

43.3.2 Particle Filter

Particle methods assume that the state equations can be modeled as a first-order Markov process with additive noise and conditionally independent outputs (Arulampalam et al. 2002). Let

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \omega_{k-1} \quad (43.1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + v_k. \quad (43.2)$$

While there are several flavors of Particle Filters, the focus here is on *Sampling Importance Resampling* (SIR), in which the posterior filtering distribution denoted as $p(\mathbf{x}_k|\mathbf{Z}_k)$ is approximated by a set of N weighted particles $\{\langle x_p^i, w_p^i \rangle; i = 1, \dots, N\}$ sampled from a distribution $q(\mathbf{x})$ that is “similar” to $\pi(\mathbf{x})$, i.e., $\pi(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$. The *importance weights* w_k^i are then normalized

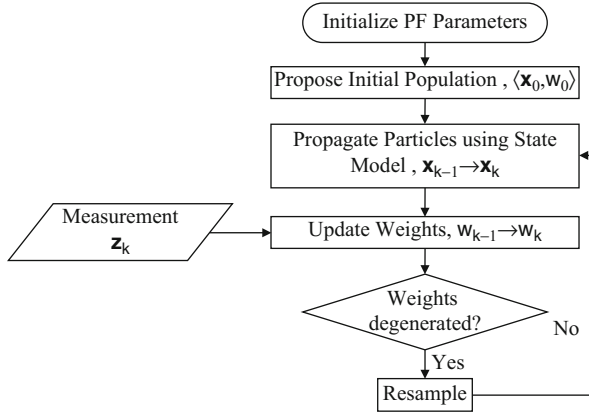


Fig. 43.1 Particle Filtering flowchart

$$w_k^i = \frac{\pi(\mathbf{x}_k^i) / q(\mathbf{x}_k^i)}{\sum_{j=1}^N \pi(\mathbf{x}_k^j) / q(\mathbf{x}_k^j)} \quad (43.3)$$

such that $\sum_i w_k^i = 1$, and the posterior distribution can be approximated as

$$p(\mathbf{x}_k | \mathbf{Z}_k) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i). \quad (43.4)$$

Using the model in Eq. (43.1), the prediction step becomes

$$p(\mathbf{x}_k | \mathbf{Z}_{k-1}) \approx \sum_{i=1}^N w_{k-1}^i \mathbf{f}_{k-1}(\mathbf{x}_{k-1}^i). \quad (43.5)$$

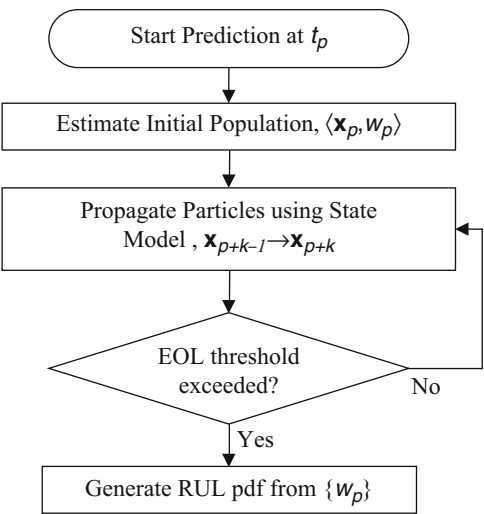
The weights are updated according to the relation

$$\bar{w}_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}, \quad (43.6)$$

$$w_k^i = \frac{\bar{w}_k^i}{\sum_{j=1}^N \bar{w}_k^j}. \quad (43.7)$$

It is possible that all but a few of the importance weights degenerate such that they are close to zero. In that case, one has a very poor representation of the system state (and also wastes computing resources on unimportant calculations). To address that, *resampling* of the weights can be used (Saha et al. 2007). The basic logical flowchart is shown in Fig. 43.1.

Fig. 43.2 Prediction flowchart



During prognosis this tracking routine is run until a long-term prediction is required, say at time t_p , at which point Eq.(43.1) will be used to propagate the posterior pdf given by $\left\{ \left\langle x_p^i, w_p^i \right\rangle; i = 1, \dots, N \right\}$ until \mathbf{x}^i fails to meet the system specifications at time t_{EOL}^i . The RUL pdf, i.e., the distribution $p(t_{\text{EOL}}^i - t_p)$, is given by the distribution of w_p^i . Figure 43.2 shows the flow diagram of the prediction process.

43.3.3 Data-Driven Algorithms

As the name implies, data-driven techniques utilize monitored operational data related to system health. Given the availability of data, data-driven approaches are appropriate when the understanding of first principles of system operation is not easy to come by or when the system is sufficiently complex that developing an accurate model is prohibitively expensive. The principal advantage of data-driven approaches is that they can often be deployed quicker and cheaper compared to other approaches and that they can provide system-wide coverage. On the other hand, data-driven approaches require a substantial amount of data for training which is a fundamental limitation for most systems, since full trajectories to failure are not recorded in large numbers for components in high-value systems like aircraft. Data-driven approaches can be further subcategorized into fleet-based statistics and sensor-based conditioning. In addition, data-driven techniques also subsume cycle-counting techniques that may include domain knowledge.

There are two basic data-driven strategies that involve either (1) modeling cumulative damage (or, equivalently, health) and then extrapolating out to a damage (or health) threshold or (2) learning directly from data the remaining useful life.

As mentioned, a principal bottleneck is the difficulty in obtaining run-to-failure data, in particular for new systems, since running systems to failure can be a lengthy and rather costly process. Even where data exist, the efficacy of data-driven approaches is not only dependent on the quantity but also on the quality of system operational data. These data sources may include temperature, pressure, oil debris, currents, voltages, power, vibration and acoustic signal, spectrometric data, as well as calibration and calorimetric data. Features must be extracted from noisy, high-dimensional data.

43.4 Case Study: Prognostics for Batteries Used in Electric UAV

This section will use batteries used for energy storage in electric UAV as an illustrative example to show the principles of prognostics. The section will start with a discussion on battery characteristics, followed by the model chosen and the implementation on the UAV.

43.4.1 Battery Characteristics

Batteries are essentially energy storage devices that facilitate the conversion, or *transduction*, of chemical energy into electrical energy, and vice versa (Huggins 2008). They consist of a pair of *electrodes* (*anode* and *cathode*) immersed in an *electrolyte* and sometimes physically divided by a *separator*. The chemical driving force across the cell is due to the difference in the chemical potentials of its two electrodes, which is determined by the difference between the *standard Gibbs free energies* the products of the reaction and the reactants. The theoretical *open-circuit voltage*, E° , of a battery is measured when all reactants are at 25 °C and at 1 M concentration or 1 atm pressure. However, the voltage during use differs from the theoretical voltage because of various passive components like the electrolyte, the separator, and terminal leads. The voltage drop due to these factors can be mainly categorized into Ohmic drop, activation polarization, and concentration polarization.

Ohmic drop ΔE_{IR} refers to the diffusion process where Li-ions migrate to the cathode via the electrolytic medium. The internal resistance to this ionic diffusion process is also referred to elsewhere as the *IR drop*. For a given load current, this drop usually decreases with time due to the increase in internal temperature that results in increased ion mobility.

Self-discharge is caused by the residual ionic and electronic flow through a cell even when there is no external current being drawn. The resulting drop in voltage has been modeled to represent the *activation polarization* of the battery ΔE_{AP} . All chemical reactions have a certain activation barrier that must be overcome in order

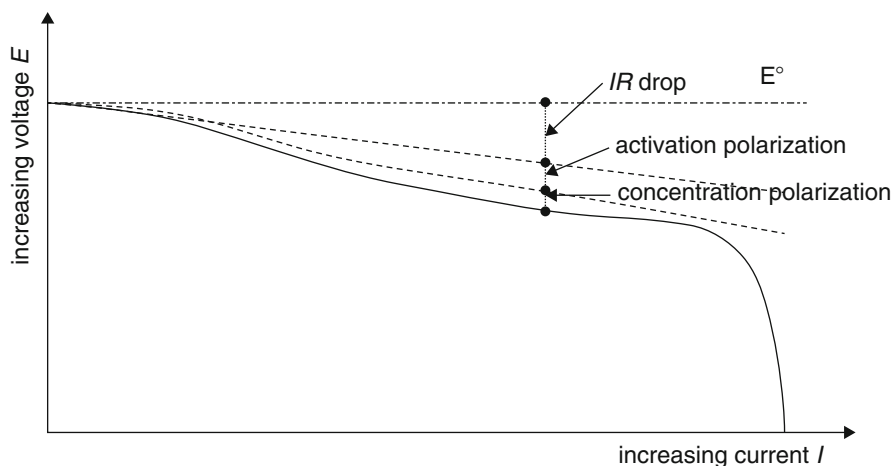


Fig. 43.3 Typical polarization curve of a battery (Saha and Goebel 2009)

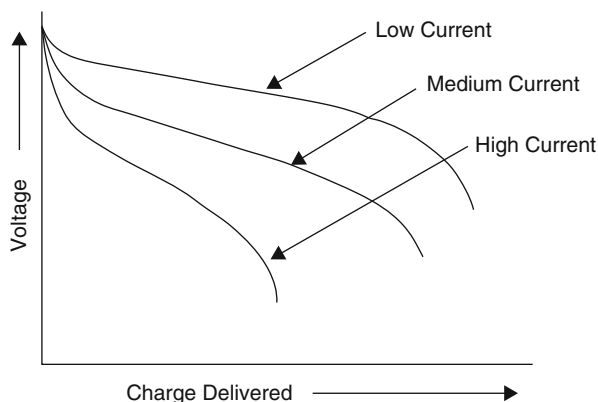
to proceed and the energy needed to overcome this barrier leads to the activation polarization voltage drop. The dynamics of this process is described by the Butler–Volmer equation (Bockris and Reddy 1973). This process was represented by an exponential function in Saha and Goebel (2009). However, a log function is a more accurate representation, as abstracted from the Butler–Volmer equation.

The *concentration polarization* ΔE_{CP} represents the voltage loss due to spatial variations in reactant concentration at the electrodes. This is mainly caused when the reactants are consumed by the electrochemical reaction faster than they can diffuse into the porous electrode, as well as due to variations in bulk flow composition. The consumption of Li-ions causes a drop in their concentration along the cell, which in turn causes a drop in the local potential near the cathode. The value of this factor is low during the initial part of the discharge cycle and grows rapidly towards the end of the discharge or when the load current increases.

Figure 43.3 depicts the typical polarization curve of a battery with the contributions of all three of the above factors shown as a function of the current drawn from the cell. The voltage drop usually increases with increasing output current.

The output current plays a prominent role in determining the losses inside a battery and is therefore an important parameter to consider when comparing battery performance. The term most often used to indicate the rate at which a battery is discharged is the *C-Rate* (Huggins 2008). The discharge rate of a battery is expressed as C/r , where r is the number of hours required to completely discharge its nominal capacity. So, a 5 Ah battery discharging at a rate of $C/10$ or 0.5 A would last for 10 h. The terminal voltage of a battery, as also the charge delivered, can vary appreciably with changes in the *C-Rate*. Furthermore, the amount of energy supplied, related to the area under the discharge curve, is also strongly *C-Rate* dependent. Figure 43.4 shows the typical discharge of a battery and its variation

Fig. 43.4 Schematic drawing showing the influence of the current density upon the discharge curve (Reproduced from Fig. 1.14 in Huggins 2008)



with C -Rate. Each curve corresponds to a different C -Rate or C/r value (the lower the r the higher the current) and assumes constant temperature conditions.

43.4.2 Model Adaptation

For many engineered systems, models for nominal operation are available, but damage propagation models like Arrhenius model or Paris' law are comparatively rare. Developing these models may require destructive testing for model validation which may not be possible in many cases. In some cases, testing may be done on subscale systems, but there may be difficulty in generalizing the models learned. Additionally, the parameter values of these models are often system specific and thus need to be relearned for every new application. The PF framework described above can help in these cases by adapting the prognostic/aging model in an online fashion.

One of the key motivating factors for using Particle Filters for prognostics is the ability to include model parameters as part of the state vector to be estimated. This allows *model adaptation* in conjunction with state tracking and, thus, produces a tuned model that can be used for long-term predictions.

Let system health x_k state evolution model \mathbf{f} and measurement model \mathbf{h} with known noise distributions ω and \mathbf{v} , respectively. Additionally, the parameter values of \mathbf{h} are assumed to be known (without lack of generality). The system health state is assumed to be one dimensional. Stationary or (better) non-stationary measurement models can be used to account for progressive degradation in sensors caused by corrosion, fatigue, wear, etc. The parameters of \mathbf{f} , denoted by $\alpha_k = \{\alpha_{j,k}; j = 1, \dots, n_f\}$ $n_f \in N$, are combined with x_k to give the state vector $\mathbf{x}_k = [x_k \ \alpha_k]^T$, where T represents the transpose of a vector or matrix. Equations (43.1) and (43.2) can then be rewritten as

$$x_k = \mathbf{f}(x_{k-1}, \alpha_{k-1}) + \omega_{k-1} \quad (43.8)$$

$$z_k = \mathbf{h}(x_k) + v_k. \quad (43.9)$$

To formulate the state equations for α_k , one can choose a *Gaussian random walk* such that

$$\alpha_{j,k} = \alpha_{j,k-1} + \omega_{j,k-1} \quad (43.10)$$

where $\omega_{j,k-1}$ is drawn from a normal distribution, $\mathcal{N}(0, \sigma_j^2)$, with zero mean and variance σ_j^2 . Given a suitable starting point $\alpha_{j,0}$, and variance σ_j^2 , the PF estimate will converge to the actual parameter value $\bar{\alpha}_j$, according to the *law of large numbers*. In this way, model adaptation has been introduced into the PF framework, adding n_f extra dimensions, yet achieving convergence (and without incurring the curse of dimensionality).

The notion of a good proposal density, though, comes into play in the choice of the values of $\alpha_{j,0}$ and σ_j^2 . If the initial estimate $\alpha_{j,0}$ is far from the actual value and variance σ_j^2 is small, then the filter may take a large number of steps to converge, if at all. The variance value may be chosen to be higher in order to cover more state space, but that can also delay convergence. One way to counter this is to make the noise variance itself a state variable that increases if the associated weight is lower than a preset threshold, i.e., the estimated parameter value is far from the true value, and vice versa. Thus,

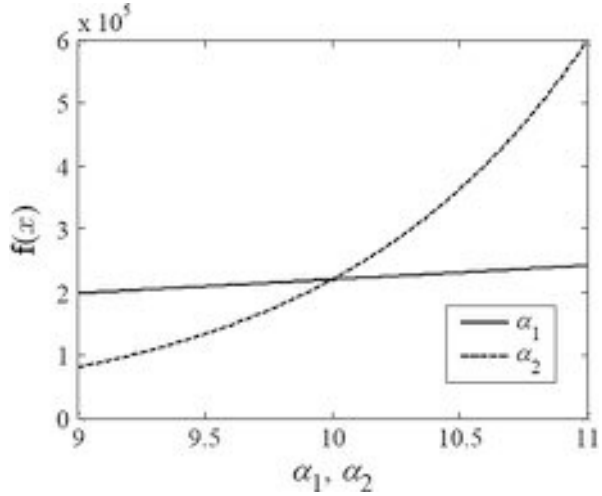
$$\begin{aligned} \alpha_{j,k} &= \alpha_{j,k-1} + \omega_{j,k-1}; \quad \omega_{j,k-1} \sim \mathcal{N}(0, \sigma_{j,k-1}^2), \\ \sigma_{j,k} &= c_{j,k} \cdot \sigma_{j,k-1}; \quad \begin{cases} c_{j,k} < 1, & \text{if } w_{k-1} > w_{\text{th}}, \\ c_{j,k} = 1, & \text{if } w_{k-1} = w_{\text{th}}, \\ c_{j,k} > 1, & \text{if } w_{k-1} < w_{\text{th}}. \end{cases} \end{aligned} \quad (43.11)$$

The multiplier $c_{j,k}$ is a positive valued real number, while the threshold w_{th} is some value in the interval $(0, 1)$. The intent is to increase the search space when the error is high and tightening the search when one is close to the target. Note that although this produces a better proposal density, it introduces a further n_f dimensions to the state vector.

It is quickly evident that it is not feasible to take this approach for all the parameters of a sufficiently high-order model. This motivates the use of sensitivity analysis techniques (SA) to determine the more sensitive parameters that need to be estimated online. SA is essentially a methodology for systematically changing parameters in a model to determine the effects on the model output. There are several methods to perform SA like local derivatives (Cacuci 2003), sampling (Helton et al. 2006), and Monte Carlo sampling (Saltelli et al. 2004). Depending upon the form of the system model, any of these methods may be used to assess which parameters to target.

Assuming that the model function \mathbf{f} in Eq.(43.8) is differentiable, i.e., $\partial \mathbf{f} / \partial a_j$ (the time index k dropped for the sake of generality), it can be computed at any

Fig. 43.5 Effect on $f(x) = \alpha_1 \exp(\alpha_2 x)$ due to 10 % variation in parameters α_1 and α_2 (Saha and Goebel 2011)



point in the state space defined by $\mathbf{x}_k = [x_k \ \alpha_k]^T$. If the partial derivative is positive, then the value of the function increases with an increase in the parameter value and vice versa.

The magnitude of the derivative indicates the degree to which the parameter affects the output of \mathbf{f} , as shown in Fig. 43.5. This directs the choice of the parameters to estimate online. If the *posterior error* given by

$$e_k^i = x_k^i - \sum_{i=1}^N w_k^i x_k^i \quad (43.12)$$

is positive, then the parameters that have a positive local partial derivative need to be reduced and those with a negative one need to be increased. The opposite holds true if the error is negative. The amount by which the parameters need to be reduced or increased also depends on the magnitude of the local partial derivative. The higher the magnitude, the smaller the steps needed in order to prevent instability while approaching the true value. This notion can be formalized in the following way (the particle index i has been dropped for the sake of generality):

$$\alpha_{j,k} = \alpha_{j,k-1} + C_{j,k} + \omega_{j,k-1}; \quad \omega_{j,k-1} \sim N(0, \sigma_j^2). \quad (43.13)$$

$$\begin{aligned} C_{j,k} &\propto -e_k, \\ &\propto \left. \frac{\partial \mathbf{f}}{\partial \alpha_{j,k}} \right|_{\mathbf{x}_k}, \\ &= -K \cdot \frac{e_k}{\left. \partial \mathbf{f} / \partial \alpha_{j,k} \right|_{\mathbf{x}_k}}. \end{aligned} \quad (43.14)$$

Note that in this model adaptation scenario, the noise variance parameter is not added to the state vector since the search process is directed and not random as discussed previously.

43.4.3 Battery Model

For the purposes of the electric UAV BHM, the battery model design space is explored at a high level of abstraction with respect to the underlying physics. To predict the end of discharge (EOD), it is required to model the SOC of the battery. For the empirical charge depletion model under consideration here, the output voltage $E(t_k)$ of the cell is expressed in terms of the effects of the changes in the internal parameters (Saha et al. 2011):

$$E(t_k) = E^o - \Delta E_{IR}(t_k) - \Delta E_{AP}(t_k) - \Delta E_{CP}(t_k) \quad (43.15)$$

where E^o is the Gibb's free energy of the cell, ΔE_{IR} is the Ohmic drop, ΔE_{AP} is the drop due to activation polarization, and ΔE_{CP} denotes the voltage drop due to concentration polarization. These individual effects are modeled as

$$\Delta E_{IR}(t_k) = \Delta I_k \alpha_6 (1 - \exp(-\alpha_7 (t_k - t_{\Delta I_k}))) - \alpha_1 t_k \quad (43.16)$$

$$\Delta E_{AP}(t_k) = \alpha_{2,k} \ln(1 + \alpha_{3,k} t_k) \quad (43.17)$$

$$\Delta E_{CP}(t_k) = \alpha_{4,k} \exp(\alpha_{5,k} I_k t_k) \quad (43.18)$$

where ΔI_k is the step change in current at time $t_{\Delta I_k}$ and the α 's represent the set of model parameters to be estimated.

The PF representation of the battery state for predicting EOD is given by

$$\alpha_{j,k} = \alpha_{j,k-1} + \omega_{j,k}, \quad j = 1, \dots, 7 \quad (43.19)$$

$$x_k = E(t_k) + \omega_k \quad (43.20)$$

$$\mathbf{x}_k = [x_k \ \alpha_{j,k}]^T, \quad j = 1, \dots, 7 \quad (43.21)$$

$$z_k = E(t_k) + v_k \quad (43.22)$$

where all but the parameters α_3 and α_5 are learnt from training data, while α_3 and α_5 are estimated by the PF online.

43.4.4 UAV Application

Electric aviation concepts are receiving increasing attention because of their potential benefits for emissions, fuel savings, and possibly noise reduction. While electric propulsion is used in concept manned aircraft, electric UAV is more common due

Fig. 43.6 Edge 540 UAV

to their lighter weight and lower loss consequences (compared to manned flight). However, like ground vehicles, battery-powered electric propulsion in aircraft suffer from uncertainties in estimating the remaining charge and hence most flight plans are highly conservative in nature. Different flight regimes like takeoff/landing and cruise have different power requirements, and a dead stick condition (battery shut off in flight) can have catastrophic consequences. To tackle this issue the battery health management algorithm was designed and implemented on an embedded hardware platform and integrated it into an UAV airframe to provide real-time onboard battery life predictions (Saha et al. 2012).

The particular UAV platform is a COTS 33 % scale model of the Zivko Edge 540T as shown in Fig. 43.6. The UAV is powered by dual tandem mounted electric outrunner motors capable of moving the aircraft up to 85 knots using a 26 in. propeller. The gas engine in the original specification was replaced by two electric outrunner motors which are mounted in tandem to power a single drive shaft. The motors are powered by a set of four Li-Poly rechargeable batteries, each rated at 6,000 mAh. The tandem motors are controlled by separate motor controllers (Quach et al. 2013).

A 12 channel JR radio system is used to control the airplane. The system communicates in the 2.4 GHz band using a proprietary DSM2 protocol. Control surfaces are manipulated by seven actuators.

The airplane is equipped with a number of sensors to collect structure, propulsion, and navigation. The health of the structure is monitored using a series of strain gauges and accelerometers. Navigation data consist of GPS location, ground speed, altitude, true heading, and magnetic heading. Power plant data (such as motor RPMs, currents, battery voltages, and temperatures) help to assess adequacy of the thrust from the motors and 26 in. propeller, the relevant parameters being.

Data from sensors are logged by two separate data systems instantiated in a combination of RCATS and PC104. The RCATS is a turn-key system with proprietary software that creates an ASCII log of connected sensors. It provides telemetry data to a laptop receiver which displays the data for callout to the pilot. It measures flight-related parameters such as motor RPM, motor temperature, airspeed, and z-axis acceleration at 10 Hz and interleaves GPS position and altitude data at 1 Hz. The PC104 stack consists of a CPU board, a DC/DC converter, an IO card, and a signal conditioning card for strain gauges. It runs MathWorks xPC Target



Fig. 43.7 The BHM system installed onboard

operating system. The data are acquired using a Simulink model that is compiled to an xPC target OS. It records strain, accelerometer, battery temperature, and motor current at 200 Hz. It also outputs a 0.5 Hz sine wave to the RCATS system for synchronizing data.

The BHM system is designed to be a relatively low cost analog-to-digital data acquisition system. The design philosophy behind the first BHM system is to use COTS solutions which would have a light weight compact footprint. Figure 43.7 shows the system installed onboard the Edge 540 airframe.

The BHM system itself utilizes several small COTS boards to convert TTL signal voltages into PC RS-232 signal voltages.

43.4.5 Implementation Results

Testing on the Edge 540 UAV platform was initially carried out with the airframe restrained on the ground. The propeller was run through various load regimes indicative of the intended flight profile (takeoff, climb, multiple cruise, turn, and glide segments, descent, and landing) by varying the propeller rotational speed. Figure 43.4 shows the voltages during a typical profile. It is desired to predict when the battery will run out of charge, i.e., the EOD event indicated by the end of the voltage plots after landing.

In order to evaluate the prognostic algorithm, multiple predictions were made at the time instants 13, 15, 17, and 19 min. It is not desired to completely discharge the batteries in flight since there needs to be some time for the UAV pilot to land the aircraft with some safety margin on the remaining battery life (Fig. 43.8).

Fig. 43.8 Predictions (vertical lines) during ground test (Saha et al. 2012)

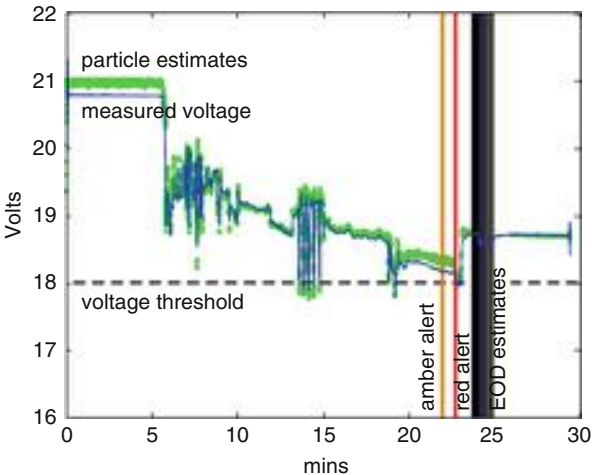
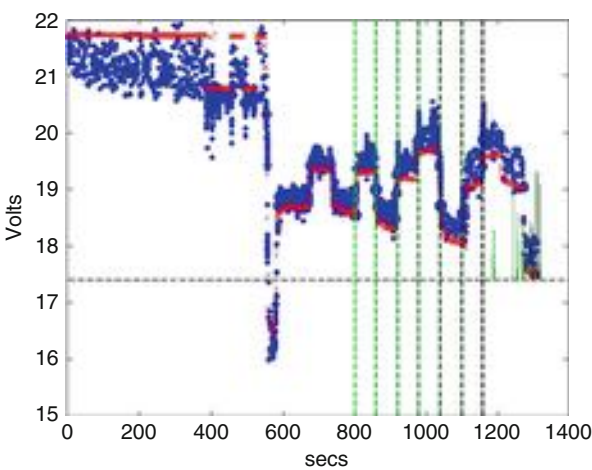


Fig. 43.9 Battery voltage prediction using Particle Filter during flight test (Saha et al. 2012)

In order to validate the learned prognostic model, several dozen actual flight tests were conducted using the UAV with randomized flight profiles. The prediction performance was accurate to within 2 min, i.e., $|t_{\text{EOD}} - \bar{t}_{\text{RUL},p}| < 2$ min, over multiple flights of durations between 15 and 25 min. Figure 43.9 shows the profile of one such flight.

The blue line indicates the measured voltage, while the green dots indicate the state values of the PF. The grey lines denote the $\mu_{\text{RUL},p}$ values plotted every second, while the amber and red lines represent early alerts for the pilot to land the plane before the dead stick condition.

43.5 Conclusion

This chapter lays out a technique for predicting component degradation as exemplified on a battery used for propulsion of an electric UAV. The approach chosen here is model-based where the form of the model has been linked to the internal processes of the battery and validated using experimental data. The model was used in a PF framework to make predictions of EOD. By profiling the power required for different flight regimes like cruise segments, banked turns, and landings, one can estimate the mission completion probability by calculating the RUL pdf. Since the prediction result is in the form of a pdf, it is easy to integrate the BHM routine into a higher-level decisioning algorithm that can provide advance warning about when to land the aircraft.

Generally, a similar process can be followed for other components where a damage progression model is built for the component at hand. Naturally, the model would be different, if, say, a bearing, an electronics component, or a structural element were the object of interest.

The next step in Prognostics Health Management is to integrate the health information, in this case remaining life, into a decision-making unit that reacts appropriately. Depending on the prognostic horizon, the reaction could be to autonomously change controller settings, reconfigure system resources to ensure primary mission goals (and perhaps extend the life of the stressed component), invoke a replanning or rescheduling routine, or provide the information for later processing in a maintenance setting.

Acknowledgments This work was performed as a cross-center collaboration between NASA Ames and Langley Research Centers (ARC and LaRC) and Dryden Flight Research Center (DFRC). The authors would like to especially thank Patrick Quach, Sixto L. Vazquez, Edward F. Hogge, Thomas H. Strom and Boyd L. Hill at LaRC, and Edwin Koshimoto at DFRC for their contributions. The funding for this work was provided by the NASA System-Wide Safety and Assurance Technologies (SSAT) project under the Aviation Safety Program of the Aeronautics Research Mission Directorate (ARMD).

References

- S. Arulampalam, S. Maskell, N.J. Gordon, T. Clapp, A tutorial on Particle Filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **50**(2), 174–188 (2002)
- P. Barton, Prognostics for combat systems of the future. *IEEE Instrum. Meas. Mag.* **10**, 10–14 (2007)
- O'M. Bockris, A.K.N. Reddy, *Modern Electrochemistry*, vol. 2 (Plenum, New York, 1973), pp. 845–1136
- J.O'M. Bockris, A.K.N. Reddy, A. Gamboa-Aldeco, *Modern Electrochemistry 2A: Fundamentals of Electrode Processes*, 2nd edn. (Kluwer Academic/Plenum Publishers, New York, 2000)
- D.G. Cacuci, *Sensitivity and Uncertainty Analysis: Theory*, vol. I (Chapman & Hall, Boca Raton, 2003)
- J. Gu, D. Barker, M. Pecht, Prognostics implementation of electronics under vibration loading. *Microelectron. Reliab.* **47**, 1849–1856 (2007)

- J.C. Helton, J.D. Johnson, J.C. Salaberry, J.B. Storlie, Survey of sampling based methods for uncertainty and sensitivity analysis. *Reliab. Eng. Syst. Saf.* **91**, 1175–1209 (2006)
- A. Hess, G. Calvello, T. Dabney, PHM a key enabler for the JSF autonomic logistics support concept, in: *Proceedings of the IEEE Aerospace Conference* Big Sky, 2004
- R. Huggins, *Advanced Batteries: Materials Science Aspects*, 1st edn. (Springer, New York/London, 2008)
- Y. Li, T. Kurfess, S. Liang, Stochastic prognostics for rolling element bearings. *Mech. Syst. Signal Process.* **14**(5), 747–762 (2000)
- D. Luchinsky, V. Osipov, V. Smelyanskiy, Model based IVHM system for the solid rocket booster, in: *Proceedings of the IEEE Aerospace Conference*, Big Sky, 2008
- S. Marble, D. Tow, Bearing health monitoring and life extension in satellite momentum/reaction wheels, in: *Proceedings of the IEEE Aerospace Conference*, Big Sky, 2006
- V. Osipov, D. Luchinsky, V. Smelyanskiy, In-flight failure decision and prognostics for the solid rocket booster, in: *AIAA 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, Cincinnati, 2007
- P. Paris, F. Erdogan, A critical analysis of crack propagation laws. *Trans. ASME J. Basic Eng.* **85**, 528–534 (1963)
- C.C. Quach, B. Bole, E. Hogge, S. Vazquez, M. Daigle, J. Celaya, A. Weber, K. Goebel, Battery charge depletion prediction on an electric aircraft, in *Proceedings of Annual Conference of the PHM Society (PHM'13)*, New Orleans, 2013
- M. Revor, E. Bechhoefer, Rotor track and balance cost benefit analysis and impact on operational availability, in: *Proceedings of the American Helicopter Society 60th Annual Forum*, Baltimore, 2004
- B. Saha, K. Goebel, Modeling Li-ion battery capacity depletion in a Particle Filtering framework, in: *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, San Diego, 2009
- B. Saha, K. Goebel, S. Poll, J. Christophersen, An integrated approach to battery health monitoring using Bayesian regression and state estimation, in: *Proceedings of the IEEE Autotestcon*, Baltimore, 2007, pp. 646–653
- B. Saha, K. Goebel, S. Poll, J. Christophersen, Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Trans. Instrum. Meas.* **58**(2), 291–296 (2009)
- B. Saha, C. Quach, K. Goebel, Exploring the model design space for battery health management, in: *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, Montreal, 2011
- B. Saha, P. Quach, K. Goebel, Optimizing battery life for electric UAVs using a Bayesian framework, in: *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, 2012
- A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models* (Wiley, Chichester/Hoboken, 2004)
- M. Valenti, B. Bethke, D. How, D. de Farias, J. Vian, Embedding health management into mission tasking for UAV teams, in: *Proceedings of the American Control Conference*, New York, 2007
- W.K. Yu, T. Harris, A new stress-based fatigue life model for ball bearings. *Tribol. Trans.* **44**, 11–18 (2001)

Guillaume Ducard

Contents

| | | |
|---------|---|------|
| 44.1 | Introduction | 1072 |
| 44.1.1 | Definition of Fault and Failure | 1072 |
| 44.1.2 | Different Approaches for FDI Systems | 1074 |
| 44.1.3 | Interaction Between Flight Controllers and FDI Systems | 1076 |
| 44.1.4 | Other Practical Challenges | 1077 |
| 44.2 | Aircraft Configuration and Dynamics | 1077 |
| 44.2.1 | Aircraft Configuration | 1078 |
| 44.2.2 | Aircraft Dynamics | 1078 |
| 44.3 | Residual Generator | 1079 |
| 44.3.1 | EKF Equations | 1080 |
| 44.3.2 | Filter's Equations | 1081 |
| 44.4 | Multiple Model Approaches for FDI Systems | 1084 |
| 44.4.1 | Modeling Actuator Faults | 1085 |
| 44.4.2 | The EMMAE Method | 1086 |
| 44.4.3 | Designing the EKF for the No-Fault Scenario | 1088 |
| 44.4.4 | Augmenting the State Vector with the Faulty Actuator Parameter δ_i | 1089 |
| 44.4.5 | Designing the EKF for the Case of a Failure on Aileron 1 | 1090 |
| 44.4.6 | Actuator Fault Isolation | 1090 |
| 44.4.7 | Simulation Results of the EMMAE-FDI with No Supervision System | 1095 |
| 44.4.8 | Remarks on the First Attempt to Use the EMMAE-FDI System | 1098 |
| 44.4.9 | Techniques to Improve Actuator Fault Diagnosis | 1098 |
| 44.4.10 | Computational Complexity of the EMMAE-FDI | 1101 |
| 44.4.11 | Realistic Flight Scenario and Conclusions | 1102 |
| 44.5 | A Single Model Active (SMAC) FDI System | 1103 |
| 44.5.1 | Residual Generator | 1103 |
| 44.5.2 | Fault Detector | 1106 |
| 44.5.3 | Excitation Signals Generator (ESG) | 1109 |

G. Ducard

I3S CNRS-UNS, Sophia Antipolis, France ETH Zurich, IDSC, Zurich, Switzerland

e-mail: ducard@i3s.unice.fr; ducard@idsc.mavt.ethz.ch

44.5.4 Actuator Health Evaluator (AHE) 1110

44.5.5 Fault Isolator..... 1112

44.5.6 Simulations 1114

44.5.7 Properties of the SMAC-FDI System..... 1117

44.5.8 Computational Load Requirements..... 1117

44.5.9 Conclusions About the SMAC-FDI System 1117

44.6 Chapter General Conclusion..... 1118

Appendix..... 1118

References 1119

Abstract

Future unmanned aerial vehicles (UAVs) will be designed to achieve their missions with increased efficiency, safety, and security. To this end, an efficient fault detection and isolation (FDI) system should be capable of monitoring the health status of the aircraft. Fault-tolerant control systems for small and low-cost UAVs should not increase significantly the number of actuators or sensors needed to achieve the safer operation. This chapter is dedicated to actuator fault detection systems for UAVs, with two main requirements: real-time capability and modularity. After defining the terminology employed in this field, this chapter reviews some commonly used techniques in FDI systems. The chapter continues by presenting briefly the mathematical model of a UAV which will serve as a basis for the design of two actuator FDI systems. The first method presents and illustrates the multiple-model approach, whereas the second method presents an FDI system which is based on a single model. Both methods have been enhanced by a mechanism that actively tests actuators in order to efficiently detect and isolate actuator faults and failures. This chapter explains the advantages and drawbacks of each method and discusses issues of robustness against model uncertainties and external perturbation. In addition, aspects of computational load are addressed. Finally, the FDI systems of this chapter are applied to a realistic model of an unmanned aircraft, and the performance of the methods is shown in simulation.

44.1 Introduction

44.1.1 Definition of Fault and Failure

According to the definition found in Isermann (2006), “a fault is an unpermitted deviation of at least one characteristic property or feature of the system from the acceptable, usual, standard condition.” Based on this definition, a fault corresponds to an abnormal behavior of the system, which may not affect the overall functioning of the system but may eventually lead to a failure (defined below). Moreover, a fault may be small or hidden and therefore difficult to detect and estimate.

For example, consider the temperature of an engine. If this temperature exceeds a certain accepted limit, say 100°C, there is a fault in the system. Although this

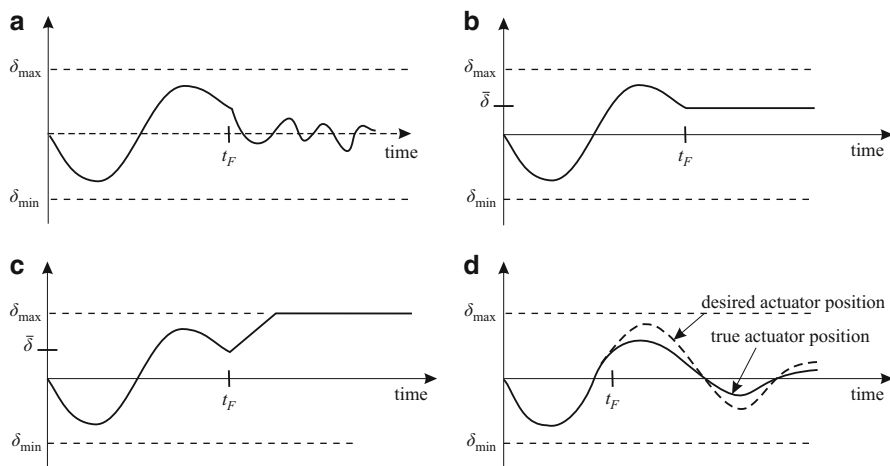


Fig. 44.1 Several types of actuator failures: (a) floating around trim, (b) locked-in-place, (c) hard-over, and (d) loss of effectiveness (actuator fault occurring after t_F)

excessive temperature does not prevent the engine from working properly for a while, it may eventually damage components of the engine and possibly lead to its breaking down.

In this chapter, an actuator fault corresponds to any abnormal behavior. This includes bias or loss of effectiveness as shown in Fig. 44.1d.

A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions. (Isermann 2006)

Resulting from one or more faults, a failure is therefore an event that terminates the functioning of a unit in the system. On an aircraft, actuators are used to deflect control surfaces such as ailerons, elevators, and rudders, and also to actuate the engine throttle or the landing-gear mechanism. An actuator is declared failed when it can no longer be used in a controlled manner.

For a control surface, there are two major types of failures (Boskovic and Mehra 2003). As shown in Fig. 44.1a, the control surface may become ineffective and float at the zero-moment position. The control surface can also be locked at any arbitrary intermediate position (Fig. 44.1b) or reach and stay at the saturation position as shown in Fig. 44.1c.

Mechanical failures may also happen. This is the case when the mechanical link between the control surface and its corresponding actuator or servo breaks. The engine may also fail.

Finally, there are other sources of possible irreversible damage to the aircraft that may be classified as *structural failures*. They correspond to the scenarios where a piece of the aircraft is missing, such as an aileron, a tail rudder, an elevator, or part of a wing.

44.1.2 Different Approaches for FDI Systems

In the context of reconfigurable flight control using an explicit online fault detection and isolation system (FDI), results from the FDI are used to modify the control system's structure, laws, parameters, or even the trajectory of the aircraft.

There are two families of FDI systems, namely, passive and active FDI systems. Passive FDI systems “wait” until a fault or failure occurs (Maybeck and Stevens 1991; Maybeck 1999), whereas active FDI systems will artificially excite the aircraft, either by flying health-check maneuvers (Azam et al. 2005; Elgersma et al. 1998) or by injecting test signals in the actuator commands and then assessing the individual health status of actuators and sensors.

Very few papers have discussed this technique so far. The work published by Honeywell in 1998 (Elgersma et al. 1998) and 2001 (Elgersma and Glavaski 2001) is among the first occurrences of using artificial exciting signals for FDI purposes. Test signals are injected into the null space of the inputs using redundant control surfaces such that these signals (ideally) cancel one another and thereby do not excite aircraft motion (Elgersma and Glavaski 2001) but contribute to better fault diagnosis; see also Buffington et al. (1999).

The use of artificial signals was demonstrated to improve significantly the performance of an FDI system based on the extended multiple model adaptive estimation (EMMAE) method (Ducard and Geering 2006, 2008). This method is described in Chap. 4 of this book (Ducard 2009). In 2007, the authors of Boskovic et al. (2007) suggested an adaptive fault-tolerant controller with self fault-diagnosis actuators. This is done by generating high-frequency signals for actuators with suspected failures and minimizing the effects of those signals on the system state using the remaining healthy actuators. In Bateman et al. (2011), the isolation procedure between redundant actuators is inspired from the excitation mechanism from Ducard and Geering (2006). Latest contributions in active FDI are reported in Ducard and Geering (2010) and extended in the second part of this chapter.

Table 44.1 provides a list of common and recent techniques that are encountered in the literature for the design of FDI systems.

44.1.2.1 Trends in Filter Design for FDI System

In the mid-1990s, several implementations of recursive least-squares (RLS) algorithms were used in FDI systems and successfully flight tested. For example, the work by Ward et al. (1998) describes a computationally efficient real-time parameter identification and reconfigurable control algorithm. The identification algorithm is based on a modified sequential least-squares (MSLS) found in Ward and Barron (1995), the recursive version of which is found in Bodson (1995). The MSLS parameter identification algorithm is based on RLS techniques and incorporates additional constraints to take into account a priori information and to adjust the size of the data window used in the regressor of the filter (Ward et al. 1994).

Many FDI filters have also been designed using mathematical models of the system being monitored. Model-based FDI methods have been enhanced using

Table 44.1 List of some recent and popular techniques used to design FDI systems for flight applications

| Technique | Example of recent books/papers using this technique (ordered chronologically) |
|--|--|
| (Modified -) recursive least-squares (RLS) | Ward et al. (1994, 1998), Bodson (1995), and Shore and Bodson (2005) |
| Kalman filters (KF) (bank of -) | Urnes et al. (1990), Maybeck and Stevens (1991), Eide and Maybeck (1996), Maybeck (1999), Ni (2001), Hajiyeve and Caliskan (2003), and Fekri et al. (2006) |
| Extended Kalman filters (EKF) (bank of -) | Hajiyeve and Caliskan (2003), Tanaka et al. (2006), Ducard and Geering (2006, 2008), and Ducard (2007) |
| Unscented Kalman filters (UKF) | Julier et al. (2000), Brunke and Campbell (2002), Campbell et al. (2007), and Perea and Elosegui (2008) |
| Linear parameter varying (LPV) filters | Szaszi et al. (2005) |
| Interaction matrix | Koh et al. (2005) |
| Particle filters | Rapoport and Oshman (2005) |
| Neural networks | Calise et al. (1998), Younghwan (1998), Wise et al. (1999), Brinker and Wise (2000), and Azam et al. (2005) |
| Statistical methods | Isermann (2006) and Samara et al. (2008) |
| Wavelet analysis | Adams (1994) and Azam et al. (2005) |
| H_∞ | Collins and Song (2000), Marcos et al. (2005), and Rotstein et al. (2006) |
| Robust model-based system | Chen and Patton (1999) and Patton et al. (2000, 2008) |
| Parity space approach | Chow and Willsky (1984), Frank (1994), Gertler (1997), Chen and Patton (1999), and Isermann (2006) |
| Unknown input observer | Patton et al. (1989), Chen et al. (1996), and Patton and Chen (1997) |

robust FDI techniques as defined in Chen and Patton (1999). It consists of incorporating during the design of FDI systems the effects of disturbance signals, model uncertainties, and measurement noise (Patton et al. 2000). It is often the case that several model-based filters are organized in a bank in which one filter is sensitive to a specified failure but the other filters remain insensitive to that failure. A recent example of this technique can be found in Patton et al. (2008), where a robust fault diagnosis for a spacecraft attitude control system is designed.

Many different variants of Kalman filters (KFs) have been constructed for detecting and isolating faults or for state estimation and state reconstruction. The use of extended Kalman filters (EKFs) applied to nonlinear systems for FDI purposes has also gained recent interest in Ducard and Geering (2008). A recent paper presented a method that uses EKFs to estimate online the aircraft's aerodynamic parameters and the components of the wind velocity. These estimates are used to update the parameters of the flight controller (Tanaka et al. 2006).

The unscented Kalman filter (UKF) is among the latest extensions of Kalman-type filters and seems to provide remarkable results for systems that are particularly nonlinear. The paper by Campbell et al. (2007) discusses the implementation of a sigma point filter (SPF) which was originally introduced as the UKF (Julier et al. 2000), where the distributions are approximated by a finite set of points. It is used to estimate aircraft states and aerodynamic derivatives in real time. This is a nonlinear estimation algorithm that can be performed online, which possesses robustness properties against parameter uncertainties, against filter tuning, and against initial conditions.

The discussion in Julier et al. (2000) explains that the SPF has similar performance to a truncated second-order EKF but without the need to calculate the Jacobian matrices. A comparison between EKF and SPF can also be found in Brunke and Campbell (2002). The main results of this chapter indicate that the SPF filter has equal or better performance than an EKF for real-time estimation application for the following reasons: the SPF is more robust against initial uncertainties and against jumps in the data, is less sensitive to tuning of the process noise, is less susceptible to divergence, is more accurate from one time step to the next, and, finally, requires equivalent computational load. Recent contributions in Perea and Eloegui (2008) focused on a new formulation for the state update equation of the filter for improved accuracy.

Recently, linear parameter-varying (LPV) filters gained the attention of some researchers in the fault-tolerant control community. For example, a design of LPV-based FDI filters is found in Szaszi et al. (2005). An example of an H_∞ control law that minimizes command tracking errors under actuator fault occurrence combined with an FDI filter based on an affine LPV model of a Boeing 747 is found in Shin et al. (2006).

44.1.2.2 Challenges of Designing Reliable FDI Systems

A reliable FDI system provides accurate information about the health status of the aircraft. In order to achieve such a result, the FDI system needs to be robust against external disturbances, model uncertainties, and sensor noise. In addition, the FDI system should not trigger false alarms and should still be sufficiently sensitive to detect the faults. Therefore, robustness is a fundamental issue in the performance of FDI systems and reconfigurable flight controller. FDI systems may experience significant performance reduction if the model uncertainties are not properly considered. A robustness analysis framework for failure detection and accommodation is provided in Chen and Patton (1999), Zhang and Jiang (2000), and Belcastro and Chang (2002).

44.1.3 Interaction Between Flight Controllers and FDI Systems

It is often the case that a reconfigurable flight control system incorporates an FDI system and a flight controller. The FDI system monitors the aircraft's behavior and identifies relevant parameters that are usually used by the flight controller to

synthesize the control commands. Therefore, the performance of the flight controller is dependent on the results provided by the FDI system and vice versa. Thus, the interactions between these two systems should be rigorously investigated.

The following observation is made in Shin et al. (2006): “it is fairly common for integration of failure detection and accommodation systems to be problematic if they are designed separately.”

Challenges exist when some aircraft parameters need to be identified during the flight in real time and under feedback control (Ward et al. 1998). This task is even more difficult and delicate when an actuator or a sensor fault happens. Moreover, the robustness of the flight controller can mask some aircraft faults and failures and make the detection problem more difficult.

There exist already many examples of integrated fault-tolerant control, sometimes referred to as IFTC (Zhang and Jiang 2000; Schierman et al. 2004; Boskovic et al. 2005; Shin et al. 2006).

New generations of reconfigurable flight control systems will not only rely on a fault-tolerant controller but will include complete and integrated systems that reconfigure the flight controllers, adapt the guidance system, and reshape online the vehicle trajectories (Schierman et al. 2004).

44.1.4 Other Practical Challenges

Usually, the flight control system relies on some nominal values for the mass, the moments of inertia, and the aerodynamic coefficients to generate the control signals. When the aircraft experiences an actuator failure or airframe damage, the aircraft becomes asymmetric. It is thus not trivial to determine which of these parameters need to be (online) reestimated to keep good flying performance.

It is often the case that the available onboard processing power is limited, in particular for small or micro UAVs. The design of a reconfigurable flight controller is therefore a trade-off between performance, complexity, and available processing power.

Finally, challenges are many when the fault diagnostic, the reconfiguration of the control system, and the reconfiguration of the path plan or of the mission are to be done autonomously under limited or no human supervision.

44.2 Aircraft Configuration and Dynamics

This section presents the mathematical model of an unmanned aircraft, whose actuators are to be monitored by an FDI system. This model has been constructed, validated, and identified through real flight tests of an aerobatic UAV and is reported in Möckli (2006). This model will serve as a basis for the design of the different filters created for the FDI systems of the subsequent sections. The numerical values of the aircraft model parameters are found in the Appendix.

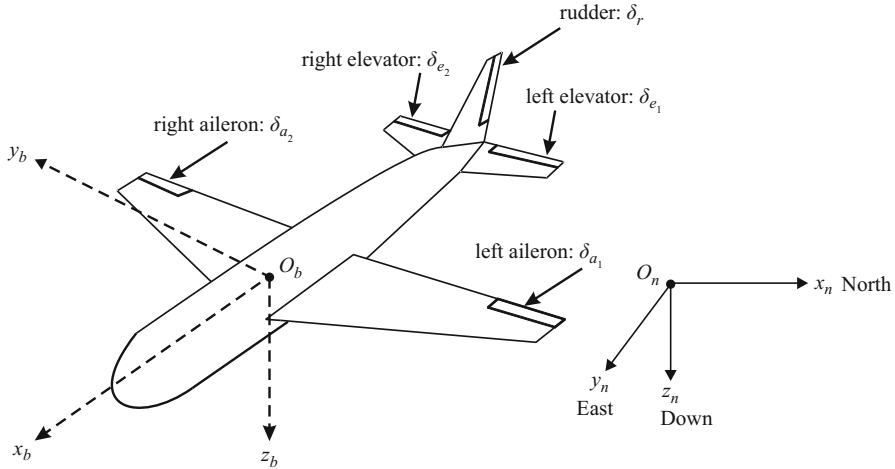


Fig. 44.2 Aircraft configuration

44.2.1 Aircraft Configuration

The five control surfaces of the aircraft under consideration are one left aileron, one right aileron, one left elevator, one right elevator, and one rudder, as shown in Fig. 44.2. All actuators are fully independent, which means that ailerons (or elevators) can individually move up or down. This configuration permits some pitch torque to be produced with ailerons or some roll torque to be produced with elevators. The control vector of the aircraft involving only actuator deflections is $\delta = [\delta_{a_1} \delta_{a_2} \delta_{e_1} \delta_{e_2} \delta_r]^T$.

44.2.2 Aircraft Dynamics

Among the nonlinear equations which describe the dynamics of an aircraft, those involving the turn rates are of primary interest. Indeed, as soon as any control surface exhibits a faulty behavior, the aerodynamic moments applied to the airframe are altered. The fault detection system constructed in this chapter is based on a simplified aircraft model that gives the explicit relationship between turn rates (p, q, r) , the inertia matrix \mathbf{I}^b , and the torques applied to the aircraft, that is, $[L \ M \ N]^T$, expressed in the body-axes frame (x_b, y_b, z_b) of the aircraft:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = (\mathbf{I}^b)^{-1} \left(\begin{bmatrix} L \\ M \\ N \end{bmatrix}^b - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I}^b \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right). \quad (44.1)$$

In the context of this work, the aircraft is a small UAV for which the aerodynamic moments have been modeled as follows (see Stevens and Lewis (2003), Stengel (2004), Möckli (2006), and Ducard (2009)):

$$\begin{aligned} L &= \bar{q} S b C_L(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, p, r, \beta), \\ M &= \bar{q} S \bar{c} C_M(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \alpha, q), \\ N &= \bar{q} S b C_N(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \delta_r, r, \beta), \end{aligned} \quad (44.2)$$

where the dynamic pressure is $\bar{q} = \frac{\rho V_T^2}{2}$, the total airspeed of the aircraft is V_T , the air density is ρ , the wing total surface is S , the wing span is b , and the mean aerodynamic wing chord is \bar{c} .

The aerodynamic derivatives are expressed as a linear combination of the state elements and control inputs as

$$\begin{aligned} C_L &= C_{L_{a1}} \delta_{a1} + C_{L_{a2}} \delta_{a2} + C_{L_{e1}} \delta_{e1} + C_{L_{e2}} \delta_{e2} + C_{L_{\bar{p}}} \bar{\bar{p}} + C_{L_{\bar{r}}} \bar{\bar{r}} + C_{L_{\beta}} \beta, \\ C_M &= C_{M1} + C_{M_{a1}} \delta_{a1} + C_{M_{a2}} \delta_{a2} + C_{M_{e1}} \delta_{e1} + C_{M_{e2}} \delta_{e2} + C_{M_{\bar{q}}} \bar{\bar{q}} + C_{M_{\alpha}} \alpha, \\ C_N &= C_{N_{\delta_r}} \delta_r + C_{N_{\bar{r}}} \bar{\bar{r}} + C_{N_{\beta}} \beta, \end{aligned} \quad (44.3)$$

with the dimensionless angular rates $\bar{\bar{p}} = \frac{bp}{2V_T}$, $\bar{\bar{q}} = \frac{\bar{c}q}{2V_T}$, $\bar{\bar{r}} = \frac{br}{2V_T}$. The last two differential equations concern the angle of attack α and the sideslip angle β as follows (see Ducard (2009) for the derivation of the following two formulae):

$$\begin{aligned} \dot{\alpha} &\approx q + \frac{g}{V_T} \left\{ 1 + \frac{\bar{q} S}{mg} ([C_{X1} + C_{Z\alpha}] \alpha + C_{Z1}) \right\}, \\ \dot{\beta} &\approx -r + \frac{\bar{q} S C_{Y1}}{m V_T} \beta, \end{aligned} \quad (44.4)$$

with the drag derivative C_{X1} , the side force derivative C_{Y1} , and the lift derivatives C_{Z1} , $C_{Z\alpha}$ being constant terms.

Finally, the inertia matrix expressed in the body-fixed frame is $\mathbf{I}^b = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix}$, with $I_{xz} = I_{zx}$. The numerical values of the parameters of the aircraft model used throughout this chapter are summarized in Table 44.2 in Appendix.

44.3 Residual Generator

In this chapter, the detection of a fault in the system is achieved by monitoring a signal that is within certain limits in normal conditions and that goes beyond these limits in faulty situations. This is the reason why a mathematical model has

been developed to describe the behavior of the aircraft. This model and the real aircraft receive the same actuator commands issued by the flight controller. The real aircraft dynamics are measured by sensors mounted aboard the vehicle. They will be compared with the predicted aircraft dynamics that are computed with the aircraft model. The difference between the real and predicted dynamics constitutes signals that are called residuals.

However, the mathematical model of the aircraft will never be perfect, leaving many effects unmodeled. Also, this model will only be approximated by the computer implementation. Furthermore, several parameters of the model will not be known exactly, and the sensor measurement data will be corrupted by noise and biases. For all these reasons, Kalman filtering techniques are employed in this work to take into account such system dynamics and measurement noise, errors, and uncertainties.

Kalman filters (KF) are well suited when the real world can be described by linear differential equations expressed in state space form and when the measurements are linear functions of the states. However, in most realistic problems, the real world is described by nonlinear differential equations. In order to take into account these nonlinearities, the residual generator of the FDI systems presented in this chapter uses extended Kalman filters (EKF), which are recalled below.

44.3.1 EKF Equations

The EKFs are designed based on a set of continuous nonlinear differential equations that describe the plant under consideration as follows (see Zarchan and Musoff (2005)):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{w}, \quad (44.5)$$

where the state vector is \mathbf{x} , the control input vector is \mathbf{u} , the set of nonlinear functions of the state and control vectors is $\mathbf{f}(\mathbf{x}, \mathbf{u})$, and the random zero-mean process noise vector is \mathbf{w} .

The continuous process noise covariance matrix describing the random process \mathbf{w} is given by

$$\mathbf{R}_w = E\{\mathbf{w}\mathbf{w}^T\}. \quad (44.6)$$

Equation 44.5 is first linearized around the current operating point and then discretized using the Euler integration method. Note that the Euler integration method is also used for the simulations presented in the following sections. The discretized form of (44.5) is expressed in state space form as

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{G}_k \mathbf{u}_k + \mathbf{w}_k, \quad (44.7)$$

where the state vector is evaluated at the discrete-time instant $t_k = kT_s$, with the constant T_s being the sampling period of the system. The control input vector at time step k is \mathbf{u}_k , and the discrete random zero-mean process noise \mathbf{w}_k is used to describe uncertainties in the model.

Finally, the discrete form of the measurement equation, either a linear or a nonlinear function of the states, is

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \quad (44.8)$$

where the discrete zero-mean random noise \mathbf{v}_k is described by the measurement noise covariance matrix $\mathbf{R}_{v,k} = E\{\mathbf{v}_k \mathbf{v}_k^T\}$ and consists of the variances of each of the measurement noise sources.

The discrete transition matrix ϕ_k is approximated by

$$\phi_k \approx \mathbf{I} + \mathbf{F}(k)T_s, \quad (44.9)$$

where the continuous system dynamics matrix $\mathbf{F}(k)$ is obtained by linearizing the continuous nonlinear equations and is evaluated at the latest available state estimate $\hat{\mathbf{x}}_{k|k}$ according to

$$\mathbf{F}(k) = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}, \mathbf{u}=\mathbf{u}_k}. \quad (44.10)$$

Similarly, the continuous measurement matrix $\mathbf{H}(k)$ is computed by linearizing the (possibly nonlinear) measurement equation $\mathbf{h}(\mathbf{x})$ and is successively evaluated at the latest available state estimate $\hat{\mathbf{x}}_{k|k}$ (see mechanization shown in Fig. 44.3) according to

$$\mathbf{H}(k) = \mathbf{H}_k = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}}. \quad (44.11)$$

The equations used in the EKF are described below (Kalman 1960; Brown and Hwang 1997; Zarchan and Musoff 2005). The schematic overview of the computation steps is shown in Fig. 44.3.

44.3.2 Filter's Equations

The equations used in the filter are as follows:

1. The Kalman gain matrix \mathbf{L}_k is computed as

$$\mathbf{L}_k = \Sigma_{k|k-1} \mathbf{H}_k^T [\mathbf{H}_k \Sigma_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_{v,k}]^{-1} \quad (44.12)$$

and is a function of the last-propagated state error covariance matrix $\Sigma_{k|k-1}$ and of the measurement noise covariance matrix $\mathbf{R}_{v,k}$.

2. The measurement update of the state estimate is as follows:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k [\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})], \quad (44.13)$$

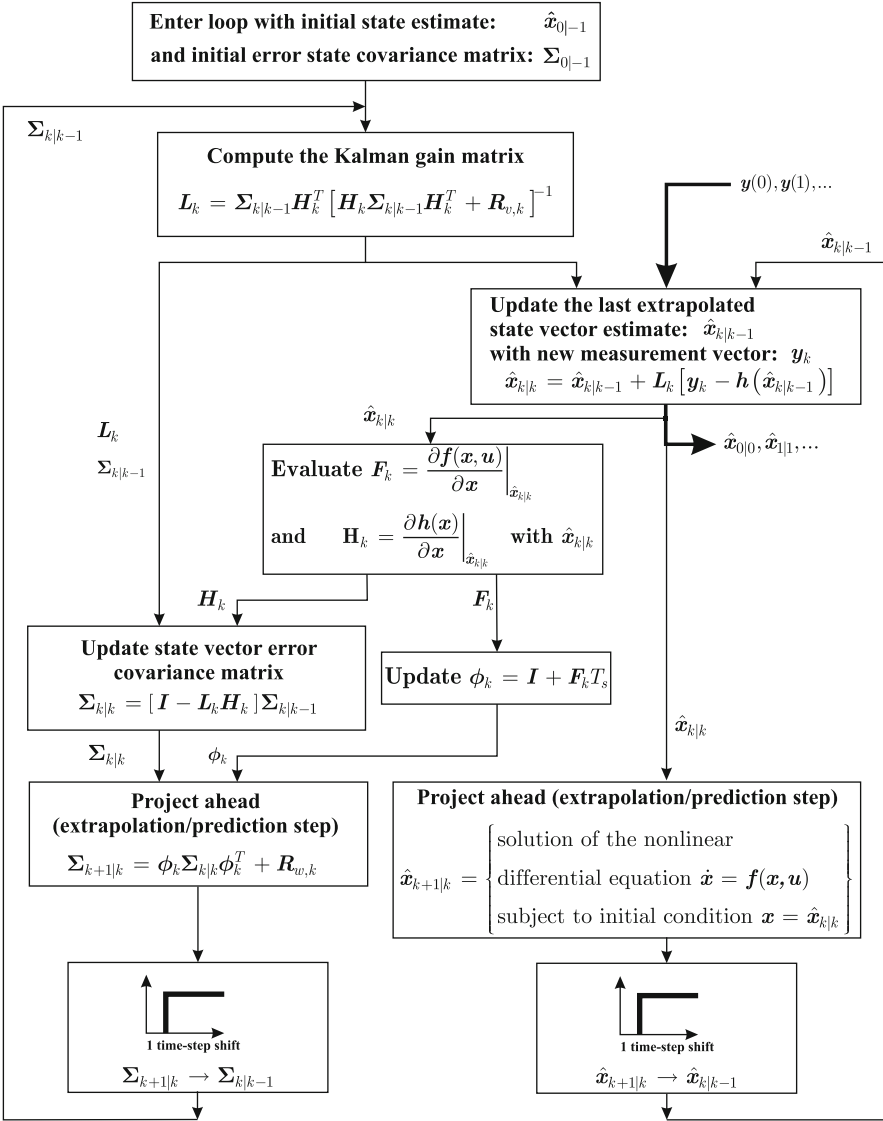


Fig. 44.3 EKF mechanization

where the last extrapolated state estimate is $\hat{\mathbf{x}}_{k|k-1}$, the measurement vector is \mathbf{y}_k , and the estimated measurement vector is $\mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$. The set of continuous nonlinear measurement equations is $\mathbf{h}(\cdot)$.

- The third step concerns the update of the state error covariance matrix $\Sigma_{k|k} = \mathbf{E}\{\mathbf{e}_{k|k} \mathbf{e}_{k|k}^T\}$ with $\mathbf{e}_{k|k} = \mathbf{x}(k) - \hat{\mathbf{x}}_{k|k}$, where $\mathbf{x}(k)$ is the true (unknown) value of the state vector at the discrete instant k . The matrix $\Sigma_{k|k}$ is recursively computed

as a function of the last predicted state error covariance matrix $\Sigma_{k|k-1}$ and the last computed Kalman gain matrix \mathbf{L}_k as follows:

$$\Sigma_{k|k} = [\mathbf{I} - \mathbf{L}_k \mathbf{H}_k] \Sigma_{k|k-1}. \quad (44.14)$$

4. The forward propagation of the state error covariance matrix is

$$\Sigma_{k+1|k} = \phi_k \Sigma_{k|k} \phi_k^T + \mathbf{R}_{w,k}, \quad (44.15)$$

where the matrix $\mathbf{R}_{w,k}$ represents the covariance of the discrete process noise acting on the elements of the state vector. The value of $\mathbf{R}_{w,k}$ is found from the continuous process noise covariance matrix \mathbf{R}_w and the continuous transition matrix ϕ according to

$$\mathbf{R}_{w,k} = \int_0^{T_s} \phi(\tau) \mathbf{R}_w \phi^T(\tau) d\tau, \quad (44.16)$$

where the matrix $\phi(t)$ is defined as follows:

$$\phi(t) = \mathbf{I} + \mathbf{F}(k)t. \quad (44.17)$$

There is an alternative solution of introducing process noise into the system through the control input matrix. Indeed, if the plant is described by the following equation:

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{G}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k, \quad (44.18)$$

then the process noise covariance matrix is defined by

$$\mathbf{R}_{w,k} = \mathbf{G}_k \mathbf{R}_w \mathbf{G}_k^T, \quad (44.19)$$

where the covariance matrix of the discrete process noise acting on the elements of the state vector is $\mathbf{R}_{w,k}$, and the discrete control input matrix is $\mathbf{G}_k =$

$$T_s \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{V_T(k)}.$$

5. The propagation forward of the state estimate does not have to be done with the discrete transition matrix ϕ_k , but rather it is done directly by integrating the actual nonlinear differential equations forward at each sampling interval. If the Euler integration technique is used, the extrapolated state estimate is computed with

$$\hat{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k|k} + \dot{\hat{\mathbf{x}}}_{k|k} T_s, \quad (44.20)$$

where the state time derivative is obtained from

$$\dot{\hat{\mathbf{x}}}_{k|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k). \quad (44.21)$$

Remarks:

- Note that the EKF presented in this section keeps track of the total estimates and not of the incremental ones (deviation from a nominal trajectory) as would be the case in a linearized KF. Indeed, the residuals are built from the difference between the true measurement vector and the predicted measurement vector using the set of nonlinear measurement equations acting on the total state estimate $\mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$.
- Moreover, the measurement update of the state estimate is done using the total estimate $\hat{\mathbf{x}}(k|k-1)$, and the propagation forward of the total state estimate is achieved using the set of nonlinear differential equations acting on the total state estimate $\mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k)$ instead of using the discrete transition matrix acting on incremental state quantities.
- Nevertheless, the computation of the Kalman gains and state error covariance matrices makes use of the recursively updated linear model through ϕ_k , \mathbf{H}_k , and $\mathbf{R}_{w,k}$.
- The schematic overview shown in Fig. 44.3 clarifies the mechanization of the EKF implemented in this work.

44.4 Multiple Model Approaches for FDI Systems

One popular approach to detect and isolate actuator or sensor faults is the multiple model adaptive estimation (MMAE) method (Magill 1965; Maybeck and Stevens 1991) as depicted in Fig. 44.4. It is based on a bank of parallel Kalman filters (KF), each of which is matching a particular fault status of the system. A hypothesis

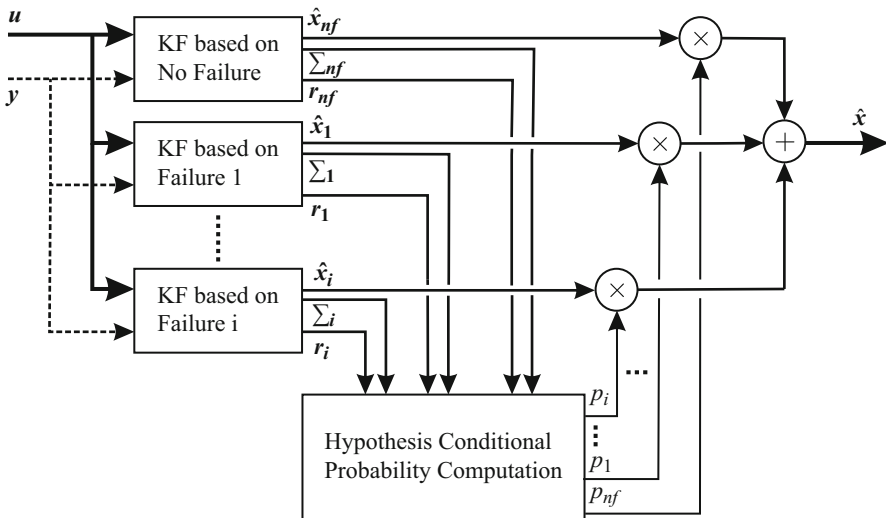


Fig. 44.4 Classical MMAE scheme

testing algorithm uses the vector of residuals \mathbf{r}_i and state error covariance matrix Σ_i from each KF to assign a conditional probability to each fault hypothesis. Several papers have demonstrated how the MMAE method can be used in the context of FDI systems for aircraft (Maybeck and Stevens 1991; Eide and Maybeck 1996; Maybeck 1999) and underwater vehicles (Ni 2001).

The main advantage of the MMAE method lies in its responsiveness to parameter variations, potentially leading to faster fault isolation than that obtained by other methods without a multiple model structure.

However, the MMAE method possesses three limitations to its successful application as explained in Ducard and Geering (2008). The first limitation concerns the number of filters that must be designed in order to span the range of possible fault scenarios, which must be limited due to computational load. The second limitation appears when an actuator is locked at an arbitrary nonzero position that biases the residuals of the KFs, leading to inaccurate fault detection and state estimation. Third, most of the implementations of an MMAE method only work efficiently around predefined operating conditions. This chapter shows how the MMAE method is modified to compensate the three limitations of the classical MMAE implementation by employing extended Kalman filters (EKF) to estimate the deflection of a faulty control surface (or actuator). The resulting method is called thus “extended multiple model adaptive estimation” (EMMAE), and its architecture is shown in Fig. 44.6 (Rupp et al. 2005).

44.4.1 Modeling Actuator Faults

A lock-in-place or floating actuator fault in the system can be seen as if the desired control input δ_j was disconnected and replaced by a faulty control signal $\bar{\delta}_j$ that takes control over the plant, as shown in Fig. 44.5. In a concise manner (Tao et al. 2004), the true input of the plant can be written as

$$u_i(t) = \delta_i(t) + \sigma_{A_i}(\bar{\delta}_i(t) - \delta_i(t)). \quad (44.22)$$

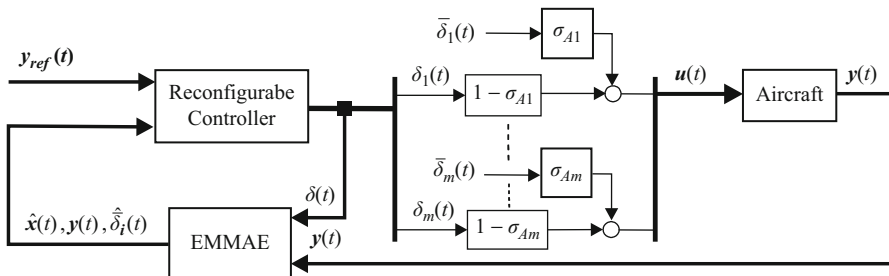


Fig. 44.5 Modeling of actuator faults

In the case of actuator failure(s), the vector of the (unknown) inputs is

$$\bar{\delta}(t) = [\bar{\delta}_1(t) \ \bar{\delta}_2(t) \ \dots \ \bar{\delta}_m(t)]^T, \quad (44.23)$$

with $\sigma = \text{diag}\{\sigma_{A1} \ \sigma_{A2} \ \dots \ \sigma_{Am}\}$, where

$$\sigma_{Aj} = \begin{cases} 1, & \text{if the } j\text{th actuator fails} \\ 0, & \text{otherwise} \end{cases}. \quad (44.24)$$

In the method presented below, the unknown parameters $\bar{\delta}_j$ are constantly estimated by their respective EKF. The conditional fault-hypothesis probabilities p_j assign the value for σ_{Aj} .

44.4.2 The EMMAE Method

The MMAE method is to be made applicable for any arbitrary lock-in-place faults or uncontrolled varying faults and at all flying conditions. Therefore, the original MMAE algorithm is modified by replacing the linear KFs by EKFs used as nonlinear estimators of the state vector and a fault parameter, namely, the deflection of a faulty control surface (or actuator) $\bar{\delta}_j$. The implementation of the EMMAE is depicted in Fig. 44.6. Contrary to the FDI designs with

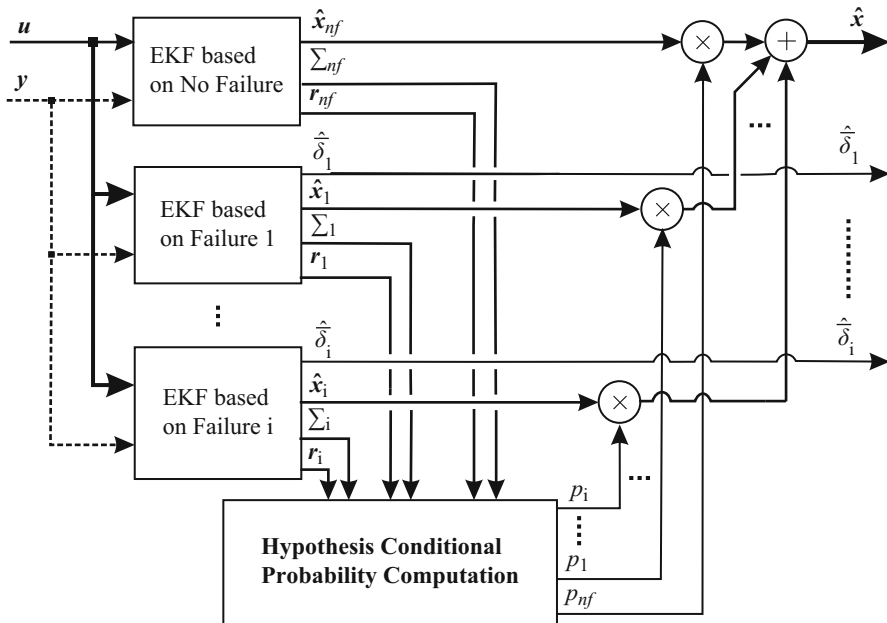


Fig. 44.6 EMMAE-FDI scheme: each EKF monitors its assigned actuator

the classical MMAE method where several KFs are designed for several faulty deflections for one actuator, in the EMMAE method, only one EKF is responsible for completely monitoring one actuator's health. Therefore, the EMMAE method drastically reduces the number of filters required for actuator health monitoring.

The addition of the actuator deflection estimate in the system state vector enables the EMMAE method to work for all the possible positions where an actuator can be locked or floating. In order to better illustrate the difference between the EMMAE and MMAE methods, Eqs. (44.25) and (44.26) recall how the models are defined in the regular MMAE method (Eide and Maybeck 1996; Maybeck 1999; Maybeck and Stevens 1991; Ni 2001) for an actuator or a sensor failure. The MMAE scheme considers a bank of linear models of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu}, \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du},\end{aligned}\tag{44.25}$$

where each model matches a fault scenario.

For example, the model that describes a failure of the j th actuator will have its \mathbf{B} matrix modified such that the j th column of the \mathbf{B} matrix is replaced by the very same column times a factor λ_j that varies from zero (complete loss of the actuator) to one (fully functioning actuator), see (44.26). Any intermediate value of λ_j indicates a reduction in the effectiveness of the j th actuator to modify the dynamics of the aircraft as shown in (44.26):

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{B}_j \mathbf{u}, \\ \dot{\mathbf{x}} &= \mathbf{Ax} + \begin{bmatrix} b_{11} & \cdots & b_{1j}\lambda_j & \cdots & b_{1N} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ b_{l1} & \cdots & b_{lj}\lambda_j & \cdots & b_{lN} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ b_{p1} & \cdots & b_{pj}\lambda_j & \cdots & b_{pN} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_j \\ \vdots \\ u_N \end{bmatrix}.\end{aligned}\tag{44.26}$$

However, this kind of approach for the modeling of actuator failure is very restrictive. Indeed, in the case of a total loss of the j th actuator, the factor λ_j equals zero. This means that whatever control input the controller generates for the j th actuator, it has no influence on the dynamics of the aircraft, and the faulty actuator deflection is considered to be zero. Note that if the j th actuator is actually locked at a nonzero deflection angle, the faulty actuator deflection has an influence on the dynamics of the aircraft. This condition results in an unknown bias term that will prevent the j th KF in the MMAE method from working properly. Therefore, the residuals will be biased, and the state estimation and the computation of the probabilities will be incorrect as well.

In the EMMAE method, both the control input matrix and the dynamics matrix are modified. Indeed, in order to define a model that describes a failure of the j th actuator, the j th column of the control input matrix is zeroed and the state vector is augmented with the j th actuator deflection $\bar{\delta}_j$. The dynamics matrix is also augmented with the original j th column of the control input matrix. In this way, the control inputs from the controller to the j th actuator are totally ignored, but the faulty deflection $\bar{\delta}_j$ that is constantly estimated ($\hat{\bar{\delta}}_j$) in the state vector contributes to modifying the dynamics of the aircraft model of the j th filter. It yields residuals that are the smallest for the filter matching the occurring fault. The next section illustrates how the filters in the EMMAE are constructed in practice.

44.4.3 Designing the EKF for the No-Fault Scenario

In the EMMAE method, the residuals $\mathbf{r}(k)$ are obtained as follows:

$$\mathbf{r}(k) = \mathbf{y}(k) - \hat{\mathbf{x}}_{k|k-1}, \quad (44.27)$$

where the filter's last-propagated state estimate vector is $\hat{\mathbf{x}}_{k|k-1} = [\hat{p} \ \hat{q} \ \hat{r} \ \hat{\alpha} \ \hat{\beta}]_{(k|k-1)}^T$ and the measurement vector is $\mathbf{y}(k) = [p, q, r, \alpha, \beta]_{(k)}^T$.

The continuous system dynamics matrix for the no-fault filter $\mathbf{F}_{nf}(k)$ evaluated at time step k can be explicitly derived from the nonlinear model detailed in Sect. 44.2 as

$$\mathbf{F}_{nf}(k) = \begin{bmatrix} 0 & \frac{Sb[I_{zz}C_{L\bar{\beta}} - I_{xz}C_{N\bar{\beta}}]}{D_1}\bar{q} \\ \mathbf{F}_1 & \frac{S\bar{c}C_{M\alpha}}{I_{yy}}\bar{q} & 0 \\ 0 & 0 & \frac{Sb[I_{xx}C_{N\bar{\beta}} - I_{xz}C_{L\bar{\beta}}]}{D_1}\bar{q} \\ 0 & 1 & 0 & \frac{\rho V_T S C_{Z\alpha}}{2m} & 0 \\ 0 & 0 & -1 & 0 & \frac{\rho V_T S C_{Y1}}{2m} \end{bmatrix} \hat{\mathbf{x}}_{nf}(k|k), \bar{q}(k) \quad (44.28)$$

with the submatrix \mathbf{F}_1 defined as

$$\mathbf{F}_1 = \begin{bmatrix} \frac{I_{zz}Sb^2C_{L\bar{\beta}}}{2D_1V_T}\bar{q} - \frac{N_1}{D_1}q & -\frac{N_1}{D_1}p + \frac{N_2}{D_1}r & \frac{(I_{zz}C_{L\bar{\beta}} - I_{xz}C_{N\bar{\beta}})Sb^2}{2D_1V_T}\bar{q} + \frac{N_2}{D_1}q \\ -\frac{I_{xx} - I_{zz}}{I_{yy}}r + 2\frac{I_{xz}}{I_{yy}}p & \frac{S\bar{c}^2C_{M\bar{q}}}{2V_T I_{yy}}\bar{q} & -\frac{I_{xx} - I_{zz}}{I_{yy}}p - \frac{2I_{xz}}{I_{yy}}r \\ -\frac{Sb^2C_{L\bar{\beta}}I_{xz}}{2D_1V_T}\bar{q} + \frac{N_3}{D_1}q & \frac{N_3}{D_1}p + \frac{N_1}{D_1}r & \frac{Sb^2[-I_{xz}C_{L\bar{\beta}} + I_{xx}C_{N\bar{\beta}}]}{2D_1V_T}\bar{q} + \frac{N_1}{D_1}q \end{bmatrix}, \quad (44.29)$$

where $N_1 = I_{xz}(I_{xx} - I_{yy} + I_{zz})$, $N_2 = I_{yy}I_{zz} - I_{xz}^2 - I_{zz}^2$, $N_3 = I_{xz}^2 - I_{xx}I_{yy} + I_{xx}^2$ and $D_1 = I_{xx}I_{zz} - I_{xz}^2$.

The discrete transition matrix for the no-fault filter is calculated with $\phi_{nf,k} = \mathbf{I} + \mathbf{F}_{nf}(k)T_s$. The control input matrix of the no-fault filter $\mathbf{G}_{nf}(k)$ is computed as follows:

$$\mathbf{G}_{nf}(k) = \bar{q}_k \begin{bmatrix} \frac{SbI_{zz}CLa_1}{D_1} & \frac{SbI_{zz}CLa_2}{D_1} & \frac{SbI_{zz}CLe_1}{D_1} & \frac{SbI_{zz}CLe_2}{D_1} & \frac{-SbI_{xz}CN_{\delta r}}{D_1} \\ \frac{S\tilde{c}C_{Ma1}}{I_{yy}} & \frac{S\tilde{c}C_{Ma2}}{I_{yy}} & \frac{S\tilde{c}C_{Me1}}{I_{yy}} & \frac{S\tilde{c}C_{Me2}}{I_{yy}} & 0 \\ \frac{-SbI_{xz}CLa_1}{D_1} & \frac{-SbI_{xz}CLa_2}{D_1} & \frac{-SbI_{xz}CLe_1}{D_1} & \frac{-SbI_{xz}CLe_2}{D_1} & \frac{SbI_{xx}CN_{\delta r}}{D_1} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (44.30)$$

The discrete control input matrix for the no-fault filter is $\mathbf{G}_{nf,k} = \mathbf{G}_{nf}(k)T_s$.

44.4.4 Augmenting the State Vector with the Faulty Actuator Parameter δ_i

The state vector of the i th filter is augmented to monitor the occurrence of the i th actuator fault. The deflection of the failed actuator is included in the state vector in a way to be estimated by the EKF. Therefore, the state vector for each filter i is

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{x} \\ \bar{\delta}_i \end{bmatrix}. \quad (44.31)$$

The augmented state vector leads to the following state space equations for each filter i :

$$\begin{aligned} \mathbf{z}_i(k+1) &= \mathbf{f}_{z_i}(\mathbf{z}_i(k), \delta(k)) + \mathbf{w}_k, \\ \mathbf{y}_i(k) &= \mathbf{h}(\mathbf{z}_i(k)) + \mathbf{v}_k, \end{aligned} \quad (44.32)$$

Using the above equations, the linearized system evaluated at each sampling time can be written as

$$\begin{aligned} \begin{bmatrix} \mathbf{x}(k+1) \\ \bar{\delta}_i(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{F}(k) & \mathbf{G}^i(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \bar{\delta}_i(k) \end{bmatrix} + \begin{bmatrix} \mathbf{G}^{(0,i)}(k) \\ 0 \end{bmatrix} \delta(k), \\ \mathbf{y}(k) &= [\mathbf{H} \quad 0] \begin{bmatrix} \mathbf{x}(k) \\ \bar{\delta}_i(k) \end{bmatrix}. \end{aligned} \quad (44.33)$$

where $\mathbf{G}^{(i)}$ represents the i th column of \mathbf{G} and $\mathbf{G}^{(0,i)}$ represents the matrix \mathbf{G} with its i th column set to zero.

44.4.5 Designing the EKF for the Case of a Failure on Aileron 1

This section provides an example of how to derive the matrices for the EKF corresponding to the scenario of a lock-in-place or floating actuator failure. Aileron 1 is taken as an example.

The system dynamics matrix for the aileron1-fault filter $\mathbf{F}_{\delta a1}(k)$ is explicitly derived from the nonlinear model as

$$\mathbf{F}_{\delta a1}(k) = \begin{bmatrix} 0 & \frac{Sb[I_{zz}C_{L\beta} - I_{xz}C_{N\beta}]}{D_1}\bar{q} & \frac{SbI_{zz}C_{La1}}{D_1}\bar{q} \\ \mathbf{F}_1 & \frac{S\bar{c}C_{Ma}}{I_{yy}}\bar{q} & 0 & \frac{S\bar{c}C_{Ma1}}{I_{yy}}\bar{q} \\ 0 & \frac{Sb[I_{xx}C_{N\beta} - I_{xz}C_{L\beta}]}{D_1}\bar{q} & \frac{-SbI_{xz}C_{La1}}{D_1}\bar{q} \\ 0 & 1 & 0 & \frac{\rho V_T S C_{Z\alpha}}{2m} & 0 \\ 0 & 0 & -1 & 0 & \frac{\rho V_T S C_{Y1}}{2m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{z}}_1(k|k) \quad (44.34)$$

The discrete transition matrix for the aileron1-fault filter is calculated with $\phi_{k,\delta a1} = \mathbf{I} + \mathbf{F}_{\delta a1}(k)T_s$. The control input matrix of the no-fault filter $\mathbf{G}_{\delta a1}(k)$ is computed as follows:

$$\mathbf{G}_{\delta a1}(k) = \begin{bmatrix} 0 & \frac{SbI_{zz}C_{La2}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le1}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{N\delta r}}{D_1}\bar{q} \\ 0 & \frac{S\bar{c}C_{Ma2}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me1}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me2}}{I_{yy}}\bar{q} & 0 \\ 0 & \frac{-SbI_{xz}C_{La2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le1}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le2}}{D_1}\bar{q} & \frac{SbI_{xx}C_{N\delta r}}{D_1}\bar{q} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \bar{q}(k) \quad (44.35)$$

The discrete control input matrix for the aileron1-fault filter is $\mathbf{G}_{\delta a1,k} = \mathbf{G}_{\delta a1}(k)T_s$. All the other filters monitoring the other actuators are designed in a similar way.

44.4.6 Actuator Fault Isolation

44.4.6.1 Hypothesis Testing

A hypothesis testing algorithm uses the residuals and the state error covariance matrix from each EKF to assign a conditional probability to each fault scenario.

The estimated state vector of the system is the sum of the state vector of each EKF weighted by its corresponding probability

$$\hat{\mathbf{x}}[k] = \sum_i \hat{\mathbf{x}}_i[k] \cdot p_i[k], \quad (44.36)$$

where $\hat{\mathbf{x}}_i[k]$ is the state estimate computed by the EKF that assumes the fault scenario θ_i . The index i covers all the fault scenarios implemented, including the no-fault case. The term $p_i[k]$ denotes the probability that the i th fault scenario is occurring. Now, the main difficulty lies in the online computation of the probability $p_i[k]$. In order to determine which fault scenario the actual plant is the closest to, it is needed to consider the measurement data from the sensors. The last available measurement vector is $\mathbf{y}[k]$, sometimes also written \mathbf{y}_k in the following. The sequence of the last measurement vectors is defined as $\mathbf{Y}_k = \{\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{y}_{k-2}, \dots, \mathbf{y}_0\}$. The fault probability $p_i[k]$ can be expressed as the a posteriori conditional probability $p_i[k] = p(\theta = \theta_i | \mathbf{Y}_k)$, that is, the probability that the actual plant can be categorized in the scenario θ_i given the sequence of the last measurements \mathbf{Y}_k . The Bayes' law states that

$$p_i[k] = p(\theta = \theta_i | \mathbf{Y}_k) = \frac{p[\mathbf{Y}_k | \theta = \theta_i] p[\theta = \theta_i]}{p[\mathbf{Y}_k]}, \quad (44.37)$$

where the probability $p[\mathbf{Y}_k]$ can be decomposed as

$$\begin{aligned} p[\mathbf{Y}_k] &= p[\mathbf{Y}_k | \theta = \theta_1] \cdot p[\theta = \theta_1] + \dots + p[\mathbf{Y}_k | \theta = \theta_N] \cdot p[\theta = \theta_N], \\ &= \sum_{j=0}^N p[\mathbf{Y}_k | \theta = \theta_j] \cdot p[\theta = \theta_j]. \end{aligned} \quad (44.38)$$

Combining (Eq. 44.37) and (Eq. 44.38) yields

$$p_i[k] = p[\theta = \theta_i | \mathbf{Y}_k] = \frac{p[\mathbf{Y}_k | \theta = \theta_i] p[\theta = \theta_i]}{\sum_{j=0}^N p[\mathbf{Y}_k | \theta = \theta_j] \cdot p[\theta = \theta_j]}, \quad (44.39)$$

with N being the number of different scenarios under consideration. In order to make a recursive form appear in the probabilities, the measurement data sequence \mathbf{Y}_k is rewritten as the sequence $\{\mathbf{y}_k, \mathbf{Y}_{k-1}\}$:

$$\begin{aligned} p[\mathbf{Y}_k | (\theta = \theta_j)] &= p[\mathbf{y}_k, \mathbf{Y}_{k-1} | (\theta = \theta_j)] \\ &= p[\mathbf{y}_k | (\mathbf{Y}_{k-1}, \theta = \theta_j)] \cdot p[\mathbf{Y}_{k-1} | (\theta = \theta_j)] \\ &= p[\mathbf{y}_k | (\theta = \theta_j, \mathbf{Y}_{k-1})] \cdot p[(\theta = \theta_j) | \mathbf{Y}_{k-1}] \\ &= p[\mathbf{y}_k | (\theta = \theta_j, \mathbf{Y}_{k-1})] \cdot p_j[k-1]. \end{aligned} \quad (44.40)$$

Using the result (Eq. 44.40) in (Eq. 44.39) yields

$$p_i[k] = p[(\theta = \theta_i) | \mathbf{Y}_k] = \frac{p[\mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})] \cdot p_i[k-1] \cdot p[\theta = \theta_i]}{\sum_{j=0}^N p[\mathbf{y}_k | (\theta = \theta_j, \mathbf{Y}_{k-1})] \cdot p_j[k-1] \cdot p[\theta = \theta_j]}. \quad (44.41)$$

Since a fault may occur at any time, regardless of which actuator may fail, the same probability is assigned to all the scenarios, that is, $p[\theta = \theta_j] = 1/N$ for $j = 1, \dots, N$. Therefore, the equation above simplifies to the following recursive expression:

$$p_i[k] = p[\theta = \theta_i | \mathbf{Y}_k] = \frac{p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})] \cdot p_i[k-1]}{\sum_{j=0}^N p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_j, \mathbf{Y}_{k-1})] \cdot p_j[k-1]}. \quad (44.42)$$

Remarks:

- By examining the probabilities, the “health status” of the system can be determined. It is either the no-fault case or a case where an actuator is locked-in-place or floating. An actuator fault is declared valid if the corresponding fault probability exceeds 90 % for a certain amount of time. A fault is declared removed when the corresponding fault probability is below 5 % for a certain amount of time.
- One may notice that the denominator of (Eq. 44.42) corresponds to the sum of each scenario probability’s numerator, such that fault probabilities add up to one.
- In practice, in order to prevent the possibility that the recursive computation of the fault probability in (Eq. 44.42) stays at zero forever as soon as the probability reaches zero, the lower bound of each probability is set to 0.001.
- This method that uses probabilities for fault isolation is sometimes called a Bayes classifier (see Isermann (2006), Chap. 16).

44.4.6.2 Gaussian Conditional Probability Density

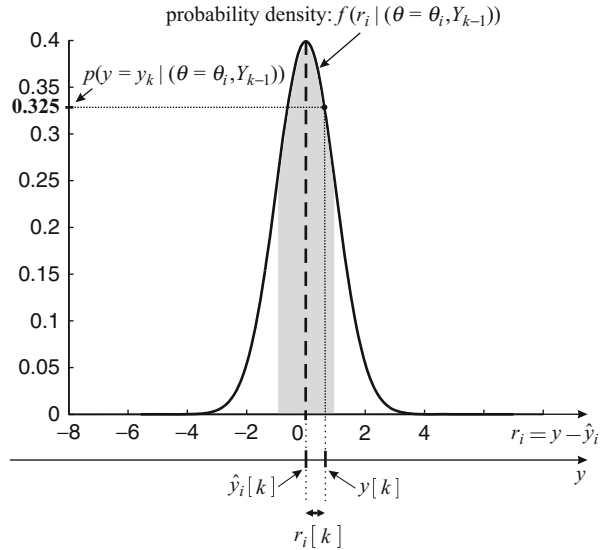
This section provides an explicit formula for the term $p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})]$, which corresponds to the probability of obtaining the measurement data $\mathbf{y}[k]$ at time $t_k = kT_s$, assuming the scenario θ_i exists and given the sequence of the last measurements \mathbf{Y}_{k-1} .

The probability density is chosen to be a Gaussian function (Maybeck 1994) with its characteristic bell-shaped curve according to the following formula:

$$p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})] = \lambda_i[k] e^{-\mathbf{r}_i[k]^T \Sigma_i^{-1}[k] \mathbf{r}_i[k]/2}, \quad (44.43)$$

with $\lambda_i[k] = \frac{1}{(2\pi)^{m/2} |\Sigma_i[k]|^{1/2}}$, where $|\dots|$ denotes the determinant of the matrix, m represents the measurement dimension, and $\Sigma_i[k]$ is the residual covariance matrix calculated at time step k by the i th EKF. The term $\mathbf{r}_i[k]$ corresponds to the residuals of the i th EKF, when the measurement update step occurs according to $\mathbf{r}_i[k] = \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_i(k|k-1))$.

Fig. 44.7 Conditional probability density in the scalar case



Intuitive Explanation of the Probability Density

In the case of a single-input single-output problem, in which the state and measurement vectors reduce to scalars, the measurement data $h(\hat{x}_i(k|k-1))$ that are expected according to the model can be seen as the mean value of the measurement data computed by the i th EKF, that is, $\hat{y}_i[k]$ in Fig. 44.7.

The width of the conditional Gaussian density is governed only by the covariance term $\Sigma_i[k]$. Figure 44.8 shows the shapes of the Gaussian function for several standard deviations $\sigma^2 = \Sigma(j,j)$. The residual $r_i[k]$ determines the relative position of the peak of the probability density with the actual measurement y_k .

In the multivariable case, which applies in this FDI system, the fault probability

$$p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})] \quad (44.44)$$

is given by $\mathbf{f}(\mathbf{r}_i = \mathbf{r}_i[k] | (\theta = \theta_i, \mathbf{Y}_{k-1}))$, with the probability density defined as a function of the residual \mathbf{r}_i with

$$f(\mathbf{r}_i | (\theta = \theta_i, \mathbf{Y}_{k-1})) = \frac{1}{(2\pi)^{m/2} |\Sigma_i[k]|^{1/2}} \cdot e^{-\mathbf{r}_i^T \Sigma_i^{-1}[k] \mathbf{r}_i / 2}. \quad (44.45)$$

Therefore, the filter that corresponds to the fault scenario produces an estimate for the measurement vector $\hat{\mathbf{y}}_i[k] = \mathbf{h}(\hat{\mathbf{x}}_i(k|k-1))$ very close (apart from noise) to the actual value of the measurement data vector $\mathbf{y}[k]$. The residual $\mathbf{r}_i[k] = \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_i(k|k-1))$ will be small and close to zero. This means that the corresponding probability $p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})]$ is the highest for the filter matching the fault scenario.

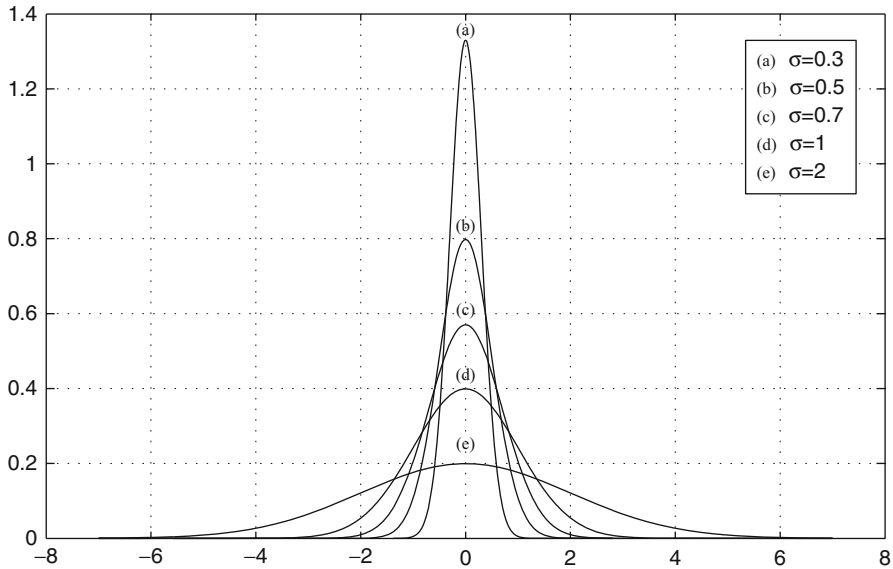


Fig. 44.8 Gaussian functions for zero-mean value and several standard deviations σ

By examining the probabilities computed with (Eq. 44.46), the health status of the system is determined; it is either the no-fault case or actuator-failure case:

$$p_i[k] = p[\theta = \theta_i | \mathbf{Y}_k] = \frac{p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})] \cdot p_i[k-1]}{\sum_{j=0}^N p[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_j, \mathbf{Y}_{k-1})] \cdot p_j[k-1]}. \quad (44.46)$$

Remarks:

- The hypothesis testing uses a Gaussian density function, which assumes that the residuals from the EKF are Gaussian distributed. When this is not the case, there is a little inconsistency with the application of the theory. However, the assumption that these residuals are Gaussian distributed is still reasonable, especially when the aircraft dynamics are slow.
- The reason why a Gaussian distribution is used to describe the probability density of the current measurement to take on the value y_k based on the fault hypothesis θ_i and the previous measurements \mathbf{Y}_{k-1} is to make the mathematics tractable. As mentioned in the book by Maybeck (1994), “the Kalman filter, which propagates the first and second-order statistics [mean and variance of a process], includes all information contained in the [Gaussian] conditional probability density, rather than only some of it, as would be the case with a different form of density.” If another probability density function was known, then (Eq. 44.45) could be changed accordingly.

44.4.7 Simulation Results of the EMMAE-FDI with No Supervision System

44.4.7.1 Simulation Conditions

In order to obtain realistic simulations, the sensor measurement data are corrupted on purpose with zero-mean white Gaussian noise corresponding to typical specifications of low-cost sensors. For the turn rate sensors, the standard deviation is $\sigma_{p,q,r} = 5^\circ/\text{s} = 0.0873 \text{ rad/s}$, which corresponds to a noise covariance of $\Sigma_{p,q,r} = 0.0076 \times I_3 \text{ (rad}^2/\text{s}^2\text{)}$. For the airflow angle sensors, the noise standard deviation is $\sigma_{\alpha,\beta} = 2^\circ = 0.0349 \text{ rad}$ ($\Sigma_{\alpha,\beta} = 0.0012 \times I_2 \text{ [rad}^2\text{]}$). The airspeed sensor noise has a standard deviation of $\sigma_{V_T} = 1 \text{ m/s}$ ($\Sigma_{V_T} = 1 \text{ m}^2/\text{s}^2$). Poor sensor quality adversely affects the FDI reliability.

When there is little excitation and when the faulty actuator deflection is close to the trim conditions, an actuator failure becomes even more difficult to detect. Indeed, due to large sensor noise, the control signals (see Fig. 44.9) become noisy as well, which reduces the difference between the actual faulty actuator deflection and its corresponding control signal. The larger this difference, the easier it is to detect the fault.

The EKF process noise covariance matrix and the sensor noise covariance matrix are selected as follows: $\mathbf{R}_w = 0.002 \times \mathbf{I}_5$ and $\mathbf{R}_v = \text{diag}[0.1 \times \mathbf{I}_3 \quad 0.02 \times \mathbf{I}_2]$.

44.4.7.2 Scenario

The scenario to test the fault detection method is chosen to put the FDI in the most difficult conditions, which are those of minimum excitation of the system. This is achieved when the aircraft is flying straight and level (no maneuvers, no wind) at a constant speed of 30 m/s.

The actuator faults are simulated by blocking the control surfaces close to the trim deflections corresponding to straight level flight conditions, because those faults are harder to detect and to estimate. For example, the ailerons and the rudder are intentionally made to fail close to the neutral deflection (0°), and the elevators are made to fail close to -2° . Fault detection with the EMMAE method is tested on a 6 degree-of-freedom nonlinear aircraft model of a UAV used in Möckli (2006) and Ducard (2009).

Simulations were performed in MATLAB®/Simulink® on closed-loop control architecture, with a nonlinear autopilot which regulates the speed, altitude, and the attitude of the aircraft. The actuator configuration of the aircraft is depicted in Fig. 44.2. The EMMAE-FDI system is therefore composed of six EKFs, one for monitoring the no-fault case, two for monitoring each aileron (one on each wing), two EKFs for monitoring each of the two independent elevators, and one EKF for the rudder.

As depicted in Fig. 44.9, a sequence of consecutive faults is generated. From $t = 10 \dots 40 \text{ s}$, aileron 1 fails and is locked at -1° deflection; for $t = 70 \dots 100 \text{ s}$, aileron 2 fails and is “floating” between the two positions -1° and 1° in a square-wave fashion. For $t = 130 \dots 160 \text{ s}$, the rudder fails and is locked at -1° .

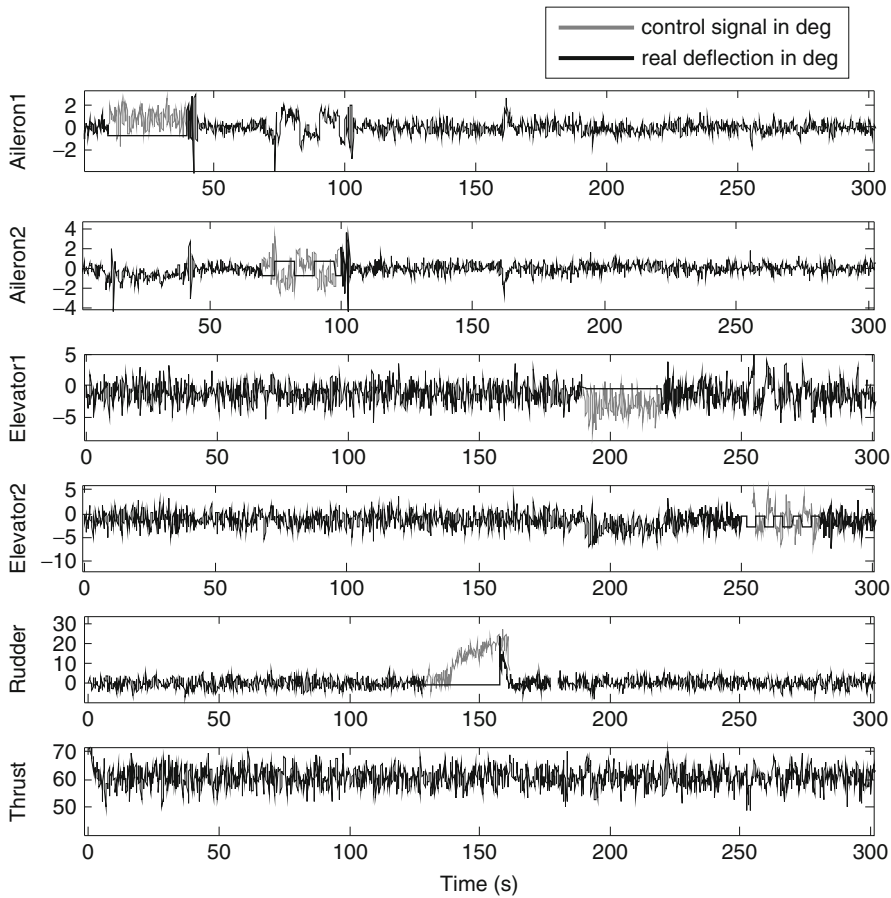


Fig. 44.9 Control signals and actual actuator deflections

For $t = 190 \dots 220$ s, elevator 1 gets locked at -0.5° , and finally, for $t = 250 \dots 280$ s, elevator 2 is floating between two uncontrolled positions -1° and -3° in a square-wave fashion. After this sequence of faults, the aircraft continues to fly straight and level, and no more faults are introduced.

44.4.7.3 Comments on the Simulation Results

Figure 44.10 shows the results obtained by the FDI system after the sequence of faults. The top plot labeled “no fault” has a probability of 1 when the EMMAE-FDI system does not detect any fault in the aircraft. An actuator fault is declared valid if the corresponding fault probability exceeds 90 % for a certain amount of time. A fault is declared removed when the corresponding fault probability is below 5 % for a certain amount of time. When aileron 1 fails at $t = 10$ s, the “No-Fault” filter needs about 6 s for its probability to go down to almost 0, which means that a failure

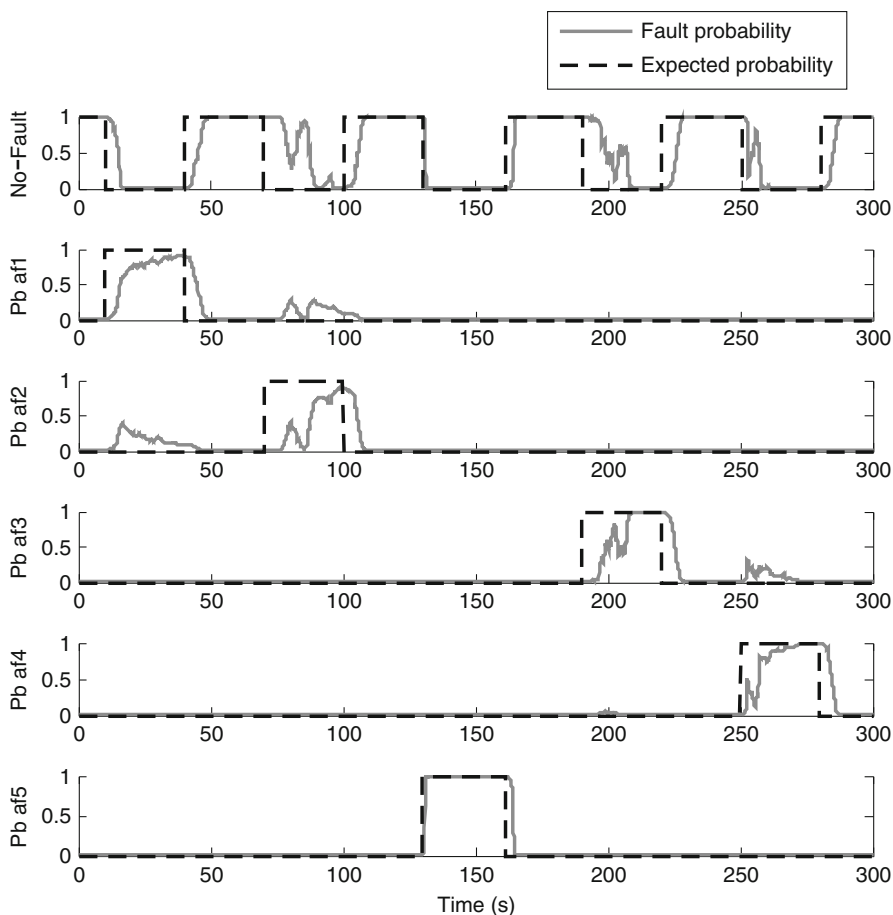


Fig. 44.10 Probabilities from each filter of the EMMAE-FDI after a sequence of faults (no supervision system)

occurred somewhere. After the aileron 1 fault is introduced, both probabilities for aileron 1 and 2 to fail ($Pbaf1$, $Pbaf2$) start rising at $t = 11.5$ s. At $t = 17$ s, the FDI system begins to distinguish between the two ailerons which one has failed, and $Pbaf2$ returns to zero while $Pbaf1$ rises up to 90 % at $t = 34$ s. Therefore, it took 24 s for the FDI system to indicate that aileron 1 experiences a failure. At $t = 40$ s, the fault of aileron 1 is removed, and the actuator behaves normally again. Figure 44.10 shows that the probability $Pbaf1$, indicating whether aileron 1 fails, decreases slowly and reaches 0 again after 10 s, while the “No-Fault” probability rises accordingly. It thus takes 10 s for the FDI system to indicate that the fault has been removed.

As Fig. 44.9 shows, at $t = 70$ s, aileron 2 fails and has an uncontrolled square-wave motion between -1° and 1° . Figure 44.10 shows that the FDI system takes 20 s to detect such a failure. An ambiguity exists between the two ailerons for a few

seconds before the probability P_{baf2} finally reaches 90 % at $t = 90$ s. At $t = 100$ s, the fault of aileron 2 is removed, and this actuator behaves normally again. However, the FDI system is not capable of detecting quickly that the fault has been removed. It takes about 8 s to do so.

For the rudder, the fault is introduced at $t = 130$ s and is isolated by the FDI system at $t = 131$ s when P_{baf5} exceeds 90 %. The fault removal is detected in less than 5 s. It takes less time to isolate a rudder fault and to detect its removal because there is only one rudder, unlike the other actuators which are redundant (two ailerons, two elevators). Therefore, a malfunctioning rudder cannot be compensated by a redundant rudder, thus resulting in no actuator-fault ambiguity.

The introduction of the elevator 1 fault at $t = 190$ s is isolated by its corresponding filter (probability signal P_{baf3} exceeding 90 %) at $t = 208$ s. After the fault removal, the FDI system takes 8 s to indicate the fault removed. The behavior of elevator 2 is similar to the behavior of elevator 1. Finally, after the last fault has been removed, 5 s are needed by the FDI system to slowly build up probability in the “No-Fault” filter to indicate that no more faults are present in the system.

44.4.8 Remarks on the First Attempt to Use the EMMAE-FDI System

- The results plotted in Fig. 44.10 indicate that the current implementation of the method is able to detect the fact that a failure occurred, even in a very low-excitation case, but it could not tell quickly and reliably which actuator experienced the fault. In cases of redundant actuators having the same influence on the aircraft aerodynamics, the EMMAE method has difficulty quickly resolving ambiguities between redundant actuators when they cannot be properly excited.
- It appears that failures of actuators near trim deflection are more difficult to detect and isolate.
- Moreover, whenever a fault is removed, the EMMAE method alone requires a long time to detect that fact. It is critical, however, that the FDI quickly detects the removal of a failure or quickly recognizes that a false alarm has been triggered due to possible external perturbations, such as strong wind gusts.
- Finally, it is critical that the probabilities quickly reach the “expected values” which correctly describe the fault scenario. Indeed, the estimated state vector of the system, which is the sum of the state vector of each EKF weighted by its corresponding probability, must be sufficiently correct and accurate if this state estimate is fed back to the controller.

44.4.9 Techniques to Improve Actuator Fault Diagnosis

This section describes some techniques added to enhance the performance of the FDI system, when there is very low excitation of the system, particularly during steady level flight. In order to improve the speed and the accuracy of the fault isolation, a supervision module is designed whose tasks are detailed below.

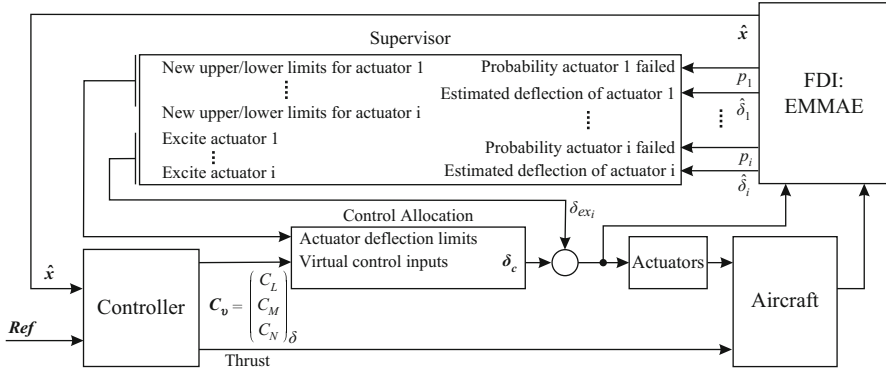


Fig. 44.11 EMMAE-FDI system in a reconfigurable flight control system

44.4.9.1 Design of an Active Supervision Module (Supervisor)

The supervision module shown in Fig. 44.11 is designed to monitor probability signals from the FDI. If an actuator-failure probability exceeds a certain threshold for some time, then the supervisor is designed to superimpose an artificial control signal on the corresponding actuator. If an actuator fails, the additional signal will have no effect on the aircraft dynamics, but it will help the FDI confirm more quickly the failure of this actuator. On the other hand, if the actuator has not actually failed, the aircraft will respond according to the additional signal, and the FDI will then remove the fault assigned to this actuator. If the corresponding fault probability falls below a certain threshold for a defined period of time, the supervisor removes the superimposed signal.

The excitation signals can also be adaptively controlled within certain limits, for example, from 1° to 4° , with the function $\delta_{ex,i}(t) = [1^\circ + 3^\circ(1 - p_i(t))] \cos(2\pi f_i t)$, see Fig. 44.12.

Note that an actuator is only excited when its corresponding fault probability p_i exceeds 5%. Most of the time, only one actuator is excited. If several actuators are to be excited simultaneously, better detection performance may be expected if the excitation signals to each actuator are independent and uncorrelated. The frequency of the signal is to be chosen within the range of the aircraft bandwidth; in the case of this chapter, $f_i = 1$ Hz.

The excitation signal has an adaptive amplitude dependent on the probability p_i of actuator i to have failed. In this way, when the probability p_i is low, the excitation amplitude is large and vice versa. Simulation results show that this adaptive amplitude for the excitation signal efficiently improves the accuracy and speed for fault isolation compared with a fixed amplitude excitation signal. This adaptive amplitude ensures that the actuator is excited as little as possible, but still enough to isolate the fault or to remove a false alarm. Figure 44.12 shows the practical implementation of the excitation signal generator and provides an example for aileron 1.

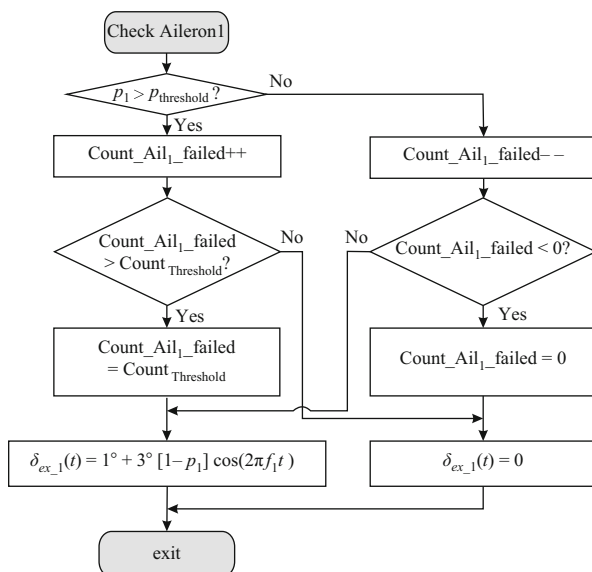


Fig. 44.12 Generation of the actuator excitation signal (example for aileron 1)

This method is therefore a systematic way of testing each triggered failure alarm, to confirm it or to remove it, hence making the FDI more robust. Whereas other proposed schemes (Ni 2001; Azam et al. 2005) suggest having the aircraft perform a “health-check maneuver” or “diagnostic maneuver” as soon as a failure is detected, in the method presented in this chapter, the actuator is directly excited by the supervision module rather than by the aircraft autopilot, yielding much faster and more accurate fault isolation.

44.4.9.2 Fault Detection Performance of the EMMAE-FDI with the Supervision System

Figure 44.13 shows how the aileron faults are accurately detected and isolated in less than 5 s. The rudder fault is isolated after 1 s. The elevator faults take longer (about 9 s) to be isolated. However, the removal of all the faults is detected in less than 5 s. Furthermore, there is no more ambiguity or false detection among the actuator faults. The results shown in Fig. 44.13 compared to those of Fig. 44.10 indicate that the performance and the robustness of the EMMAE-FDI system have greatly improved due to the supervision module.

It is recalled that these results are obtained in the most difficult conditions for the FDI system. Indeed, there is no external disturbance such as wind gusts, the aircraft flies straight and level, and the actuators fail close to their trim deflection.

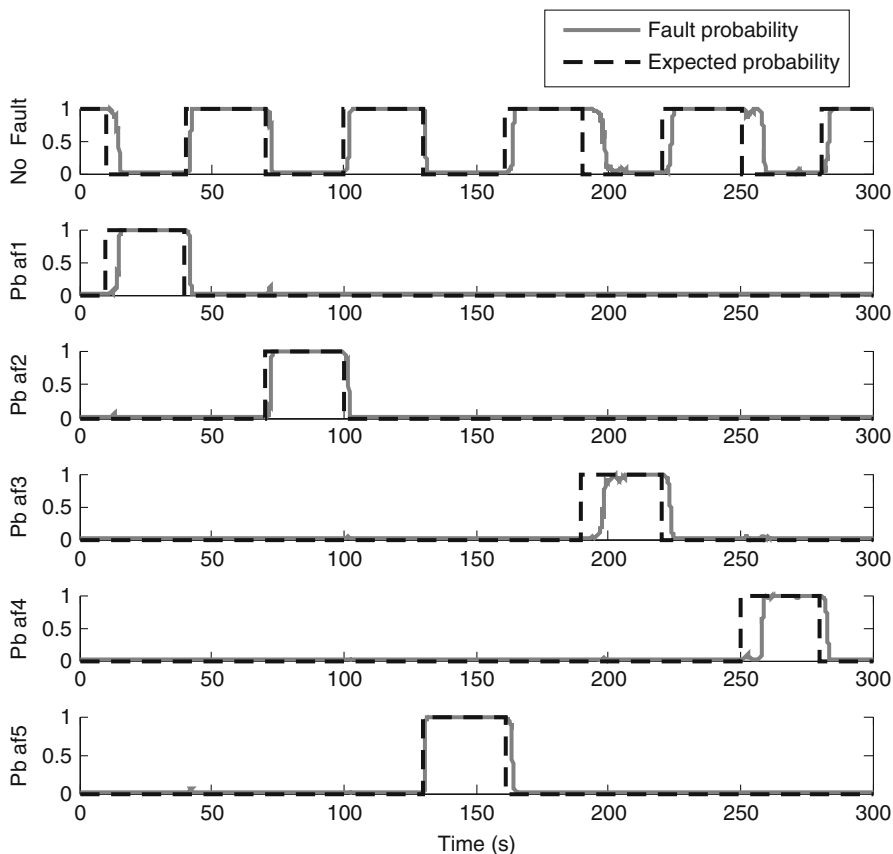


Fig. 44.13 Probabilities from each filter of the EMMAE-FDI after a sequence of faults (with the supervisor)

44.4.10 Computational Complexity of the EMMAE-FDI

Let us denote by N the number of actuators that are monitored by an FDI system. In the classical implementation of an MMAE-FDI, for each actuator, k filters are to be designed for k different possible positions of the failed actuator. Therefore, $kN + 1$ KFs are required (+1 refers to the no-fault filter). If the appearance of a second fault is to be checked as well, a new bank of KFs has to be reloaded, based on the knowledge of the first fault that occurred. In total $Nk + 1 + (N - 1)kN = N^2k + 1$, KFs must be designed.

One major advantage of the EMMAE-FDI method presented in this chapter over classical MMAE schemes is that it requires only one filter to completely monitor one actuator. Any possible actuator-fault scenario is taken into account by only one filter. Therefore, for the monitoring of a single actuator fault, only $N + 1$ filters are

required with the EMMAE-FDI method. For the monitoring of a second actuator fault with the EMMAE-FDI method, no other bank of filters has to be loaded. Indeed, if actuator i fails, it suffices to feed all the other filters with the estimate of the faulty control surface deflection $\hat{\delta}_i$ instead of the input δ_i . Thus, again only $N + 1$ filters are needed with the EMMAE-FDI system to detect and isolate a second fault after a first actuator fault has been introduced.

As a simple example, the UAV described in Sect. 44.2 is equipped with five actuators. A classical MMAE scheme designed for 3 possible faulty deflections per actuator requires 16 filters, whereas the EMMAE method needs only 6 filters for any lock-in-place and floating actuator fault scenarios.

Finally, compared to other works (e.g., Ducard and Geering (2006)) with larger state vectors, the choice of the state vector $\mathbf{x} = [p \ q \ r \ \alpha \ \beta \ \delta]^T$ minimizes the number of relevant state elements for the satisfactory operation of the filters, thus limiting the number of computations required for the EKFs.

44.4.11 Realistic Flight Scenario and Conclusions

More simulation results are reported in Ducard (2009), where the EMMAE FDI system is tested in a realistic flight scenario, in which the UAV takes off, tracks a predefined trajectory, and follows an altitude and a speed reference profile. First, the simulation is done without wind, and in a second phase, Dryden Wind turbulences are included in the simulation to test the behavior of the FDI system in the presence of external disturbances. The speed profile is chosen to cover a significant range of the aircraft speed in order to test the performance of the fault detection system at different operating conditions. The measured airspeed V_T is corrupted by noise, like all the other measurement signals defined in Sect. 44.4.7.1 that are used in the FDI system. Also, the altitude reference signal is chosen such that the aircraft has a typical vertical motion during a short mission.

The results show that the EMMAE-FDI algorithm, combined with an active supervision module, offers fast and accurate fault detection and isolation. Moreover, the addition of the estimation of the faulty control surface deflection in the state vector makes the method applicable for actuator failures such as frozen or floating at an arbitrary position. Only one filter is needed to monitor the health of one actuator. The filters used in the EMMAE-FDI are EKFs, which provide nonlinear state estimations at any flight-operating condition. An active FDI technique is developed, which generates appropriate artificial excitation of the aircraft when needed. In Ducard (2009), an additional filtering stage for the fault-probability signals has been designed to enhance the robustness of the diagnosis, even in the event of severe wind turbulence. The whole system has been demonstrated using nonlinear simulations of a realistic flight scenario. The FDI system was shown to be capable of handling two simultaneous actuator failures without increasing the computational load. Finally, when a fault is clearly isolated, the faulty actuator deflection estimate

can be advantageously used to modify online the settings of a control allocator, making the whole system suitable for flight control reconfiguration without any change in the initial controller and any additional actuator position sensor.

44.5 A Single Model Active (SMAC) FDI System

In both the MMAE and EMMAE methods, actuator fault detection and isolation is done through the computation of fault probabilities. Although the EMMAE is more computationally efficient and reliable than the MMAE method, it still requires a bank of EKF, where each EKF monitors one actuator in the system.

This section presents an FDI system which is single model based and thus does not employ a bank of filters. This FDI system employs a single EKF to generate residuals, which indicate the deviation of the monitored system from its normal behavior (fault detection). The residuals are further used to assess the origin of the system's behavior discrepancy. In order to isolate which actuator experiences a fault, an innovative mechanism that artificially excites actuators has been designed. The method presented next has been named single model active fault detection and isolation (SMAC-FDI) system (Ducard and Geering 2010).

This section is organized in order to (1) present an innovative architecture of an FDI system based on a single filter for the generation of residuals independently on the fault scenario; (2) describe an active strategy that searches in the residuals of this single filter the signature of an actuator fault and to test the actuators in a systematic way; (3) show a new procedure to isolate actuator fault that is not based on residuals manipulations but on the observation of control signals only; (4) show that the SMAC-FDI method is highly computationally efficient and, therefore, can run on microcontrollers with low computational resources; and (5) demonstrate the performance of this simple method in simulation.

Figure 44.14 shows the architecture of the SMAC-FDI system. It is composed of five main subsystems: a residual generator, a fault detector, an excitation signals generator (ESG), an actuator health evaluator (AHE), and a fault isolator. Each of these subsystems is described in the following:

44.5.1 Residual Generator

The residual generator monitors the control signals sent to actuators and the behavior of the aircraft through the reading of selected measurement of the aircraft's dynamics. In the SMAC-FDI system, the turn rates p , q , and r are measured and compared to the expected turn rates \hat{p} , \hat{q} , and \hat{r} , which are obtained using a model of the aircraft. This model receives the same actuator commands as the real aircraft. A filter is used to take into account model uncertainties and sensor measurement data noise. This filter is an EKF, whose useful output – for FDI purpose – is the vector of residuals $\tilde{\omega}_k$ defined as follows:

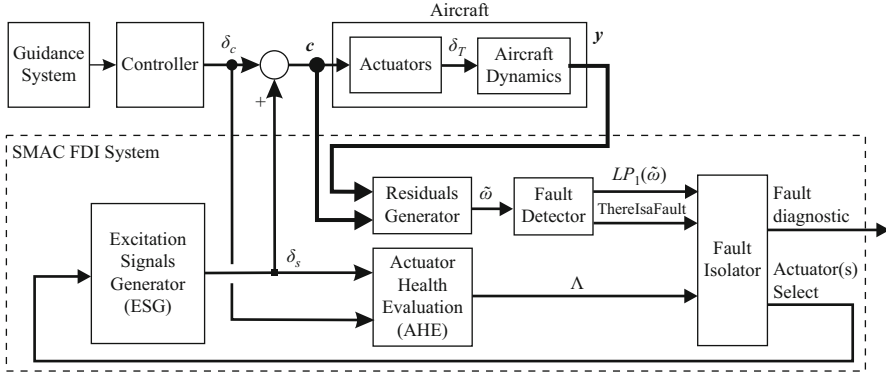


Fig. 44.14 Architecture of the SMAC-FDI system

$$\tilde{\omega}_k = \mathbf{y}_k - \hat{\mathbf{x}}_{k|k-1} = \begin{bmatrix} \tilde{p} \\ \tilde{q} \\ \tilde{r} \end{bmatrix} = \begin{bmatrix} p - \hat{p}_{(k|k-1)} \\ q - \hat{q}_{(k|k-1)} \\ r - \hat{r}_{(k|k-1)} \end{bmatrix}, \quad (44.47)$$

where the measurement vector at the discrete-time instant k is $\mathbf{y}_k = [p \ q \ r]^T$ and the filter's state vector is the estimate of the turn rates. These residuals indicate whether the aircraft has dynamics that deviate from the model-expected dynamics. If the residual values are higher than a certain threshold, it is most likely that an actuator fault is present in the system.

The EKF used in this residual generator has the structure described in Sect. 44.3. The matrices and vectors involved in this filter are described next.

44.5.1.1 Discrete Transition Matrix for the SMAC-FDI System

The discrete transition matrix ϕ_k is computed as $\phi_k = \mathbf{I}_3 + \mathbf{F}(k)T_s$. The continuous transition matrix $\mathbf{F}(k) \in \Re^{3 \times 3}$ is obtained using the model of the aircraft described in Sect. 44.2 and updated at each sample time by the estimates of the turn rates $\hat{p}_{(k|k)}$, $\hat{q}_{(k|k)}$, $\hat{r}_{(k|k)}$ and with the measurement data for the airspeed V_T involved in the dynamic pressure $\bar{q} = \frac{1}{2}\rho V_T^2$ as follows:

$$\mathbf{F}(k) = \begin{bmatrix} A_{11\bar{q}} \bar{q} + A_{11q} q & A_{12p} p + A_{12r} r & A_{13\bar{q}} \bar{q} + A_{13q} q \\ A_{21p} p + A_{21r} r & A_{22\bar{q}} \bar{q} & A_{23p} p + A_{23r} r \\ A_{31\bar{q}} \bar{q} + A_{31q} q & A_{32p} p + A_{32r} r & A_{33\bar{q}} \bar{q} + A_{33q} q \end{bmatrix} \begin{cases} p = \hat{p}_{(k|k)} \\ q = \hat{q}_{(k|k)} \\ r = \hat{r}_{(k|k)} \\ \bar{q} = \bar{q}_{(k)} \end{cases} \quad (44.48)$$

with the terms of the transition matrix defined as

$$\begin{aligned}
 A_{11\bar{q}} &= \frac{I_{zz} S b^2 C_{L\bar{p}}}{2D_1 V_T} & A_{11q} &= A_{12p} = \frac{-N_1}{D_1} & A_{12r} &= A_{13q} = \frac{I_{yy} I_{zz} - I_{xz}^2 - I_{zz}^2}{D_1} \\
 A_{13\bar{q}} &= \frac{S b^2 [I_{zz} C_{L\bar{p}} - I_{xz} C_{N\bar{p}}]}{2D_1 V_T} & A_{21r} &= A_{23p} = \frac{I_{zz} - I_{xx}}{I_{yy}} & A_{21p} &= -A_{23r} = \frac{2I_{xz}}{I_{yy}} \\
 A_{22\bar{q}} &= \frac{S \bar{c}^2 C_{M\bar{q}}}{2V_T I_{yy}} & A_{31q} &= A_{32p} = \frac{I_{xx}^2 - I_{xx} I_{yy} + I_{xz}^2}{D_1} & A_{31\bar{q}} &= -\frac{I_{xz} S b^2 C_{L\bar{p}}}{2D_1 V_T} \\
 A_{32r} &= A_{33q} = \frac{N_1}{D_1} & A_{33\bar{q}} &= \frac{S b^2 [I_{xx} C_{N\bar{p}} - I_{xz} C_{L\bar{p}}]}{2D_1 V_T}
 \end{aligned} \tag{44.49}$$

using the auxiliary constants $D_1 = I_{xx} I_{zz} - I_{xz}^2$ and $N_1 = I_{xz}(I_{xx} - I_{yy} + I_{zz})$.

Remark: In the above equations, the constant terms have been written taking into account the fact that $I_{xz} = I_{zx}$.

44.5.1.2 Process Noise Covariance Matrix for the SMAC-FDI System

The value of $\mathbf{R}_{w,k}$ is found from the continuous process noise covariance matrix \mathbf{R}_w and the continuous transition matrix ϕ as defined in Eq. 44.17, according to

$$\mathbf{R}_{w,k} = \int_0^{T_s} \phi(\tau) \mathbf{R}_w \phi^T(\tau) d\tau \tag{44.50}$$

$$\mathbf{R}_w = E(\mathbf{w}\mathbf{w}^T) = \text{diag}(w_1^2, w_2^2, w_3^2) \tag{44.51}$$

$$\mathbf{w} = [w_1, w_2, w_3]^T, E(w_i w_j) = 0, \forall i \neq j. \tag{44.52}$$

The matrix $\mathbf{R}_{w,k} \in \Re^{3 \times 3}$ is symmetric, and the lower-triangle terms are obtained as follows:

$$\begin{aligned}
 \mathbf{R}_{w,k}(1, 1) &= w_1^2 T_s \left[1 + \phi(1, 1) T_s + \phi(1, 1)^2 \frac{T_s^2}{3} \right] \\
 &\quad + \frac{T_s^3}{3} \left[w_2^2 \phi(1, 2)^2 + w_3^2 \phi(1, 3)^2 \right] \\
 \mathbf{R}_{w,k}(2, 2) &= w_2^2 T_s \left[1 + \phi(2, 2) T_s + \phi(2, 2)^2 \frac{T_s^2}{3} \right] \\
 &\quad + \frac{T_s^3}{3} \left[w_1^2 \phi(2, 1)^2 + w_3^2 \phi(2, 3)^2 \right] \\
 \mathbf{R}_{w,k}(3, 3) &= w_3^2 T_s \left[1 + \phi(3, 3) T_s + \phi(3, 3)^2 \frac{T_s^2}{3} \right] \\
 &\quad + \frac{T_s^3}{3} \left[w_1^2 \phi(3, 1)^2 + w_2^2 \phi(3, 2)^2 \right]
 \end{aligned}$$

$$\begin{aligned}
\mathbf{R}_{w,k}(2, 1) &= A_1\phi(2, 1) + A_2\phi(1, 2) + w_3^2 \frac{T_s^3}{3} \phi(1, 3)\phi(2, 3) \\
\mathbf{R}_{w,k}(3, 1) &= A_1\phi(3, 1) + A_3\phi(1, 3) + w_2^2 \frac{T_s^3}{3} \phi(1, 2)\phi(3, 2) \\
\mathbf{R}_{w,k}(3, 2) &= A_2\phi(2, 1) + A_3\phi(2, 3) + w_1^2 \frac{T_s^3}{3} \phi(2, 1)\phi(3, 1) \quad (44.53)
\end{aligned}$$

with the constants

$$\begin{aligned}
A_1 &= w_1^2 \frac{T_s^2}{2} \left[1 + \frac{2}{3} T_s \phi(1, 1) \right], \quad A_2 = w_2^2 \frac{T_s^2}{2} \left[1 + \frac{2}{3} T_s \phi(2, 2) \right], \\
A_3 &= w_3^2 \frac{T_s^2}{2} \left[1 + \frac{2}{3} T_s \phi(3, 3) \right] \quad (44.54)
\end{aligned}$$

where $\phi(i, j)$ and $\mathbf{R}_{w,k}(i, j)$ designate the term found at the i th line and j th column of the matrix ϕ and matrix $\mathbf{R}_{w,k}$, respectively.

44.5.1.3 State Estimate Propagation Step

The state estimate propagation step is done with $\hat{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k|k} + \dot{\hat{\mathbf{x}}}_{k|k} T_s$, where time derivative of the state $\dot{\hat{\mathbf{x}}}_{k|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k)$ is obtained by direct calculation of (Eq. 44.1).

44.5.1.4 Sensor Noise and Filter Tuning

The measurement vector involved in the EKF of the SMAC-FDI system is $\mathbf{y} = [p, q, r]^T$, where the body-axis turn rates p, q , and r are measured by gyroscopes. The measurement noise is assumed to be Gaussian distributed with zero mean. The noise standard deviations are $\sigma_{pqr} = 0.08$ (rad/s) (variance $\Sigma_{pqr} = 0.0064$ [rad/s]²) for the turn rates; $\sigma_{\alpha, \beta} = 0.0349$ (rad) for the angles α and β ; and $\sigma_{V_T} = 5$ (m/s) for the airspeed measurement.

The EKF is tuned by selecting the measurement noise covariance matrix $\mathbf{R}_v = 0.1 \times \mathbf{I}_3$ (rad/s)² and the process noise covariance matrix $\mathbf{R}_w = 2 \times \mathbf{I}_3$ (rad/s)². The filter is initialized with the initial state estimate $\hat{\mathbf{x}}_{0|-1} = [0 \ 0 \ 0]^T$ and initial state error covariance matrix $\Sigma_{0|-1} = \mathbf{I}_3$.

44.5.2 Fault Detector

44.5.2.1 Architecture of the Fault Detector

The architecture of the fault detector is shown in Fig. 44.15. First, the vector of turn-rate residuals $\tilde{\omega}$ coming from the EKF is filtered by a first-order low-pass filter LP_1 with a cutoff frequency f_1 . Second, the residuals are assembled as follows:

$$\Gamma_1 = |LP_1(\tilde{p})| + |LP_1(\tilde{q})| + |LP_1(\tilde{r})|. \quad (44.55)$$

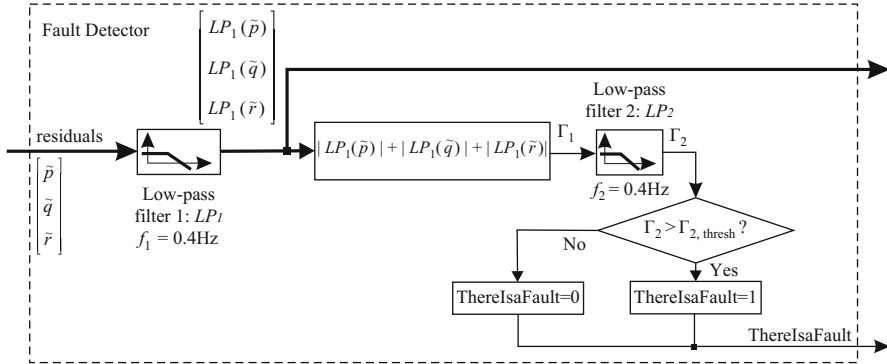


Fig. 44.15 Architecture of the fault detector subsystem

Third, the first-order low-pass filter with cut-off frequency f_2 is used to transform the signal Γ_1 into the signal Γ_2 . Fourth, if the signal Γ_2 exceeds the threshold value $\Gamma_{2, \text{Thresh}}$, the fault detector indicates that there is a fault somewhere in the system by setting to one the output signal labeled “ThereIsaFault” shown in Fig. 44.15. The origin of the fault is not known at this stage. It is the task of the fault isolator subsystem – described below – to investigate the origin of the fault.

Remark: The fault isolator is disabled when the flag $\text{ThereIsaFault}=0$ and enabled when $\text{ThereIsaFault}=1$, thus keeping the total amount of computational load required by the SMAC-FDI system to the minimum needed, as opposed to an MMAE or EMMAE structure in which the whole bank of filters is permanently running.

44.5.2.2 Selection of the Cutoff Frequencies

Since the excitation signals have a fixed frequency of 1 Hz and that the purpose of the low-pass filters is to attenuate these sinusoidal signals in the residuals, the cutoff frequencies f_1 and f_2 are both chosen in the range $]0, 1[\text{ Hz}$. In this work, the chosen value is $f_1 = f_2 = 0.4 \text{ Hz}$, which provides a trade-off between smoothing and delaying the residual signals.

44.5.2.3 Determination of the Threshold Value $\Gamma_{2, \text{thresh}}$

The success of fault detection is dependent on the possibility to create relevant signals that deviate from a nominal value in the case of a fault. In this work, the residuals of the Kalman filter are used and assembled as shown in Fig. 44.15. As soon as the signal named Γ_2 exceeds a certain threshold $\Gamma_{2, \text{thresh}}$, this indicates that a fault is present somewhere in the system.

In order to build these residuals, the Kalman filter is using a model of the aircraft’s turn-rate dynamics. However, this model contains parameters which are uncertain. In order to cope with uncertainties, the Kalman filter possesses intrinsically a mechanism to compute the observer gain matrix L in (Eq. 44.12) that mitigates the confidence level between the quality of the model (through the process

noise covariance matrix \mathbf{R}_w) and the quality of the measurement data (through the measurement noise covariance matrix \mathbf{R}_v).

For the determination of a suitable value for $\Gamma_{2,\text{thresh}}$, a linear SISO model is constructed for the roll, pitch, and yaw axis as follows:

$$\dot{\chi} = (\hat{a}_\chi + \Delta a_\chi) \chi + (\hat{b}_\chi + \Delta b_\chi) \delta_\chi, \quad (44.56)$$

where the variable χ is p, q, r , respectively, and the real control input to the system δ_χ – that may include a fault – is $\delta_a, \delta_e, \delta_r$, respectively. Note that (Eq. 44.56) is used as a template for three different equations.

The terms \hat{a}_χ and \hat{b}_χ represent the uncertain model parameters in use in the FDI system. They are made of the sum of the true value a_χ, b_χ and the uncertainty part Δa_χ and Δb_χ , respectively, as follows:

$$a_\chi = \hat{a}_\chi + \Delta a_\chi \quad b_\chi = \hat{b}_\chi + \Delta b_\chi. \quad (44.57)$$

For analysis purpose, let us consider a linear Kalman-type estimator of the form:

$$\dot{\hat{\chi}} = \hat{a}_\chi \hat{\chi} + \hat{b}_\chi \delta_{\chi,c} + L_\chi (\chi - \hat{\chi}), \quad (44.58)$$

where the Kalman gain is L_χ and the control signal generated by the flight controller is $\delta_{\chi,c}$. By subtracting (Eq. 44.58) to (Eq. 44.56), the differential equation for the residuals of turn-rate dynamics is obtained as follows:

$$\dot{\tilde{\chi}} = -(L_\chi - \hat{a}_\chi) \tilde{\chi} + \hat{b}_\chi \Delta \delta_\chi + \Delta \epsilon(\chi(t), x(t)), \quad (44.59)$$

where the term $\Delta \delta_\chi = \delta_\chi - \delta_{\chi,c}$ is the deviation between a true actuator position and its commanded value. All of the uncertainty contributions are regrouped in the term $\Delta \epsilon(\chi(t), x(t))$, which may be dependent on the turn rate itself $\chi(t)$ and other state elements, such as angle of attack α , sideslip angle β , and aircraft total airspeed V_T .

The stability of the filter is obtained if the condition

$$L_\chi - \hat{a}_\chi > 0 \quad (44.60)$$

is satisfied, in which case the residual $\tilde{\chi}$ converges to a certain value if the system approaches steady-state conditions. The condition in (Eq. 44.60) is fulfilled by choosing the terms in the process noise covariance matrix \mathbf{R}_w sufficiently large.

In the presence of model uncertainties or actuator fault, the residuals will no longer be zero-mean centered. This is precisely what will trigger the detection of a fault in the system. It is therefore necessary to be able to assess the amount of bias in the residuals that is due to model uncertainties that are not zero-mean centered.

To this end, suppose that an actuator failure occurs, the flight controller will compensate by actuating the remaining actuators in order to stabilize the turn rates of the vehicle, that is, $p = q = r = 0$. Moreover, if the vehicle is flying straight,

level and at constant speed, the terms $\Delta\delta_\chi$ and $\Delta\epsilon$ will reach a constant value $\Delta\delta_{\chi,ss}$ and $\Delta\epsilon_{ss}$. Therefore, the residual in (Eq. 44.59) reaches the following steady-state value:

$$\tilde{\chi}_{ss} = \frac{\hat{b}_\chi \Delta\delta_{\chi,ss} + \Delta\epsilon_{ss}}{L_\chi - \hat{a}_\chi} = \frac{\hat{b}_\chi (\Delta\delta_{\chi,ss} + \Delta\gamma_{ss})}{L_\chi - \hat{a}_\chi}, \quad \text{with } \Delta\gamma_{ss} = \frac{\Delta\epsilon_{ss}}{\hat{b}_\chi}. \quad (44.61)$$

Once the maximum amount of uncertainty in the model, $\sup |\Delta\epsilon|$, is specified, the threshold value of Γ_2 is chosen according to the following inequality:

$$\frac{\hat{b}_\chi \sup |\Delta\gamma|}{L_\chi - \hat{a}_\chi} < \Gamma_{2, \text{thresh}} < \frac{\hat{b}_\chi (|\Delta\delta_{\chi, \min}| - \sup |\Delta\gamma|)}{L_\chi - \hat{a}_\chi}. \quad (44.62)$$

Remarks:

- The results shown in Sect. 44.5.6 are obtained with the value of $\Gamma_{2, \text{thresh}} = 0.05$ rad/s. But note that a value of $\Gamma_{2, \text{thresh}}$ can be computed online and therefore be adapted to the current flight conditions.
- If the value of the true model term b_χ is known within some uncertainty bounds $\hat{b}_\chi = b_\chi \pm \Delta b_\chi$, it is better to choose in the FDI system an overestimated value for \hat{b}_χ , that is, $\hat{b}_\chi = b_\chi + |\Delta b_\chi|$ as this tends to increase the detectable fault level to uncertainty level ratio.

44.5.2.4 Effect of Uncertainties on Actuator Fault Detection Performance

The effect of model uncertainties on the minimum detectable value of the actuator fault $\Delta\delta_{\chi, \min}$ is visible in (Eq. 44.62), where it appears that a lower bound for $\Delta\delta_{\chi, \min}$ given the maximum uncertainty level $\sup |\Delta\gamma|$ is

$$|\Delta\delta_{\chi, \min}| > 2 \sup |\Delta\gamma|. \quad (44.63)$$

Clearly, the less uncertainty there is in the model, the smaller the fault that can be detected.

44.5.3 Excitation Signals Generator (ESG)

As shown in Fig. 44.14, the ESG “listens” to the fault isolator that selects which actuator to test. The ESG superimposes on that actuator’s command an artificial excitation signal of the form:

$$\delta_s(t) = A \sin(\omega t). \quad (44.64)$$

The frequency ω has to be chosen within the bandwidth of the aircraft under consideration. Indeed, as explained in Sect. 44.5.4, the excitation signal must be

of sufficiently low frequency such that the aircraft dynamics get disturbed by this excitation signal. As a result, the flight controller will generate actuator commands to compensate this perturbation. If the presence of this excitation signal is visible in the actuator control signals, it means the tested actuator has actually not failed, and vice versa.

In order to discuss a suitable value for the amplitude A , let us consider the dynamics of the residuals when the signal δ_s is applied to a failed actuator. The dynamics of the residual vector $\tilde{\chi}$ is as follows:

$$\dot{\tilde{\chi}} = -(L_\chi - \hat{a}_\chi)\tilde{\chi} + \hat{b}_\chi (\Delta\delta_\chi - A \sin \omega t + \Delta\gamma). \quad (44.65)$$

It can be shown that the asymptotic solution to (Eq. 44.65) is

$$\tilde{\chi}_\infty = C_1 - C_2 \sin(\omega t - \phi_0), \quad (44.66)$$

with the terms $C_1 = \frac{\hat{b}_\chi}{L_\chi - \hat{a}_\chi} (\Delta\delta_\chi + \Delta\gamma)$, $C_2 = \frac{\hat{b}_\chi A}{\sqrt{\omega^2 + (L_\chi - \hat{a}_\chi)^2}}$, and $\phi_0 = \tan^{-1} \left(\frac{\omega}{L_\chi - \hat{a}_\chi} \right)$.

In the presence of an actuator fault in the system, it is shown in Fig. 44.15 that the signal Γ_2 must exceed a certain threshold $\Gamma_{2,\text{thresh}}$ for the fault to be detected. The signal Γ_2 is obtained by low-pass filtering the signal Γ_1 . In order to simplify a bit the discussion, let us assume that the signal $\Gamma_1 = |\tilde{\chi}|$, which is the case when the residuals build up dominantly on a certain aircraft axis.

In view of (Eq. 44.66), it can be shown that the mean value of Γ_2 is mostly given by the “constant” part of $|\tilde{\chi}|$, that is, C_1 . As soon as the excitation signal is triggered, additional calculations show that whatever the amplitude of C_2 compared to C_1 , ($C_2 > C_1$ or $C_2 < C_1$), the condition $\Gamma_2 > \Gamma_{2,\text{thresh}}$ which triggered the excitation signal still remains satisfied after the addition of excitation signals. In fact, if the term C_1 is smaller than C_2 , the excitation term has the positive effect of contributing to an higher value of Γ_2 .

Therefore, the amplitude A is chosen as a compromise between (1) a value of A as large as possible such that the fault isolation explained in Sect. 44.5.4 can maximize the amplitude of the isolation signals Λ_i , despite the possible presence of external perturbation, such as wind gusts or aircraft maneuvers, thus improving FDI robustness; and (2) a value of A sufficiently small, not to unnecessarily destabilize the aircraft as it flies, if the excitation signal is applied to a non-failed actuator.

Remark: In Sect. 44.5.6, the results shown are obtained with a frequency $\omega = 2\pi$ (rad/s) and an amplitude $A = 2 \frac{\pi}{180}$ (rad).

44.5.4 Actuator Health Evaluator (AHE)

The actuator health evaluator (AHE) subsystem is responsible for assessing whether a selected actuator δ_i is properly functioning or not. As shown in Fig. 44.14,

the AHE module receives the control signals $\delta_c = [\delta_{c,1} \dots \delta_{c,n}]$ of n actuators generated by the flight controller and the excitation signals $\delta_s = [\delta_{s,1} \dots \delta_{s,n}]$ generated by the ESG module.

For the i th actuator, the output of the AHE is given by the following equation:

$$\Lambda_i(k) = LP_{i2} \left(|\delta_{s,i}(k - \lambda T_s) + [\delta_{c,i} - \bar{\delta}_{c,i}](k)| \right), \quad (44.67)$$

where $LP_{i2}(X)$ means that the signal X is filtered through a first-order low-pass filter $\frac{1}{1+\tau_{i2}s}$ with cutoff frequency $f_{i2} = \frac{1}{2\pi\tau_{i2}}$ (Hz). The choice of the frequency f_{i2} is not really important. This low-pass filter is only meant to attenuate the noise level of its incoming signal. In steady-state conditions, the signal $\bar{\delta}_{c,i}$ equals the mean value of $\delta_{c,i}$. The excitation signal $\delta_{s,i}$ is delayed by an integer number of sampling periods λT_s , $\lambda \in N^+$, such that the signals $[\delta_{c,i} - \bar{\delta}_{c,i}](k)$ and $\delta_{s,i}(k - \lambda T_s)$ are in opposite phase when actuator i is non-failed. The delay λ is determined experimentally and corresponds to the delay of the flight controller to react to the disturbance caused by $\delta_{s,i}$ superimposed onto actuator i when non-failed.

44.5.4.1 Case 1: Testing a Failed Actuator

If the actuator i being tested has really failed, the excitation signal $\delta_{s,i}$ has no effect on the control signal $\delta_{c,i}$ but has an effect on the residuals generated by the residual generator. The AHE detects the nonresponse of the flight controller to the excitation $\delta_{s,i}$ and thus generates the signal Λ_i whose amplitude is compared to a threshold value (Ail1FaultThresh, Ail2FaultThresh, Elev1FaultThresh, Elev2FaultThresh, and RuddFaultThresh, respectively) inside the fault isolator module. If Λ_i is above the threshold value, the actuator i is declared faulty, as shown in Figs. 44.21 and 44.19.

44.5.4.2 Case 2: Testing a Non-failed Actuator

If the actuator i has actually not failed, the superimposed excitation signal $\delta_{s,i}$ has an influence of the aircraft dynamics, and the flight controller will try to compensate the disturbance input signal. However, there is a small delay in the controller response. The term $\delta_{s,i}$ is delayed in such a way to be in opposite phase with the term $[\delta_{c,i} - \bar{\delta}_{c,i}]$. Therefore, their sum Λ_i is close to zero and below the threshold value (Ail1FaultThresh, Ail2FaultThresh, Elev1FaultThresh, Elev2FaultThresh, and RuddFaultThresh, respectively) in the fault isolator module. The actuator i is thus declared non-faulty, as shown in Figs. 44.21 and 44.19.

44.5.4.3 Tuning of the AHE

The tuning of the AHE subsystem consists in finding the appropriate cutoff frequency noted f_{i1} as shown in Fig. 44.16. The frequency f_{i1} is chosen in such a way that the term $\bar{\delta}_{c,i}$ converges to the mean value of $\delta_{c,i}$: $E(\delta_{c,i})$ after a time interval ΔT :

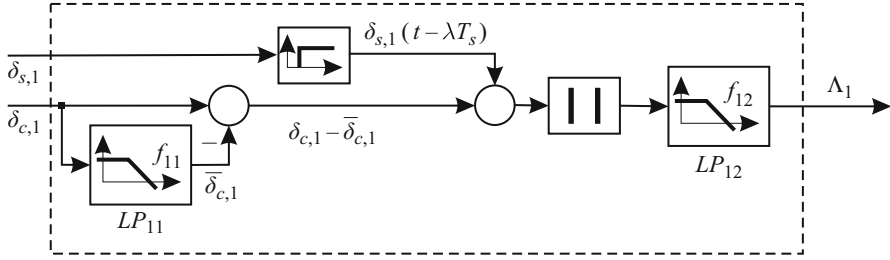


Fig. 44.16 Architecture of the actuator health evaluator, AHE subsystem (example of actuator 1)

$$\left| \frac{E(\delta_{c,i}) - \bar{\delta}_{c,i}(t > \Delta T)}{E(\delta_{c,i})} \right| < \epsilon, \quad (44.68)$$

with the relative error $\epsilon = 5\%$, for example. The frequency f_{i1} is chosen as

$$f_{i1} > \frac{\ln(1/\epsilon)}{2\pi\Delta T}. \quad (44.69)$$

44.5.4.4 Selection of the Threshold Values for Actuator Fault Diagnosis

The threshold values (Ail1FaultThresh, Ail2FaultThresh, Elev1FaultThresh, Elev2FaultThresh, RuddFaultThresh) are used inside the fault isolator module to declare an actuator to have failed.

The signal Λ_i at the output of the i th AHE subsystem has a value close to zero in the non-fault case and a maximum value of

$$\Lambda_{i,\max} = \frac{2A}{\pi}, \quad (44.70)$$

if actuator i experiences a complete failure case. If actuator i experiences a fault, such as loss of effectiveness, the signal Λ_i takes an intermediate value within the range $0 < \Lambda_i < \frac{2A}{\pi}$.

Example: The simulations provided in Sect. 44.5.6 have been performed with $A = 2\frac{\pi}{180}$ (rad). According to (Eq. 44.70), during the complete failure of aileron1 and aileron2, the values of Λ_1 and Λ_2 should be 0.022 (rad), which is confirmed by the results shown in Fig. 44.21. Therefore, the threshold values for actuator fault diagnosis have been chosen to be $\Lambda_{i,\text{thresh}} = 0.01$ (rad).

44.5.5 Fault Isolator

The complete architecture of the fault isolator is shown in Fig. 44.17. It manipulates the residuals in order to choose which actuator group to test (aileron group, elevator group, or rudder group). In each of these groups, the fault isolator selects an

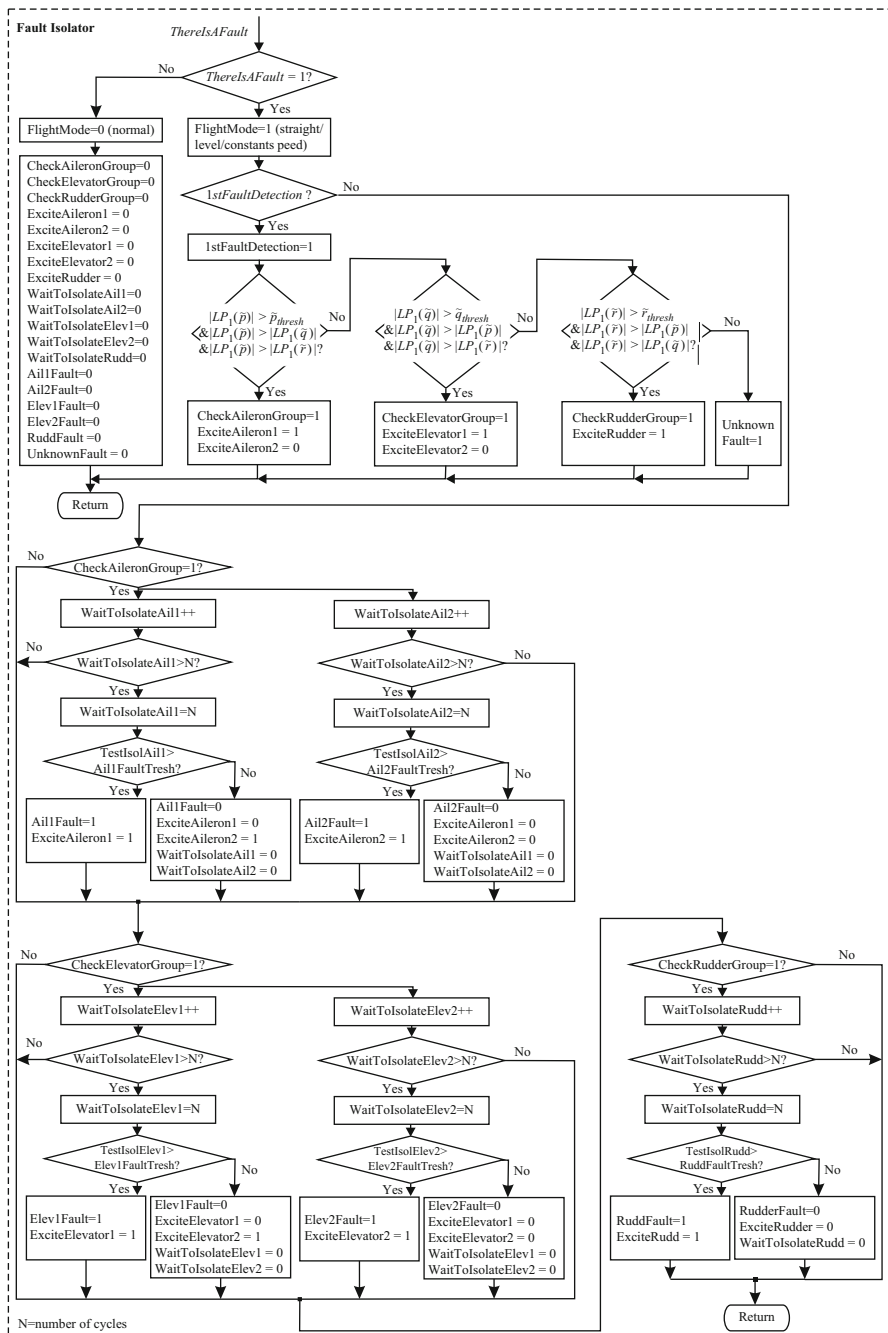


Fig. 44.17 Architecture of the fault isolator subsystem

actuator to test. It is the responsibility of the ESG subsystem shown in Fig. 44.14 to generate the artificial excitation signals to superimpose on the selected actuator. The threshold values $\tilde{p}_{\text{thresh}}$, $\tilde{q}_{\text{thresh}}$, and $\tilde{r}_{\text{thresh}}$ that appear in Fig. 44.17 are chosen according to (Eq. 44.62) for each axis.

44.5.6 Simulations

The simulations are done in Matlab/Simulink using a nonlinear 6 degree-of-freedom model of an aircraft controlled by its autopilot described in Ducard (2009). The aircraft is flying straight, level, and at a constant speed of 30 m/s. These conditions correspond to difficult circumstances for an FDI system due to the lack of excitation. The actuator fault scenario consists of the failure of aileron 1 at a small deflection of 1° during the time interval [25 ... 50 s], the failure of aileron 2 that gets stuck at 2° deg during [75 ... 100 s], and finally aileron 1 is stuck again at 1 deg during [125 ... 175 s].

Simulations are carried out with Gaussian sensor noise with zero mean and a standard deviation of (1) $\sigma_{pqr} = 0.08$ (rad/s) for the body-axis turn rates p , q , and r ; (2) $\sigma_{V_T} = 5$ (m/s) for the aircraft total airspeed V_T ; and (3) $\sigma_\alpha = \sigma_\beta = 0.0349$ (rad/s) for the measurement data of the angle of attack α and the sideslip angle β . Moreover, uncertainties – in the aircraft model used in the prediction step of the EKF in the SMAC-FDI system – have been introduced in the form of reducing by 25 % the values of the torque vector $[L, M, N]^T$ involved in (Eq. 44.2). This is similar to taking an underestimated value of the model input coefficient \hat{b}_χ used in Sect. 44.5.2.3, and it corresponds to a worse case scenario in terms of FDI performance than if the coefficient \hat{b}_χ had been overestimated, as explained in the Remarks of Sect. 44.5.2.3.

This simulation scenario is chosen to show the capability of the SMAC-FDI system to (1) detect actuator failure even if the angle of the stuck control surface is very small and (2) distinguish the precise location of an actuator failure despite the ambiguity that may arise due to actuator redundancies in the system. It is indeed shown in Ducard and Geering (2008) how, for example, an MMAE or EMMAE-FDI scheme performs poorly to isolate a failed actuator between two actuators having the same aerodynamic effect on the aircraft.

The SMAC-FDI is run in simulation at a rate of 50 Hz; therefore, the strategy shown in Fig. 44.17 has a cycle period of 20 ms. Figure 44.18 shows that the SMAC-FDI detects the presence of the fault after 1 s. Figures 44.19 shows that the isolation of a faulty actuator takes about 1.8 s and the detection of a fault removal is detected in both cases in less than 1.2 s. Figure 44.20 and 44.21 show the actuator control signals and results of the AHE subsystem, respectively.

Note that real external perturbations, like wind gusts, may shortly contribute to generate residuals and potentially could trigger a false alarm. However, the SMAC-FDI is architected to systematically excite the actuator – thought to have failed – and eventually conclude whether it has really failed.

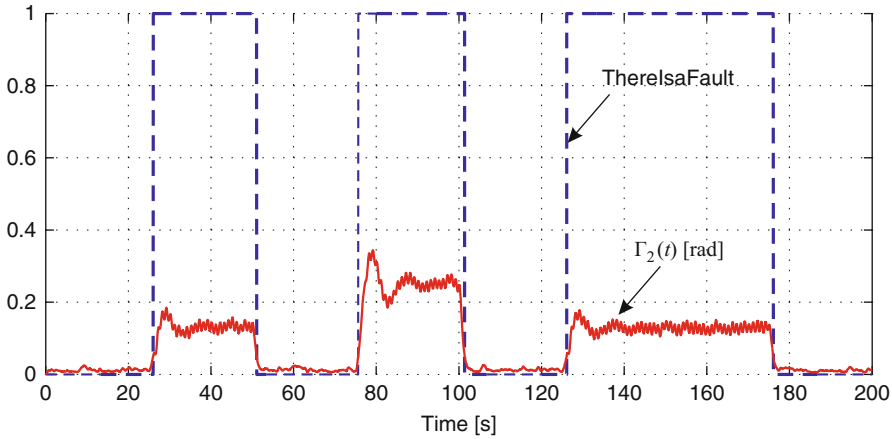


Fig. 44.18 Results of the fault detector: variable *ThereIsaFault* in blue and Γ_2 in red

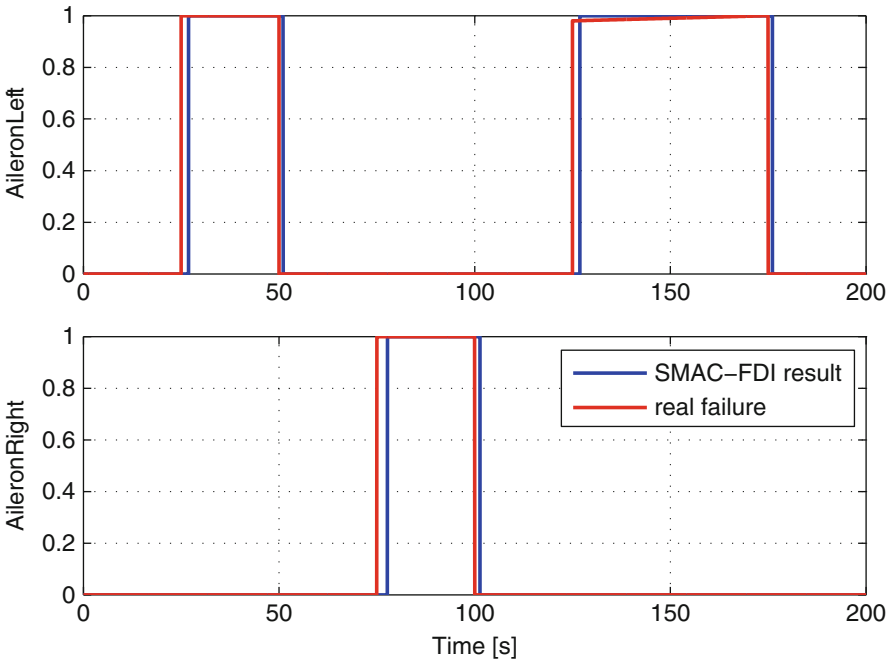


Fig. 44.19 Diagnostic results of the fault isolator: 0 means no fault; 1 means actuator fault

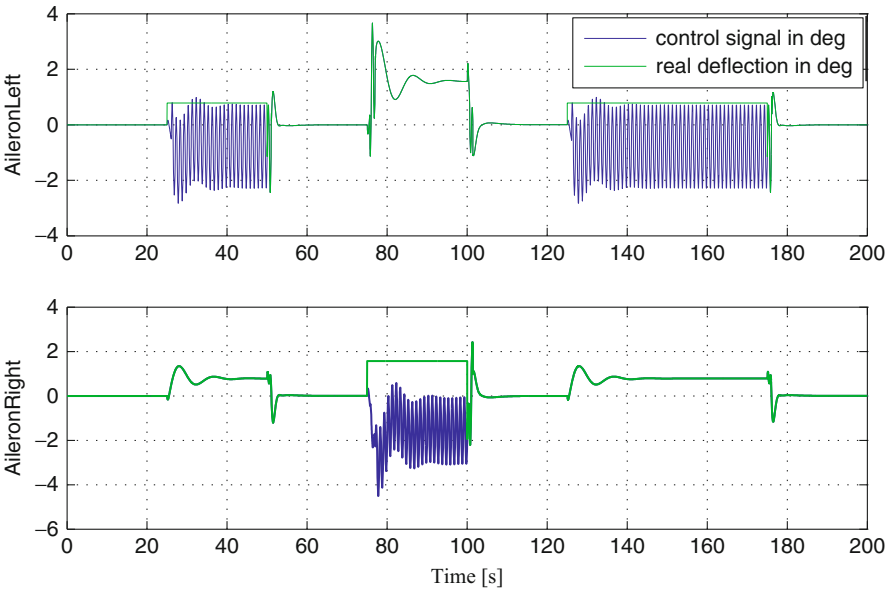


Fig. 44.20 Control signals of the two ailerons during the sequence of failures

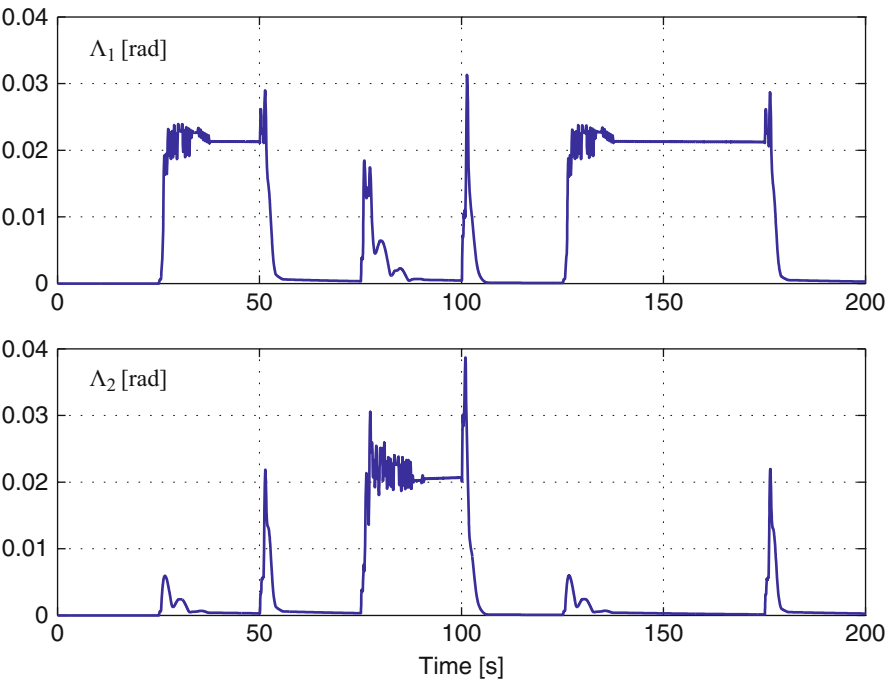


Fig. 44.21 Results of the actuator health evaluator subsystem

44.5.7 Properties of the SMAC-FDI System

The benefits of the SMAC-FDI system are the following:

- It is able to detect the presence of an actuator fault.
- Only 1 EKF for residual generation is used with only three-state elements p, q, r . It is therefore of low computational load (no need for a bank of KFs or EKFs).
- The fault isolation is based on an active and systematic actuator-testing procedure, thus reducing isolation latencies, removing isolation ambiguities among equivalent actuators, and increasing the robustness of this FDI method against external perturbations and model uncertainties.
- The SMAC-FDI system is scalable to a larger number of actuators and can deal with simultaneous faults.
- It is easy to tune.
- It does not require any additional sensor.
- The algorithm does not disturb the flight (the aircraft is not destabilized during actuator testing, failed or not).
- The isolation process is insensitive to the control signals amplitude, contrary to methods based on residuals.
- The isolation process is insensitive to model uncertainties, since it is not model based.
- The SMAC-FDI system can also track a fault once detected even with low natural excitation of the aircraft.
- The SMAC-FDI system is capable of detecting and isolating multiple actuator failures, provided that the logic in the fault isolator is slightly modified to search for the presence of a subsequent fault in the system.

44.5.8 Computational Load Requirements

The SMAC-FDI algorithm automatically reduces to the minimum amount of computation required, since parts of the algorithm are only activated if the presence of a fault has been detected. With the EMMAE method described in Ducard and Geering (2008), a bank of $N + 1$ EKF with at least six states is constantly running to monitor N actuators. The MMAE method is even more computationally intense. In the SMAC-FDI, the computational load is considerably reduced since only one EKF with only three states is constantly running. Therefore, the SMAC-FDI is a very fast method and well suited for a small microcontroller.

44.5.9 Conclusions About the SMAC-FDI System

The SMAC-FDI system is a computationally efficient system to detect and isolate actuator fault (FDI system) with application to an aircraft. This FDI system is based on a single-model-based filter that generates residuals as soon as the behavior of the aircraft deviates from expected. If a fault is detected, the guidance system of the aircraft is reconfigured to fly the aircraft straight and level. A procedure is

engaged to systematically excite each suspected actuator and decide whether it is failed. The residuals of a three-state only EKF are used to detect the presence of a fault in the system, whereas the isolation process is done by observing the control signals. The system is simple to implement and tune, is robust against model uncertainties and external perturbation, does not require additional sensor, and shows good performance in simulation.

44.6 Chapter General Conclusion

In this chapter, some concepts of fault detection and isolation with application to unmanned aircraft have been presented. The fundamental part in an FDI system is the generation of signal(s) that indicate the presence of a fault in the system.

This chapter focused on using residuals calculated by Kalman-type filters that predict the dynamics of the aircraft through a model. The mathematical aircraft model used and the equations of extended kalman filters have been recalled. Two FDI systems have been detailed. The first uses a multiple-model approach and estimates the position of a faulty actuator. The second approach is less computationally intense as it relies on a unique filter to generate residuals for fault detection, and fault isolation is carried out with an actuator-testing procedure. The chapter contains discussions about dealing with model uncertainties, techniques to enhance the robustness of the fault diagnosis, and finally about computational load.

Appendix

Table 44.2 Aircraft model parameters

| Parameter | Value | Unit | Definition |
|----------------------|--|----------------------|-------------------------|
| I^b | $I^b = \begin{bmatrix} 2.56 & 0 & 0.5 \\ 0 & 10.9 & 0 \\ 0.5 & 0 & 11.3 \end{bmatrix}$ | [kg·m ²] | Airplane inertia matrix |
| S | 1.80 | [m ²] | Wing surface |
| \bar{c} | 0.58 | [m] | Mean aerodynamic Chord |
| b | 3.1 | [m] | Wing span |
| C_{Z1} | 1.29×10^{-2} | [—] | Lift derivative |
| $C_{Z\alpha}$ | −3.25 | [—] | Lift derivative |
| C_{X1} | -2.12×10^{-2} | [—] | Drag derivative |
| $C_{X\alpha}$ | -2.66×10^{-2} | [—] | Drag derivative |
| C_{Y1} | -3.79×10^{-1} | [—] | Side force derivative |
| $C_{La1} = -C_{La2}$ | -3.395×10^{-2} | [—] | Roll derivative |
| $C_{Le1} = -C_{Le2}$ | -0.485×10^{-2} | [—] | Roll derivative |
| $C_{L\beta}$ | -1.30×10^{-2} | [—] | Roll derivative |

Table 44.2 (continued)

| Parameter | Value | Unit | Definition |
|---------------------|------------------------|------|------------------|
| $C_{L\bar{p}}$ | -1.92×10^{-1} | [-] | Roll derivative |
| $C_{L\bar{r}}$ | 3.61×10^{-2} | [-] | Roll derivative |
| C_{M1} | 2.08×10^{-2} | [-] | Pitch derivative |
| C_{Me} | 5.45×10^{-1} | [-] | Pitch derivative |
| $C_{Me1} = C_{Me2}$ | 2.725×10^{-1} | [-] | Pitch derivative |
| $C_{Ma1} = C_{Ma2}$ | 0.389×10^{-1} | [-] | Pitch derivative |
| $C_{M\alpha}$ | -9.03×10^{-2} | [-] | Pitch derivative |
| $C_{M\bar{q}}$ | -9.83 | [-] | Pitch derivative |
| $C_{N\delta_r}$ | 5.34×10^{-2} | [-] | Yaw derivative |
| $C_{N\beta}$ | 8.67×10^{-2} | [-] | Yaw derivative |
| $C_{N\bar{r}}$ | -2.14×10^{-1} | [-] | Yaw derivative |

References

- R.J. Adams, *Robust Multivariable Flight Control* (Springer, London/New York, 1994)
- M. Azam, K. Pattipati, J. Allanach, S. Poll, A. Petterson-Hine, In-flight fault detection and isolation in aircraft flight control systems, in *Proceedings of IEEE Aerospace Conference, paper 1429*, (BigSky, MT, 2005)
- F. Bateman, H. Noura, M. Ouladsine, Active fault diagnosis and major actuator failure accommodation: application to a UAV, in *Advances in Flight Control Systems*, ed. by A. Balint (InTech, Rijeka, 2011), pp. 137–158
- C. Belcastro, B.-C. Chang, Uncertainty modeling for robustness analysis of failure detection and accommodation systems, in *Proceedings of the IEEE American Control Conference*, Anchorage, 2002, pp. 4776–4782
- M. Bodson, An adaptive algorithm with information-dependant data forgetting, in *Proceedings of the IEEE American Control Conference*, Seattle, WA, 1995, pp. 3485–3489
- J. Boskovic, R. Mehra, Failure detection, identification and reconfiguration in flight control, in *Fault Diagnosis and Fault Tolerance for Mechatronic Systems: Recent Advances*, ed. by F. Caccavale, L. Villani. Springer Tracts in Advanced Robotics, vol. 1 (Springer, Berlin/Heidelberg, 2003), pp. 129–167. 10.1007/3-540-45737-2-5
- J.D. Boskovic, S.E. Bergstrom, R.K. Mehra, Robust integrated flight control design under failures, damage, and state-dependant disturbances. *AIAA J. Guid. Control Dyn.* **28**(5), 902–917 (2005)
- J.D. Boskovic, J. Redding, R.K. Mehra, Stable adaptive reconfigurable flight control with self-diagnostics, in *Proceedings of the IEEE American Control Conference*, New York, 2007, pp. 5765–5770
- J. Brinker, K.A. Wise, Flight testing of a reconfigurable flight control law on the X-36 tailless fighter aircraft, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, 2000
- R.G. Brown, P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering* (Wiley, New York, 1997)
- S. Brunke, M. Campbell, Estimation architecture for future autonomous vehicles, in *Proceedings of the IEEE American Control Conference*, Anchorage, 2002, pp. 1108–1114
- J. Buffington, P. Chandler, M. Pachter, On-line identification for aircraft with distributed control effectors. *AIAA J. Guid. Control Dyn.* **9**, 1033–1049 (1999)

- A.J. Calise, S. Lee, M. Sharma, Direct adaptive reconfigurable control of a tailless fighter aircraft, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Boston, MA, 1998
- M.E. Campbell, J.W. Lee, E. Scholte, D. RathBun, Simulation and flight test of autonomous aircraft estimation, planning, and control algorithms. *AIAA J. Guid. Control Dyn.* **30**(6), 1597–1609 (2007)
- J. Chen, R.J. Patton, *Robust Model Based Diagnosis for Dynamic Systems* (Kluwer, Dordrecht, 1999)
- J. Chen, R.J. Patton, H. Zhang, Design of unknown input observers and robust fault detection filters. *Int. J. Control* **63**(1), 85–105 (1996)
- E.Y. Chow, A.S. Willsky, Analytical redundancy and the design of robust detection systems. *IEEE Trans. Autom. Control* **29**(7), 603–614 (1984)
- E.G. Collins, T. Song, Robust H_∞ estimation and fault detection of uncertain dynamic systems. *AIAA J. Guid. Control Dyn.* **23**(5), 857–864 (2000)
- G.J.J. Ducard, *Fault-Tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle*. Ph.D. thesis, ETH Zürich, 2007 Diss. No. 17505
- G. Ducard, *Fault-Tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles* (Springer, London, 2009). ISBN:978-1-84882-560-4
- G. Ducard, H.P. Geering, A reconfigurable flight control system based on the EMMAE method, in *Proceedings of the IEEE American Control Conference*, Minneapolis, MN, 2006, pp. 5499–5504
- G. Ducard, H.P. Geering, Efficient nonlinear actuator fault detection and isolation system for unmanned aerial vehicles. *AIAA J. Guid. Control Dyn.* **31**(1), 225–237 (2008)
- G. Ducard, H.P. Geering, SMAC-FDI: new active fault detection and isolation scheme with high computational efficiency, in *Proceedings of the IEEE 2010 Conference on Control and Fault Tolerant Systems*, Nice, France, 2010, pp. 30–37
- P. Eide, P.S. Maybeck, An MMAE failure detection system for the F-16. *IEEE Trans. Aerosp. Electron. Syst.* **32**(3), 1125–1136 (1996)
- M. Elgersma, S. Glavaski, Reconfigurable control for active management of aircraft system failures, in *Proceedings of IEEE American Control Conference*, Arlington, VA, 2001, pp. 2627–2639
- M. Elgersma, D. Enns, S. Shald, P. Voulgaris, Parameter identification for systems with redundant actuators, in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Boston, MA, 1998
- S. Fekri, M. Athans, A. Pascoal, Issues, progress and new results in robust adaptive control. *Int. J. Adapt. Control Signal Process.* **20**(10), 519–579 (2006)
- P. Frank, Enhancement of robustness in observer-based fault detection. *Int. J. Control* **59**(4), 955–984 (1994)
- J.J. Gertler, Fault detection and isolation using parity relations. *Control Eng. Pract.* **5**(5), 653–661 (1997)
- C. Hajiye, F. Caliskan, *Fault-Diagnosis and Reconfiguration in Flight Control Systems* (Kluwer Academic Publishers, Dordrecht, 2003). ISBN:978-1-4020-7605-3
- R. Isermann, *Fault-Diagnosis Systems, An Introduction from Fault Detection to Fault Tolerance* (Springer, Berlin/Heidelberg, 2006)
- S. Julier, J. Uhlmann, H.F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control* **45**(3), 477–482 (2000)
- R.E. Kalman, A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(1), 35–46 (1960)
- B.H. Koh, Z. Li, P. Dharap, S. Nagarajaiah, M.Q. Phan, Actuator failure detection through interaction matrix formulation. *AIAA J. Guid. Control Dyn.* **28**(5), 895–901 (2005)
- D.T. Magill, Optimal adaptive estimation of sampled stochastic processes. *IEEE Trans. Autom. Control* **10**(4), 434–439 (1965)

- A. Marcos, S. Ganguli, G.J. Balas, An application of H_∞ fault detection and isolation to a transport aircraft. *Control Eng. Pract.* **13**, 105–119 (2005)
- P.S. Maybeck, (1994). *Stochastic Models, Estimation, and Control, Volume 1* (Academic, New York, Inc, 1979); republished by Navtech, Arlington, VA, 1994
- P.S. Maybeck, Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems. *Int. J. Robust Nonlinear Control* **9**(14), 1051–1070 (1999)
- P.S. Maybeck, R.D. Stevens, Reconfigurable flight control via multiple model adaptive control methods. *IEEE Trans. Aerosp. Electron. Syst.* **27**(3), 470–479 (1991)
- M. Möckli, *Guidance and Control for Aerobatic Maneuvers of an Unmanned Airplane*. Ph.D. thesis, ETH Zurich, 2006. Diss No. 16586
- L. Ni, *Fault-Tolerant Control of Unmanned Underwater Vehicles*. Ph.D. thesis, VA Tech. Univ., Blacksburg, VA, 2001
- R.J. Patton, J. Chen, Observer-based fault detection and isolation: robustness and applications. *Control Eng. Pract.* **5**(5), 671–682 (1997)
- R.J. Patton, P.M. Frank, R.N. Clark, *Fault Diagnosis in Dynamic Systems: Theory and Applications* (Prentice-Hall, Englewood Cliffs, 1989)
- R.J. Patton, P.M. Frank, R.N. Clark, *Issues of Fault Diagnosis for Dynamic Systems* (Springer, London, 2000)
- R.J. Patton, F.J. Uppal, S. Simani, B. Polle, Reliable fault diagnosis scheme for a spacecraft control system. *J. Risk Reliab.* **222**, 139–152 (2008). doi:10.1243/1748006XJRR98
- L. Perea, P. Elsegui, New state update equation for the unscented Kalman filter. *AIAA J. Guid. Control Dyn.* **31**(5), 1500–1504 (2008)
- I. Rapoport, Y. Oshman, Fault-tolerant particle filtering by using interacting multiple model-based Rao-Blackwellisation. *AIAA J. Guid. Control Dyn.* **28**(6), 1171–1177 (2005)
- H.P. Rotstein, R. Ingvalson, T. Keviczky, G.J. Balas, Fault-detection design for uninhabited aerial vehicles. *AIAA J. Guid. Control Dyn.* **29**(5), 1051–1060 (2006)
- D. Rupp, G. Ducard, H.P. Geering, E. Shafai, Extended multiple model adaptive estimation for the detection of sensor and actuator faults, in *Proceedings of IEEE Control and Decision Conference, and European Control Conference*, Seville, Spain, 2005, pp. 3079–3084
- P.A. Samara, G.N. Fouskitakis, J.S. Sakellariou, S.D. Fassois, A statistical method for the detection of sensor abrupt faults in aircraft control systems. *IEEE Trans. Control Syst. Technol.* **16**(4), 789–798 (2008)
- J.D. Schierman, D.G. Ward, J.R. Hull, N. Gandhi, M.W. Oppenheimer, D.B. Doman, Integrated adaptive guidance and control for re-entry vehicles with flight-test results. *AIAA J. Guid. Control Dyn.* **27**(6), 975–988 (2004)
- J.-Y. Shin, C. Belcastro, T. Khong, Closed-loop evaluation of an integrated failure identification and fault tolerant control system for a transport aircraft, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, 2006. AIAA 2006–6310
- D. Shore, M. Bodson, Flight testing of a reconfigurable control system on an unmanned aircraft. *AIAA J. Guid. Control Dyn.* **28**(4), 698–707 (2005)
- R.F. Stengel, *Flight Dynamics* (Princeton University Press, Princeton, 2004)
- B. Stevens, F. Lewis, *Aircraft Control and Simulation*, 2nd edn. (Wiley, New York, 2003)
- I. Szaszi, A. Marcos, G. Balas, J. Bokor, Linear parameter-varying detection filter design for a Boeing 747-100/200 aircraft. *AIAA J. Guid. Control Dyn.* **28**(3), 461–470 (2005)
- N. Tanaka, S. Suzuki, K. Masui, H. Tomita, Restructurable guidance and control for aircraft with failures considering gusts effects. *AIAA J. Guid. Control Dyn.* **29**(3), 635–642 (2006)
- G. Tao, S. Chen, X. Tang, S.M. Joshi, *Adaptive Control of Systems with Actuator Failures* (Springer, London/Berlin/Heidelberg, 2004)
- J. Urnes, R. Yeager, J. Stewart, Flight demonstration of the self-repairing flight control system in a NASA F-15 aircraft, in *National Aerospace Electronics Conference*, Rept. 90CH2881-1, Dayton, OH, 1990
- D. Ward, R. Barron, A self-designing receding horizon optimal flight controller, in *Proceedings of the IEEE American Control Conference*, Seattle, WA, 1995, pp. 3490–3494

- D. Ward, R.L. Barron, M.P. Carley, T.J. Curtis, Real-time parameter identification for self-designing flight control, in *Proceedings of the National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, 1994
- D.G. Ward, J.F. Monaco, M. Bodson, Development and flight testing of a parameter identification algorithm for reconfigurable control. *AIAA J. Guid. Control Dyn.* **21**(6), 948–956 (1998)
- K. Wise, J. Brinker, A. Calise, D. Enns, M. Elgersma, P. Voulgaris, Direct adaptive reconfigurable flight control for a tailless advanced fighter aircraft. *Int. J. Robust Nonlinear Control* **9**, 999–1012 (1999)
- A. Younghwan, *A Design of Fault Tolerant Flight Control Systems for Sensor and Actuator Failures Using On-Line Learning Neural Networks*. Ph.D. thesis, West Virginia University, 1998
- P. Zarchan, H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*. Progress in Astronautics and Aeronautics, vol. 208, 2nd edn. (AIAA Inc., Reston, 2005)
- Y. Zhang, J. Jiang, Integrated design of reconfigurable fault-tolerant control systems. *AIAA J. Guid. Control Dyn.* **24**(1), 133–136 (2000)

Experimental Validation of Fault Detection and Diagnosis for Unmanned Aerial Vehicles

45

Abbas Chamseddine, Mohammad Hadi Amoozgar, and
Youmin M. Zhang

Contents

| | | |
|--------|--|------|
| 45.1 | Introduction | 1124 |
| 45.2 | Description and Dynamics of the Quadrotor UAV System | 1131 |
| 45.2.1 | Dynamics | 1133 |
| 45.2.2 | ESCs, Motors, and Propellers | 1134 |
| 45.2.3 | Geometry | 1135 |
| 45.3 | Fault Detection and Diagnosis for the Quadrotor Helicopter | 1136 |
| 45.3.1 | Unscented Kalman Filter | 1136 |
| 45.3.2 | Two-Stage Kalman Filter | 1139 |
| 45.3.3 | Adaptive Two-Stage Kalman Filter | 1143 |
| 45.4 | Experimental Results | 1145 |
| 45.4.1 | Single Fault Scenario: Control Effectiveness Loss in the Third Motor | 1145 |
| 45.4.2 | Simultaneous Faults Scenario: Control Effectiveness Loss in All Motors | 1147 |
| 45.4.3 | Analysis of the Obtained Results | 1151 |
| 45.5 | Conclusion | 1152 |
| | References | 1153 |

Abstract

This chapter investigates the problems of fault detection, diagnosis, and fault-tolerant control for unmanned aerial vehicles (UAVs). It presents first a detailed overview on the existing experimental works considering these problems in the literature. The existing works consider fixed-wing as well as rotorcraft UAVs including the single-rotor and the multi-rotor helicopters (also known as quadrotors). Later on, the chapter discusses three Kalman filters employed for actuator fault detection and diagnosis, namely, the unscented Kalman filter, the two-stage Kalman filter, and the adaptive two-stage Kalman filter. The three

A. Chamseddine • M.H. Amoozgar • Y.M. Zhang (✉)

Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

e-mail: abbasc@encs.concordia.ca; m.amoozg@encs.concordia.ca; ymzhang@encs.concordia.ca

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

1123

DOI 10.1007/978-90-481-9707-1_41,

© Springer Science+Business Media Dordrecht 2015

filters are experimentally applied to a quadrotor helicopter UAV test bed at the Department of Mechanical and Industrial Engineering of Concordia University. The obtained results are shown, compared, and discussed.

45.1 Introduction

Unmanned aerial vehicles are being incorporated in an increasing variety of operations such as surveillance, remote sensing, search, geographic studies, as well as various military and security applications. This expansion will bring UAVs into difficult new situations such as flight in urban environments where reliability is particularly critical. The poor reliability of current unmanned vehicles presents a roadblock to their success in demanding new flight environments (Drozeski et al. 2005). The Office of the Secretary of Defense acknowledges this shortcoming in the UAV Roadmap 2002–2027 (Office of the Secretary of Defense 2002). It identifies the development of self-repairing, smart flight-control systems as a crucial step in the overall advancement of UAV autonomy: “Improving UAV reliability is the single most immediate and long-reaching need to ensure their success.” Damage tolerance capability is also crucial for UAVs such as unmanned combat aircraft systems operating in extremely hazardous environments. The technology for this damage tolerance would need to provide virtually instantaneous, autonomous assessment of damage incurred, followed by an immediate response that alters the flight-control system to compensate for the effects of that damage. A survey on fault diagnosis and fault-tolerant control for UAVs can be found in Qi and Han (2007) and Sadeghzadeh and Zhang (2011). This chapter will only focus on research works with experimental application of fault detection, diagnosis, and fault-tolerant control in the presence of actuator, control surface faults, and structural damages.

In the context of the Joint Unmanned Combat Aircraft Systems (J-UCAS) program sponsored by the Defense Advanced Research Projects Agency (DARPA), Rockwell Collins developed a new approach to autonomously mitigate the effects of physical damage to aircraft in an air combat environment. The efforts focused on airborne demonstrations of autonomous damage detection, recovery, and the subsequent real-time reconfiguration of the control laws needed to maintain vehicle stability. These adjustments enable air vehicles to complete their missions after incurring battle damage and allow them to safely return home. The developed technology is based on a unique, software-based approach to vehicle control called automatic supervisory adaptive control (ASAC) system. Many flight demonstration videos have been released for damage-tolerant flight control and autonomous landing capabilities on an unmanned subscale F/A-18 and the RQ-7B Shadow unmanned aircraft system (Fig. 45.1). Battle damage and in-flight failure is simulated in Rockwell Collins (2008) by ejecting more than 60 % of the subscale F/A-18’s right wing. Upon ejection, the ASAC system reacts to the airplane’s new vehicle configuration, automatically regains baseline performance, and continues to fly. The airplane successfully lands automatically despite the wing damage. An in-flight comparison is provided in Rockwell Collins (2009) where the ASAC system

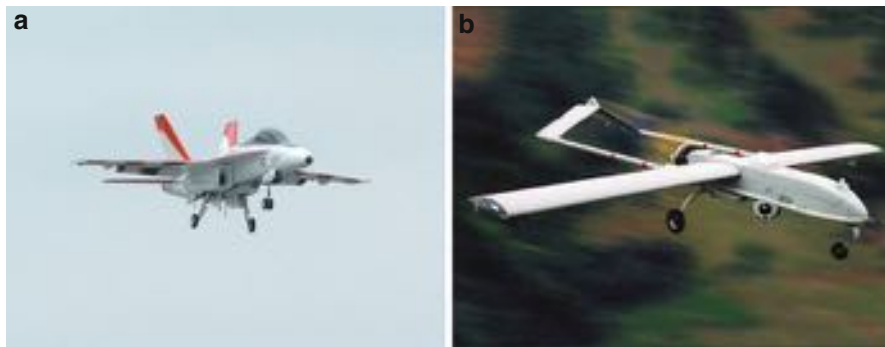


Fig. 45.1 The UAVs used by Rockwell Collins for the ASAC flight demonstration. (a) The unmanned subscale F/A-18. (b) The RQ-7B Shadow UAV

is activated and deactivated in the presence of a wing damage. A more severe fault is considered in Rockwell Collins (2010b) where the damage tolerance flight test involved ejecting 80 % of the F/A-18's right wing. In that flight, the ASAC technology quickly brings the aircraft back to trimmed flight after damage, and the vehicle continues to fly its mission before completing its flight with an autonomous landing. The technology also shows to be efficient in the presence of multiple surface damages: in Rockwell Collins (2010a), ASAC is shown to recover the aircraft from damage in the right wing, the right horizontal stabilizer, and the right vertical stabilizer. After the F/A-18 completes its takeoff, 60 % of its right wing and 30 % of its right vertical stabilizer are ejected simultaneously. The aircraft violently rolls over 100° before ASAC brings the wings back to level position and the vehicle continues to fly its mission. Later in the flight, 30 % of the right horizontal stabilizer is ejected. The UAV performs then an autonomous landing with its damaged airframe. Later on, the DARPA, the U.S. Army, and Rockwell Collins demonstrated damage-tolerance control on the RQ-7B Shadow unmanned aircraft system (Fig. 45.1), marking the first time that the technology has been tested on a fielded platform. The tests included ejecting 20 in. of the Shadow's wing during flight (Rockwell Collins 2010c). Despite the damage sustained by the Shadow, it remains steady in-flight and lands successfully.

Similar in-flight wing damage scenarios have been carried out at the Mechanical Engineering Department, University of Utah, in 2001–2002 using the Twinstar radio-controlled model aircraft (Shore and Bodson 2004). In this latter work, four in-flight failures were planned: frozen left elevator, frozen left aileron, left or right engine failure, and separation of left stabilizer/elevator. A recursive parameter identification is then employed to track the effects of the failures, such as reduced effectiveness of pitch commands due to a locked elevator or roll and sideslip due to engine failure. For the fault-tolerant flight control, a continuously adaptive approach is used where the gains of the control law are computed from the estimated parameters, and no attempt is made to categorize the health or failure state of the aircraft. The reconfigurable control algorithm can be viewed as a special case of



Fig. 45.2 The Twinstar by Multiplex

model reference adaptive control. The Twinstar UAV shown in Fig. 45.2 has been also considered at the School of Aerospace Engineering of Georgia Institute of Technology. The Twinstar, produced by Multiplex, is a commercially available multi-engine model-scale aircraft. It is made from highly resilient Styrofoam material and is an ideal candidate for fault-tolerant control work (Chowdhary et al. 2011). This vehicle has the capability to easily produce in-flight fault scenarios such as control surface failures, structural damage, and wing breakage. The vehicle has been equipped with the AFI FCS-20 onboard flight computer system (Adaptive Flight online). The AFI FCS-20 is a complete autopilot with a reliable navigation solution, fusing outputs from multiple embedded sensors to provide high-fidelity measurements of aircraft velocity. Two videos in Georgia Tech (2009a,b) show the Twinstar with 50 % in-flight right-wing ejection along with complete loss of right aileron functionality. Although that there is no information if autonomous landing is performed, one can see that the aircraft is still capable to autonomously track a predefined trajectory in spite of the severe structural damage. It should be noted that a baseline nonadaptive controller is employed in fault-free and fault conditions, and thus oscillations in angle of attack and roll are observed, which result in oscillations in altitude. In future works, adaptive control algorithms will be augmented to the baseline control algorithm to mitigate the observed oscillations and enable safe landing in the presence of significant structural faults.

As for fault-tolerant control approaches, several neural networks (NN) techniques have been applied to many aircrafts in the presence of actuator faults. The NASA/Boeing X-36 (Fig. 45.3) Tailless Fighter Agility Research Aircraft program successfully demonstrated the tailless fighter design using advanced technologies to improve the maneuverability and survivability of possible future fighter aircraft (RESTORE online). In a follow-on effort, the Air Force Research Laboratory (AFRL), Wright-Patterson Air Force Base, Ohio, contracted with Boeing to fly AFRL's Reconfigurable Control for Tailless Fighter Aircraft (RESTORE) software as a demonstration of the adaptability of NN techniques to compensate for in-flight damage/malfunction of flaps, ailerons, and rudders. Two RESTORE research



Fig. 45.3 The X-36 tailless fighter

flights were flown in December 1998, proving the viability of the software approach. The NN was developed to adaptively regulate the inversion error between the pre-estimated aircraft model and the true aircraft dynamics. The inversion error can be caused by the model uncertainty, actuator failure, or aircraft damage. A control allocation module was used to distribute the desired control response from the control algorithms to the remaining “healthy” control actuators. A system identification module uses a least-squares algorithm to estimate aerodynamic parameters. Null-space injection is used to briefly excite control surfaces to obtain these estimates without significant performance degradation. Another project using the NN is the Intelligent Flight Control System (IFCS) research project (IFCS online) at NASA Dryden Flight Research Center. The IFCS project was established to design aircraft flight controls that can optimize aircraft performance in both normal and failure conditions. IFCS was designed to incorporate self-learning NN concepts with different purposes and levels of criticality into the flight-control software. This enables a pilot to maintain control and safely land an aircraft that has suffered a major systems failure and/or combat damage. The test-bed aircraft for the IFCS project is the NASA NF-15B. This aircraft has been highly modified from a standard F-15 (Fig. 45.4) configuration to include canard control surfaces, thrust vectoring nozzles, and a digital fly-by-wire flight-control system. Later on, West Virginia University (WVU) has been involved in the IFCS controller design and flight testing as well as the design of a “safety monitor” scheme to allow smooth and safe transitions from conventional to research control laws and from research control laws at nominal conditions to failure conditions (Gu 2004; Napolitano 2002). Furthermore, one of the WVU YF-22 Research Aircraft Models (Fig. 45.5) is used to test a set of IFCS control laws as a scaled-down version of the IFCS flight-testing program. A recent video (FTC 2011) shows a fault-tolerant flight control and safe landing for a model-scale aircraft in the presence of 20 % wing ejection using neural networks.

The quadrotor helicopter system has been also considered in the context of fault-tolerant control. The quadrotor UAV is a relatively simple, affordable, and easy



Fig. 45.4 The F-15 aircraft



Fig. 45.5 The YF-22 Research Aircraft Models of Flight Control Systems Laboratory of WVU

to fly system, and thus it has been widely used to develop, implement, and test-fly methods in control, fault diagnosis, fault-tolerant control as well as multiagent based technologies in formation flight, cooperative control, distributed control, mobile wireless networks, and communications. Few research laboratories are carrying out advanced theoretical and experimental works on the system. Among others, one may cite, for example, the UAV SWARM health management project of the Aerospace Controls Laboratory at MIT (SWARM online) supported by Boeing Phantom Works, the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control project (STARMAC online), and the Micro Autonomous Systems Technologies project (MAST online). The Active Adaptive Control Laboratory (AACLAB online) at MIT considered the fault-tolerant control problem for the quadrotor UAV using the Draganflyer V Ti Pro (see Fig. 45.6): a video in Dydek (2010b) shows a three-way comparison between nominal (nonadaptive), model reference adaptive control (MRAC), and combined/composite model reference adaptive control (CMRAC) in the presence of 30 % thrust loss. In Dydek (2010a), a video shows a comparison of nominal (nonadaptive) and adaptive controllers in response to a sudden loss of thrust due to one of the propellers being cut mid-flight. A team of researchers at



Fig. 45.6 The Draganflyer V Ti Pro

the Department of Mechanical and Industrial Engineering of Concordia University has been working on a research and development project on fault-diagnosis, fault-tolerant, and cooperative control of multiple UAVs since 2007 (see the Networked Autonomous Vehicles laboratory (NAV online)). This work is financially supported by NSERC (Natural Sciences and Engineering Research Council of Canada) through a Strategic Project Grant and a Discovery Project Grant and three Canadian-based industrial partners (Quanser Inc., Opal-RT Technologies Inc., and Numerica Technologies Inc.) as well as Defence Research and Development Canada (DRDC) and Laval University. In addition to the work that has been carried out for the multi-vehicles case, many fault-tolerant control (FTC) strategies have been developed and applied to the single-vehicle quadrotor UAV system. The objective is to consider actuator faults and to propose FTC methods to accommodate as much as possible the fault effects on the system performance. Detailed information about the group activities can be found in Zhang and Chamseddine (2011). The proposed methods have been tested either in simulation, experimental, or both frameworks where the experimental implementation has been carried out using a cutting-edge quadrotor UAV test bed known as Qball-X4. The developed approaches include the gain-scheduled PID (GS-PID), model reference adaptive control (MRAC), sliding mode control (SMC), backstepping control (BSC), model predictive control (MPC), and flatness-based trajectory planning/replanning (FTPR). As an example of the work carried out at Concordia University, a comparison is shown in Chamseddine (2011a,b) between the linear quadratic regulator (LQR) and the conventional MRAC in the presence of a mid-flight propeller damage of 12 and 16 %, respectively. In Chamseddine and Sadeghzadeh (2011), the quadrotor X4 (4 motors) unmanned aerial helicopter is modified to an X6 (6 motors) helicopter, and the control reallocation technique is used to maintain the helicopter stability in the presence of 70 % and up to 100 % propeller damage.

Some of the above FTC methods require information about the time of occurrence, the location, and the amplitude of faults whereas others do not. In the former



Fig. 45.7 The MARVIN helicopter

case, a fault detection and diagnosis (FDD) module is needed to detect, isolate, and identify the occurred faults. In Heredia et al. (2008), the authors investigate the FDD problem for unmanned rotorcraft systems in the presence of actuator faults. The system that has been considered is the MARVIN helicopter (Fig. 45.7) where stuck-type faults has been produced in the main and tail rotors in different flight conditions (near hover, forward flight at different velocities, etc.). An input-output model of the helicopter system is identified for output prediction then linear observers are used for fault detection. A fault-tolerant control architecture that couples techniques for fault detection and identification with reconfigurable flight control is developed at Georgia Institute of Technology (Drozeski et al. 2005; Drozeski 2005). Neural networks are employed for fault detection and identification, and an adaptive neural network feedback linearization technique is employed to stabilize the vehicle after fault detection. Actual flight test results are carried out on the Yamaha GTMax helicopter shown in Fig. 45.8 to support the validity of the approach. The fault-tolerant control architecture is tested in the presence of several actuator faults such as faults in the swashplate and the collective rotor. The Shenyang Institute of Automation, Chinese Academy of Sciences, has been focusing its research on rotorcraft UAV for several years now (Qi et al. 2009). Several experimental platforms are designed for advanced control algorithm research demonstration including navigation, advanced flight control, 3D path planning, and fault-tolerant control. Among these platforms, the ServoHeli-20 is a model class platform which has 20 kg takeoff weight. The ServoHeli-40 and ServoHeli-120 (Fig. 45.9) are engineering class platforms for highway patrol, electrical line patrol, and photography. They have 40 and 120 kg takeoff weight and have performed fully autonomous flight-control experimental demonstration. In the recent research of Qi et al. (2009), a novel adaptive unscented Kalman filter (AUKF) is proposed for onboard failure coefficient estimation and a new fault-tolerant control method is designed against the actuator failure of the ServoHeli series rotorcraft UAVs. The filter method with adaptability to statistical characteristic of noise is presented to improve the estimation accuracy of traditional UKF. The algorithm with the adaptability to statistical characteristic of noise, named Kalman filter (KF)-based adaptive UKF,



Fig. 45.8 The Yamaha GTMax helicopter

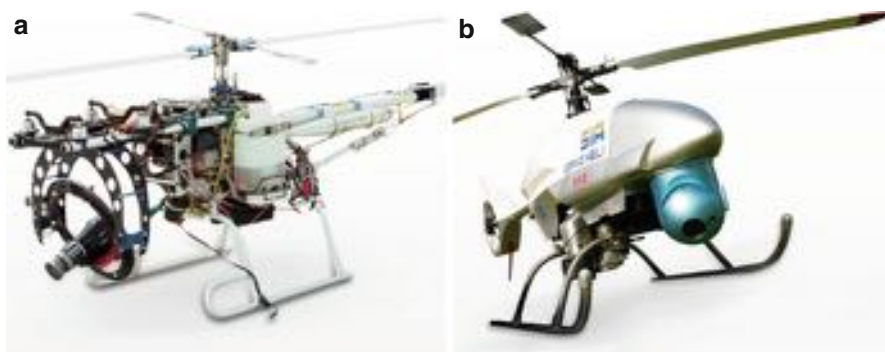


Fig. 45.9 The ServoHeli-40 and ServoHeli-120 rotorcraft UAV systems. (a) The ServoHeli-40. (b) The ServoHeli-120

is proposed to improve the UKF performance. By introducing the actuator health coefficients (AHCs) into the dynamics equation of a rotorcraft UAV, the proposed AUKF is utilized to online estimate both the flight states and the AHCs. A fault adaptive control is further designed based on the estimated states and AHCs.

The subsequent sections of this chapter present some experimental results on the actuator FDD problem applied to a quadrotor helicopter test bed. Three Kalman filters are considered: the unscented Kalman filter, the two-stage Kalman filter, and the adaptive two-stage Kalman filter. The quadrotor helicopter test bed is first presented in the next section before proceeding with the FDD problem.

45.2 Description and Dynamics of the Quadrotor UAV System

The Qball-X4 test bed developed by Quanser Inc. through the above-mentioned NSERC-SPG project is shown in Fig. 45.10a. It is enclosed within a protective carbon fiber round cage (therefore a name of Qball-X4) to ensure safe operation of the vehicle and protection to the personnel who is working with the vehicle in an indoor

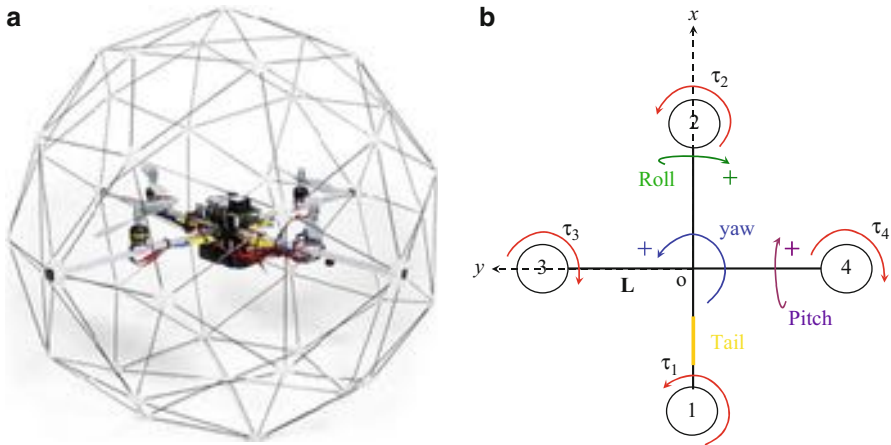


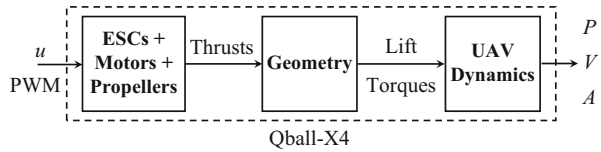
Fig. 45.10 The Quanser Qball-X4 quadrotor UAV and its schematic representation. **(a)** The Quanser Qball-X4 quadrotor UAV, **(b)** Qball-X4 schematic representation

research and development environment. It uses four 10-in. propellers and standard motors and speed controllers. It is equipped with the Quanser embedded control module (QECM), which is comprised of a Quanser HiQ aero data acquisition card and a QuaRC-powered Gumstix single-board embedded computer where QuaRC is Quanser's real-time control software. The Quanser HiQ provides high-resolution accelerometer, gyroscope, and magnetometer IMU sensors as well as servo outputs to drive four motors. The onboard Gumstix computer runs QuaRC, which allows to rapidly develop and deploy controllers designed in MATLAB/Simulink environment to real-time control of the Qball-X4. The controllers run onboard the vehicle itself, and runtime sensors measurement, data logging, and parameter tuning are supported between the ground host computer and the target vehicle (Quanser 2010).

The block diagram of the entire UAV system is illustrated in Fig. 45.11. It is composed of three main parts:

- The first part represents the electronic speed controllers (ESCs), the motors, and the propellers in a set of four. The input to this part is $u = [u_1 \ u_2 \ u_3 \ u_4]^T$ which are pulse-width modulation (PWM) signals. The output is the thrust vector $T = [T_1 \ T_2 \ T_3 \ T_4]^T$ generated by four individually controlled motor-driven propellers.
- The second part is the geometry that relates the generated thrusts to the applied lift and torques to the system. This geometry corresponds to the position and orientation of the propellers with respect to the system's center of mass.
- The third part is the dynamics that relate the applied lift and torques to the position P , velocity V , and acceleration A of the Qball-X4.

The subsequent sections describe the corresponding mathematical model for each of the blocks in Fig. 45.11.

Fig. 45.11 The UAV system block diagram

45.2.1 Dynamics

The Qball-X4 dynamics in a hybrid coordinate system are given hereafter where the position dynamics are expressed in the inertial frame and the angular dynamics are expressed in the body frame (Bresciani 2008):

$$\begin{aligned}
 m\ddot{x} &= u_z (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - k_x \dot{x} \\
 m\ddot{y} &= u_z (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) - k_y \dot{y} \\
 m\ddot{z} &= u_z (\cos \phi \cos \theta) - mg - k_z \dot{z} \\
 J_x \dot{p} &= u_p + (J_y - J_z) qr - J_T q \Omega - k_p p \\
 J_y \dot{q} &= u_q + (J_z - J_x) pr - J_T p \Omega - k_q q \\
 J_z \dot{r} &= u_r + (J_x - J_y) pq - k_r r
 \end{aligned} \tag{45.1}$$

where x , y , and z are the coordinates of the quadrotor UAV center of mass in the inertial frame. m is the system mass and J_x , J_y , and J_z are the moments of inertia along y , x , and z directions, respectively. θ , ϕ , and ψ are the pitch, roll, and yaw Euler angles and p , q , and r are the angular velocities in the body-fixed frame. k_x , k_y , k_z , k_p , k_q , and k_r are drag coefficients and are constant. J_T is the moment of inertia for each motor, and Ω is the overall speed of propellers:

$$\Omega = -\Omega_1 - \Omega_2 + \Omega_3 + \Omega_4 \tag{45.2}$$

where Ω_i is the i th propeller speed.

The angular velocities in the inertial frame (Euler rates) can be related to those in the body frame as follows:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{45.3}$$

Close to hovering conditions, the matrix in the above equation is close to identity matrix, and therefore the angular velocities in the body frame can be seen as the angular velocities in the inertial frame. The model (45.1) can then be written as

$$\begin{aligned}
m\ddot{x} &= u_z (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - k_x \dot{x} \\
m\ddot{y} &= u_z (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) - k_y \dot{y} \\
m\ddot{z} &= u_z (\cos \phi \cos \theta) - mg - k_z \dot{z} \\
J_x \ddot{\theta} &= u_\theta + (J_y - J_z) \dot{\phi} \dot{\psi} - J_T \dot{\phi} \Omega - k_\theta \dot{\theta} \\
J_y \ddot{\phi} &= u_\phi + (J_z - J_x) \dot{\theta} \dot{\psi} - J_T \dot{\theta} \Omega - k_\phi \dot{\phi} \\
J_z \ddot{\psi} &= u_\psi + (J_x - J_y) \dot{\theta} \dot{\phi} - k_\psi \dot{\psi}
\end{aligned} \tag{45.4}$$

where u_p , u_q , u_r , k_p , k_q , and k_r have been respectively changed to u_θ , u_ϕ , u_ψ , k_θ , k_ϕ , and k_ψ for notation convenience. At low speeds, one can obtain a simplified nonlinear model of (45.4) by neglecting drag terms and gyroscopic and Coriolis-centripetal effects:

$$\begin{aligned}
m\ddot{x} &= u_z (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\
m\ddot{y} &= u_z (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\
m\ddot{z} &= u_z (\cos \phi \cos \theta) - mg \\
J_x \ddot{\theta} &= u_\theta \\
J_y \ddot{\phi} &= u_\phi \\
J_z \ddot{\psi} &= u_\psi
\end{aligned} \tag{45.5}$$

A further simplified linear model can be obtained by assuming hovering conditions ($u_z \approx mg$ in the x and y directions) with no yawing ($\psi = 0$) and small roll and pitch angles:

$$\begin{aligned}
\ddot{x} &= \theta g; & J_x \ddot{\theta} &= u_\theta \\
\ddot{y} &= -\phi g; & J_y \ddot{\phi} &= u_\phi \\
\ddot{z} &= u_z/m - g; & J_z \ddot{\psi} &= u_\psi
\end{aligned} \tag{45.6}$$

The nonlinear model (45.5) will be used later for the unscented Kalman filter design whereas the simplified model (45.6) will be used for the two-stage and adaptive two-stage Kalman filters.

45.2.2 ESCs, Motors, and Propellers

The motors of the Qball-X4 are outrunner brushless motors. The generated thrust T_i of the i th motor is related to the i th PWM input u_i by a first-order linear transfer function:

$$T_i = K \frac{\omega}{s + \omega} u_i ; i = 1, \dots, 4 \quad (45.7)$$

where K is a positive gain and ω is the motor bandwidth. K and ω are theoretically the same for the four motors, but this may not be the case in practice, and therefore this can be one of sources of modeling errors/uncertainties for the FDD schemes.

45.2.3 Geometry

A schematic representation of the Qball-X4 is given in Fig. 45.10b. The motors and propellers are configured in such a way that the back and front (1 and 2) motors spin clockwise and the left and right (3 and 4) motors spin counterclockwise. Each motor is located at a distance L from the center of mass o , and when spinning, a motor produces a torque τ_i . The origin of the body-fixed frame is the system's center of mass o with the x -axis pointing from back to front and the y -axis pointing from right to left. The thrust T_i generated by the i th propeller is always pointing upward in the z -direction in parallel to the motor's rotation axis. The thrusts T_i and the torques τ_i result in a lift in the z -direction (body-fixed frame) and torques about the x , y , and z axes. The relations between the lift/torques and the thrusts are

$$\begin{aligned} u_z &= T_1 + T_2 + T_3 + T_4 \\ u_\theta &= L(T_1 - T_2) \\ u_\phi &= L(T_3 - T_4) \\ u_\psi &= \tau_1 + \tau_2 - \tau_3 - \tau_4 \end{aligned} \quad (45.8)$$

The torque τ_i produced by the i th motor is directly related to the thrust T_i via the relation of $\tau_i = K_\psi T_i$ with K_ψ as a constant. In addition, by setting $T_i \approx K u_i$ from (45.7), the relation (45.8) reads

$$\begin{aligned} u_z &= K(u_1 + u_2 + u_3 + u_4) \\ u_\theta &= KL(u_1 - u_2) \\ u_\phi &= KL(u_3 - u_4) \\ u_\psi &= KK_\psi(u_1 + u_2 - u_3 - u_4) \end{aligned} \quad (45.9)$$

where u_z is the total lift generated by the four propellers and applied to the quadrotor UAV in the z -direction (body-fixed frame). u_θ , u_ϕ , and u_ψ are, respectively, the applied torques in θ , ϕ , and ψ directions (see Fig. 45.10b).

45.3 Fault Detection and Diagnosis for the Quadrotor Helicopter

Fault detection and diagnosis (FDD) aims to detect abnormal behaviors of a process due to a component failure and eventually isolate the exact location of the failed component and identify the failure type and its severity. This problem has been extensively considered in the literature (see, e.g., Ding 2008; Isermann 2006; Zhang and Jiang 2008 and the references therein) since a better knowledge of the failure location, type, and amplitude greatly helps in minimizing the fault effects on the process behavior. This is particularly important for safe operation and/or fault-tolerant control of safety-critical systems, such as aircrafts, spacecrafts, nuclear power plants, and chemical plants processing hazardous materials where the consequences of a minor fault in a system component can be catastrophic.

This chapter considers the FDD problem for the quadrotor helicopter in the presence of actuator faults. Such faults can be modeled as follows:

$$u_i^f = (1 - w_i) u_i \leq (1 - w_i) u_{\max} ; \quad (i = 1, \dots, 4) \quad (45.10)$$

where w_i represents the loss of effectiveness in the i th rotor, $w_i = 0$ denotes a healthy rotor, $w_i = 1$ denotes a complete loss of the i th rotor, and $0 < w_i < 1$ represents a partial loss of control effectiveness. During the mission and after the occurrence of a fault in one of the UAV rotors, two cases can be distinguished: (a) the damaged UAV cannot recover and the system is lost and (b) the damaged UAV recovers and maintains its stability. It can be shown from (45.6) and (45.9) that the required PWM inputs to keep a quadrotor UAV in hovering flight condition (i.e., for $x = 0$, $y = 0$, and $\psi = 0$) are $u_i^* = mg/4K$ for $i = 1, \dots, 4$. Thus, a damaged quadrotor can always maintain its altitude if

$$u_i^* = \frac{mg}{4K} \leq (1 - w_i) u_{\max} ; \quad i = 1, \dots, 4 \quad (45.11)$$

or

$$w_i \leq 1 - \frac{mg}{4K u_{\max}} ; \quad i = 1, \dots, 4 \quad (45.12)$$

45.3.1 Unscented Kalman Filter

This work employs a parameter estimation-based FDD where the unscented Kalman filter (UKF) is used to estimate the parameters w_i of (45.10). In the fault-free case, the estimates of w_i ($i = 1, \dots, 4$) are close to 0, and a deviation from 0 will be an indication for a fault occurrence. The UKF is a member of the linear regression Kalman filters (LRKF) and it was developed to overcome some of the shortcomings of the extended Kalman filter (EKF) in the estimation of nonlinear systems (VanDyke et al. 2004). The UKF is an extension of the traditional Kalman filter for the estimation of nonlinear systems by using the unscented

transformation (UT) where the UT is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation (Julier and Uhlmann 1997). It uses a set of sample or *sigma points* that are determined from the a priori mean and covariance of the variable. The sigma points undergo the nonlinear transformation, and the mean and covariance of the variable are determined from the transformed sigma points. The main motivation for using UKF for FDD resides in the advantages and the superiority of this approach over the EKF and more specifically the better convergence characteristics and greater accuracy in the case of nonlinear systems (VanDyke et al. 2004).

In this chapter, based on the fact that all state variables are available from the sensors in the Qball-X4 system, UKF has been used for estimating only necessary parameters w_i ($i = 1, \dots, 4$). This strategy is different from the traditional scheme for using UKF for state or simultaneous state and parameter estimation (Vitale et al. 2009; Ma and Zhang 2010), and it has the advantage of simplifying the FDD scheme to be more suitable for real-time applications. The parameter estimation problem, also referred to as system identification, involves determining a nonlinear mapping $y_k = G(x_k, w)$ where x_k is the input, y_k is the output, and the nonlinear map G is parameterized by the parameter vector w . Typically, a training set is provided with sample pairs consisting of known inputs and desired outputs, $\{x_k, d_k\}$, where $k = 1, \dots, N$ and N is the total number of sample data. The error is defined as $e_k = d_k - G(x_k, w)$, and the goal of learning involves solving for the parameters w in order to minimize some given function of the error (der Merwe and Wan 2001). Parameters can be efficiently estimated online by writing a new state-space representation as follows:

$$w_k = w_{k-1} + r_k \quad (45.13)$$

$$d_k = G(x_k, w_k) + e_k \quad (45.14)$$

where the parameters w_k correspond to a stationary process with identity state transition matrix, driven by process noise r_k . The output d_k corresponds to a nonlinear observation on w_k . The UKF implementation for parameter estimation is then given hereafter.

Initialize the parameters vector and covariance with

$$\hat{w}_0 = E\{w\} \quad (45.15)$$

$$P_{w_0} = E\{(w - \hat{w}_0)(w - \hat{w}_0)^T\} \quad (45.16)$$

For $k \in \{1, \dots, \infty\}$, calculate the time update of the parameters vector and covariance:

$$\hat{w}_k^- = \hat{w}_{k-1} \quad (45.17)$$

$$P_{w_k}^- = P_{w_{k-1}} + Q_w \quad (45.18)$$

Calculate the sigma points from the a priori parameters mean and covariance:

$$\sigma_k = \begin{bmatrix} \hat{w}_k^- & \hat{w}_k^- + \nu \sqrt{P_{w_k}^-} & \hat{w}_k^- - \nu \sqrt{P_{w_k}^-} \end{bmatrix} \quad (45.19)$$

Calculate the expected measurement matrix using the nonlinear measurement model:

$$D_k = G(x_k, \sigma_k) \quad (45.20)$$

Calculate the mean measurement \hat{d}_k :

$$\hat{d}_k = \sum_{i=0}^{2L} w_i^m D_{i,k} \quad (45.21)$$

Calculate the measurement and the cross-correlation covariances:

$$P_{d_k} = \sum_{i=0}^{2L} w_i^c (D_{i,k} - \hat{d}_k) (D_{i,k} - \hat{d}_k)^T + R_w \quad (45.22)$$

$$P_{w_k d_k} = \sum_{i=0}^{2L} w_i^c (\sigma_{i,k} - \hat{w}_k^-) (D_{i,k} - \hat{d}_k)^T \quad (45.23)$$

Calculate the Kalman gain matrix from the cross-correlation and measurement covariances:

$$K_k = P_{w_k d_k} P_{d_k}^{-1} \quad (45.24)$$

Update measurements using equations:

$$\hat{w}_k = \hat{w}_k^- + K_k (d_k - \hat{d}_k) \quad (45.25)$$

$$P_{w_k} = P_{w_k}^- - K_k P_{d_k} K_k^T \quad (45.26)$$

Q_w and R_w are the process and measurement noise covariance matrices. $D_{i,k}$ is the i th column of the matrix D_k , and w_i^m and w_i^c are scalar weights defined as follows:

$$w_0^m = \frac{\lambda}{l + \lambda} \quad (45.27)$$

$$w_0^c = \frac{\lambda}{l + \lambda} + (1 - \alpha^2 + \beta) \quad (45.28)$$

$$w_i^m = w_i^c = \frac{1}{2(l + \lambda)} \quad ; \quad i = 1, \dots, 2l \quad (45.29)$$

where l is the dimension of the vector w , and $\lambda = \alpha^2(l + \kappa) - l$ and $v = \sqrt{(l + \lambda)}$ are two scaling parameters. α determines the spread of the sigma points around \hat{w} and is usually set to $1 \times 10^{-4} \leq \alpha \leq 1$. κ is a secondary scaling parameter set to $3 - l$ for parameter estimation. β is used to incorporate prior knowledge of the distribution of w (for Gaussian distributions, $\beta = 2$ is optimal).

The application of the above UKF algorithm to the Qball-X4 consists in plugging (45.9) into system dynamics given by (45.5) while taking into consideration the model of actuator faults in (45.10). By defining $(1 - w_i)$ as \bar{w}_i and discretizing the resulting equations using Euler's method, one obtains

$$\begin{aligned}
 x_{k+2} &= 2x_{k+1} - x_k + h^2 \frac{K}{m} (\bar{w}_1 u_{1k} + \bar{w}_2 u_{2k} + \bar{w}_3 u_{3k} + \bar{w}_4 u_{4k}) \quad (45.30) \\
 &\quad (\cos \phi_k \sin \theta_k \cos \psi_k + \sin \phi_k \sin \psi_k) \\
 y_{k+2} &= 2y_{k+1} - y_k + h^2 \frac{K}{m} (\bar{w}_1 u_{1k} + \bar{w}_2 u_{2k} + \bar{w}_3 u_{3k} + \bar{w}_4 u_{4k}) \\
 &\quad (\cos \phi_k \sin \theta_k \sin \psi_k - \sin \phi_k \cos \psi_k) \\
 z_{k+2} &= 2z_{k+1} - z_k + h^2 \frac{K}{m} (\bar{w}_1 u_{1k} + \bar{w}_2 u_{2k} + \bar{w}_3 u_{3k} + \bar{w}_4 u_{4k}) \\
 &\quad (\cos \phi_k \cos \theta_k) - h^2 g \\
 \theta_{k+2} &= 2\theta_{k+1} - \theta_k + h^2 \frac{KL}{J_1} (\bar{w}_1 u_{1k} - \bar{w}_2 u_{2k}) \\
 \phi_{k+2} &= 2\phi_{k+1} - \phi_k + h^2 \frac{KL}{J_2} (\bar{w}_3 u_{3k} - \bar{w}_4 u_{4k}) \\
 \psi_{k+2} &= 2\psi_{k+1} - \psi_k + h^2 \frac{KK_\psi}{J_3} (\bar{w}_1 u_{1k} + \bar{w}_2 u_{2k} - \bar{w}_3 u_{3k} - \bar{w}_4 u_{4k})
 \end{aligned}$$

where h is the sampling period. Finally, to perform fault detection and isolation, many rules can be used to evaluate the estimated parameters w_i such as the two-stage statistical hypothesis tests developed in Zhang and Jiang (2002).

45.3.2 Two-Stage Kalman Filter

The advantage of using two-stage Kalman filter (TSKF) is to simultaneously estimate both states and fault parameters, for the purpose of fault detection, isolation, and identification as well as providing full-state estimation for state feedback-based controllers when state vector is not available through measurements. To explain the basic idea of the TSKF, consider the following discrete linear state-space model:

$$\begin{aligned}
 x_{k+1} &= A_k x_k + B_k u_k + v_k^x \\
 y_{k+1} &= C_k x_{k+1} + v_{k+1}^y
 \end{aligned} \quad (45.31)$$

where $x_k \in R^n$, $u_k \in R^l$ and $y_{k+1} \in R^m$ are the state, control input, and output vectors, respectively. v_k^x and v_{k+1}^y are uncorrelated Gaussian random vectors with zero means and covariance matrices Q_k^x and R_k , respectively. In the application of Kalman filtering techniques, an accurate model of the process dynamics and measurements is required. However, in many practical cases, constant bias affects the system dynamics and observations and may lead to performance degradation of the filter if the bias is not incorporated in the model. By considering a bias vector $w_k \in R^p$ in the state-space equation, model (45.31) can be written as

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + F w_k + v_k^x \\ w_{k+1} &= w_k + v_k^w \\ y_{k+1} &= C_k x_{k+1} + v_{k+1}^y \end{aligned} \quad (45.32)$$

where v_k^w is an uncorrelated Gaussian random vector with zero mean and covariance matrix Q_k^w . One possible solution to estimate the bias vector w_k is to augment it into the state vector to make an augmented state vector which is estimated by using the augmented state Kalman filter (ASKF). The augmented state vector is of dimension $n + p$ which makes the ASKF computationally expensive. Another drawback of this method is that numerical problems may arise during implementation. To reduce the complexity of this problem, Keller and Darouach (1997) presented two parallel reduced-order filters which optimally implement the augmented state filter. The proposed algorithm is called two-stage Kalman filter (TSKF). In the context of fault detection and diagnosis, Wu et al. (1998, 2000) modeled actuator faults (loss of control effectiveness) as a bias vector in state equations then used TSKF to estimate the augmented bias vector. In Wu et al. (1998), the authors proposed the use of forgetting factor for TSKF.

In the current study, the effectiveness of actuators is estimated as the augmented random bias vector in the TSKF structure. By referring to (45.10), the loss of control effectiveness modeled as a bias vector is

$$w_k = [w_{1k} \ w_{2k} \ \dots \ w_{lk}]^T ; \ 0 \leq w_{ik} \leq 1 ; \ i = 1, 2, \dots, l \quad (45.33)$$

where $w_{ik} = 0$ and $w_{ik} = 1$ means that the i th actuator is completely healthy or fully damaged, respectively. By incorporating actuator faults in (45.32), the bias-augmented discrete linear time-varying state-space model is written as

$$x_{k+1} = A_k x_k + B_k u_k - B_k U_k w_k + v_k^x \quad (45.34)$$

$$w_{k+1} = w_k + v_k^w \quad (45.35)$$

$$y_{k+1} = C_k x_{k+1} + v_{k+1}^y \quad (45.36)$$

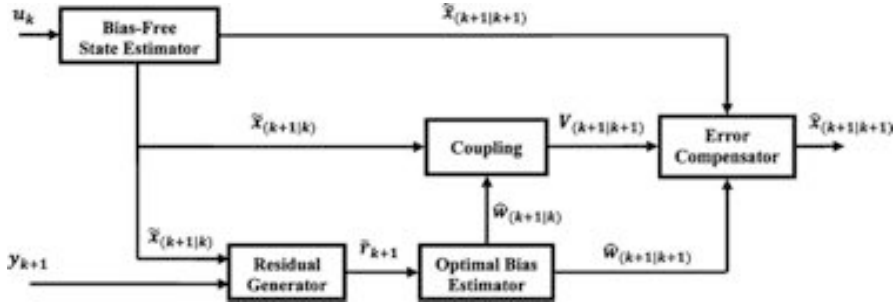


Fig. 45.12 The schematic diagram of the two-stage Kalman filter

where U_k is

$$U_k = \begin{bmatrix} u_{1k} & 0 & \dots & 0 \\ 0 & u_{2k} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & u_{lk} \end{bmatrix} \quad (45.37)$$

Since the additive noise v_k^w introduced into the bias evolution equation bears no relation to either the process noise v_k^x or the measurement noise v_{k+1}^y in the dynamic system model, the two-stage filtering algorithm by Keller and Darouach can be extended with some modification to obtain the bias estimates (Wu et al. 2000). It is worthy to note that the choice of Q_k^w plays an important role in the performance of the filter. In the most of real world situations, the loss of control effectiveness occurs abruptly, while in (45.34) it is modeled as a bias. To make the bias vector change fast enough (and also to track the true values), one may think of a large value for bias covariance. On the other hand, once the steady-state condition is reached, the bias covariance must be small to reduce the steady-state error in estimation. Then there is a trade-off between convergence speed and the accuracy of steady-state estimation of control effectiveness. The minimum variance solution to estimate the true values of biases is obtained by application of the TSKF.

Like other types of Kalman filters, the equations can be divided into two sets: the *time-update* equations and the *measurement-update* equations. The time-update equations, which can be distinguished by the $(k+1|k)$ subscription, are responsible to obtain a priori estimates by moving the state and error covariances one step ahead in the time domain. The measurement-update equations, shown by the $(k+1|k+1)$ subscription, are responsible to obtain a posteriori estimates through feedbacking measurements into the a priori estimates. In other words, time-update equations are used for prediction, while measurement-update equations are used for correction. Indeed the whole prediction-correction process is used to estimate the states as close as possible to their real values. Figure 45.12 shows a schematic flow diagram of the TSKF, and its implementation is given hereafter.

Estimate the bias-free state:

$$\tilde{x}_{k+1|k} = A_k \tilde{x}_{k|k} + B_k u_k + W_k \hat{w}_{k|k} - V_{k+1|k} \hat{w}_{k|k} \quad (45.38)$$

$$\tilde{P}_{k+1|k}^x = A_k \tilde{P}_{k|k}^x A_k^T + Q_k^x + W_k P_{k|k}^w W_k^T - V_{k+1|k} P_{k+1|k}^w V_{k+1|k}^T \quad (45.39)$$

$$\tilde{x}_{k+1|k+1} = \tilde{x}_{k+1|k} + \tilde{K}_{k+1}^x (y_{k+1} - C_{k+1} \tilde{x}_{k+1|k}) \quad (45.40)$$

$$\tilde{K}_{k+1}^x = \tilde{P}_{k+1|k}^x + C_{k+1}^T (C_{k+1} \tilde{P}_{k+1|k}^x C_{k+1}^T + R_{k+1})^{-1} \quad (45.41)$$

$$\tilde{P}_{k+1|k+1}^x = (I - \tilde{K}_{k+1}^x C_{k+1}) \tilde{P}_{k+1|k}^x \quad (45.42)$$

Determine the filter residual and its covariance:

$$\tilde{r}_{k+1} = y_{k+1} - C_{k+1} \tilde{x}_{k+1|k} \quad (45.43)$$

$$\tilde{S}_{k+1} = C_{k+1} \tilde{P}_{k+1|k}^x C_{k+1}^T + R_{k+1} \quad (45.44)$$

Estimate the optimal bias:

$$\hat{w}_{k+1|k} = \hat{w}_{k|k} \quad (45.45)$$

$$P_{k+1|k}^w = P_{k|k}^w + Q_k^w \quad (45.46)$$

$$\hat{w}_{k+1|k+1} = \hat{w}_{k+1|k} + K_{k+1}^w (\tilde{r}_{k+1} - H_{k+1|k} \hat{w}_{k|k}) \quad (45.47)$$

$$K_{k+1}^w = P_{k+1|k}^w H_{k+1|k}^T (H_{k+1|k} P_{k+1|k}^w H_{k+1|k}^T + \tilde{S}_{k+1})^{-1} \quad (45.48)$$

$$P_{k+1|k+1}^w = (I - K_{k+1}^w H_{k+1|k}) P_{k+1|k}^w \quad (45.49)$$

Calculate the coupling equations:

$$W_k = A_k V_{k|k} - B_k U_k \quad (45.50)$$

$$V_{k+1|k} = W_k P_{k|k}^w (P_{k+1|k}^w)^{-1} \quad (45.51)$$

$$H_{k+1|k} = C_{k+1} V_{k+1|k} \quad (45.52)$$

$$V_{k+1|k+1} = V_{k+1|k} - \tilde{K}_{k+1}^x H_{k+1|k} \quad (45.53)$$

Estimate the compensated error and its covariance:

$$\hat{x}_{k+1|k+1} = \tilde{x}_{k+1|k+1} + V_{k+1|k+1} \hat{w}_{k+1|k+1} \quad (45.54)$$

$$P_{k+1|k+1} = \tilde{P}_{k+1|k+1}^x + V_{k+1|k+1} P_{k+1|k+1}^w V_{k+1|k+1}^T \quad (45.55)$$

The application of the TSKF to the quadrotor UAV starts by deriving a linearized state-space model. Combining (45.6) and (45.9) yields to the linearized state-space model:

$$\begin{aligned}
\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \\ \psi \\ \dot{\psi} \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K/m & K/m & K/m & K/m \\ 0 & 0 & 0 & 0 \\ KL/J_1 & -KL/J_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & KL/J_2 & -KL/J_2 \\ 0 & 0 & 0 & 0 \\ KK_\psi/J_3 & KK_\psi/J_3 & -KK_\psi/J_3 & -KK_\psi/J_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} g \\
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ . \\ . \\ \psi \\ \dot{\psi} \end{bmatrix}
\end{aligned}$$

The covariance matrices for the implementation on Qball-X4 UAV are set to

$$\begin{aligned}
Q^x &= \begin{bmatrix} 10^{-6} \times I_{6 \times 6} & 0_{6 \times 6} \\ 0_{6 \times 6} & 10^{-12} \times I_{6 \times 6} \end{bmatrix}; \quad R = \begin{bmatrix} 10^{-6} \times I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 10^{-12} \times I_{3 \times 3} \end{bmatrix}; \\
Q^w &= 10^{-1} \times I_{4 \times 4}
\end{aligned}$$

45.3.3 Adaptive Two-Stage Kalman Filter

The adaptive two-stage Kalman filter (ATSKF) is addressed for the purpose of adaptive estimation of control effectiveness factors. To make the ordinary TSKF more responsive to abrupt changes in control effectiveness factor (which has been modeled as a bias vector in (45.34)), some modification has been suggested in Wu et al. (2000) and Zhang and Jiang (2002). A common way to make recursive algorithm (like Kalman filter's family) less sensitive to previous data history is to use forgetting factor. In other words, the main goal of using forgetting factor is to make the Kalman filter more sensitive to abrupt changes in the control effectiveness parameters. When a forgetting factor is added to the optimal bias estimator part in (45.46), the new time-updated equation of bias covariance is written as

$$P_{k+1|k}^w = P_{k|k}^w / \lambda + Q_k^w, \quad 0 < \lambda \leq 1 \quad (45.56)$$

where λ is a forgetting factor which can have a predefined constant value. Using a constant value for forgetting factor may lead to instability of the TSKF. Indeed estimation error covariance matrix may increase rapidly due to the recursive nature of the algorithm. One way to prevent “blow up” in TSKF is to keep $P_{k+1|k}^w$ in a certain range. Parkum et al. (1992) used a nonuniform forgetting factor in recursive least-square-based parameter identification schemes. The idea is that, under the normal system operation condition, the error covariance matrix $P_{k|k}^w$ describes the bias estimation error. The bias estimation error should be kept in a range means that the error covariance matrix should not be so large or so small. Inspired by this fact, an adaptive technique is suggested in Parkum et al. (1992) and considered later in Zhang and Wu (1999) and Wu et al. (2000). This adaptive technique forces the error covariance matrix $P_{k+1|k}^w$ to stay within pre-described limits:

$$\sigma_{\min} I \leq P_{k+1|k}^w \leq \sigma_{\max} I \quad (45.57)$$

where σ_{\min} and σ_{\max} are positive constants and I is the identity matrix. A dyadic expansion of $P_{k|k}^w$ is given as

$$P_{k|k}^w = \sum_{i=1}^l \alpha_{k|k}^i e_k^i (e_k^i)^T, \quad \|e_k^1\| = \dots = \|e_k^l\| = 1 \quad (45.58)$$

where e_k^1, \dots, e_k^l and $\alpha_k^1, \dots, \alpha_k^l$ are the eigenvectors and eigenvalues of $P_{k|k}^w$, respectively. Substituting (45.58) into (45.56) gives

$$P_{k+1|k}^w = \sum_{i=1}^l \frac{\alpha_{k|k}^i}{\lambda_k^i} e_k^i (e_k^i)^T + Q_k^w, \quad 0 < \lambda_k^i \leq 1 \quad (45.59)$$

Replacing (45.46) by (45.59), the TSKF will turn to ATSKF. As suggested in Parkum et al. (1992), the forgetting factor λ_k^i can be chosen as a decreasing function of the amount of information received in the direction e_k^i . Since eigenvalue $\alpha_{k|k}^i$ of $P_{k|k}^w$ is a measure of the uncertainty in the direction of e_k^i , a choice of forgetting factor λ_k^i based on the above constraints can be

$$\lambda_k^i = \begin{cases} 1 & \text{if } \alpha_{k|k}^i > \alpha_{\max} \\ \alpha_{k|k}^i \left[\alpha_{\min} + \frac{\alpha_{\max} - \alpha_{\min}}{\alpha_{\max}} \alpha_{k|k}^i \right]^{-1} & \text{if } \alpha_{k|k}^i \leq \alpha_{\max} \end{cases} \quad (45.60)$$

where α_{\max} and α_{\min} are positive constants. The choice of α_{\max} and α_{\min} is made by the designer. If the ratio $\alpha_{\max}/\alpha_{\min}$ is close to one, then the adaptive forgetting factor approaches a constant value.



Fig. 45.13 The NAV Lab of Concordia University

45.4 Experimental Results

The UKF, TSKF, and ATSKF have been tested on the Qball-X4 test bed of the Networked Autonomous Vehicles (NAV) lab of Concordia University (Fig. 45.13). Several experiments are carried in the presence of actuator faults, flight data is collected, and then the filters are applied to the data off-board to diagnose faults. The experiments are taking place indoor in the absence of GPS signals, and thus the OptiTrack camera system from NaturalPoint is employed to provide the system position in the 3D space. In all experiments, the system is required to hover at an altitude of 1 m, and the faults are taking place at time instant $t = 15$ s.

45.4.1 Single Fault Scenario: Control Effectiveness Loss in the Third Motor

In the first scenario, a loss of control effectiveness of 45 % is simulated in the third motor (i.e., $w_3 = 0.45$). As can be seen in Fig. 45.14a, this fault does not affect the system along the x -direction, but it results in loss of altitude of 20 cm and a deviation of 80 cm from the desired position along the y -direction. Due to the presence of a controller, the system recovers and goes back to the desired hover

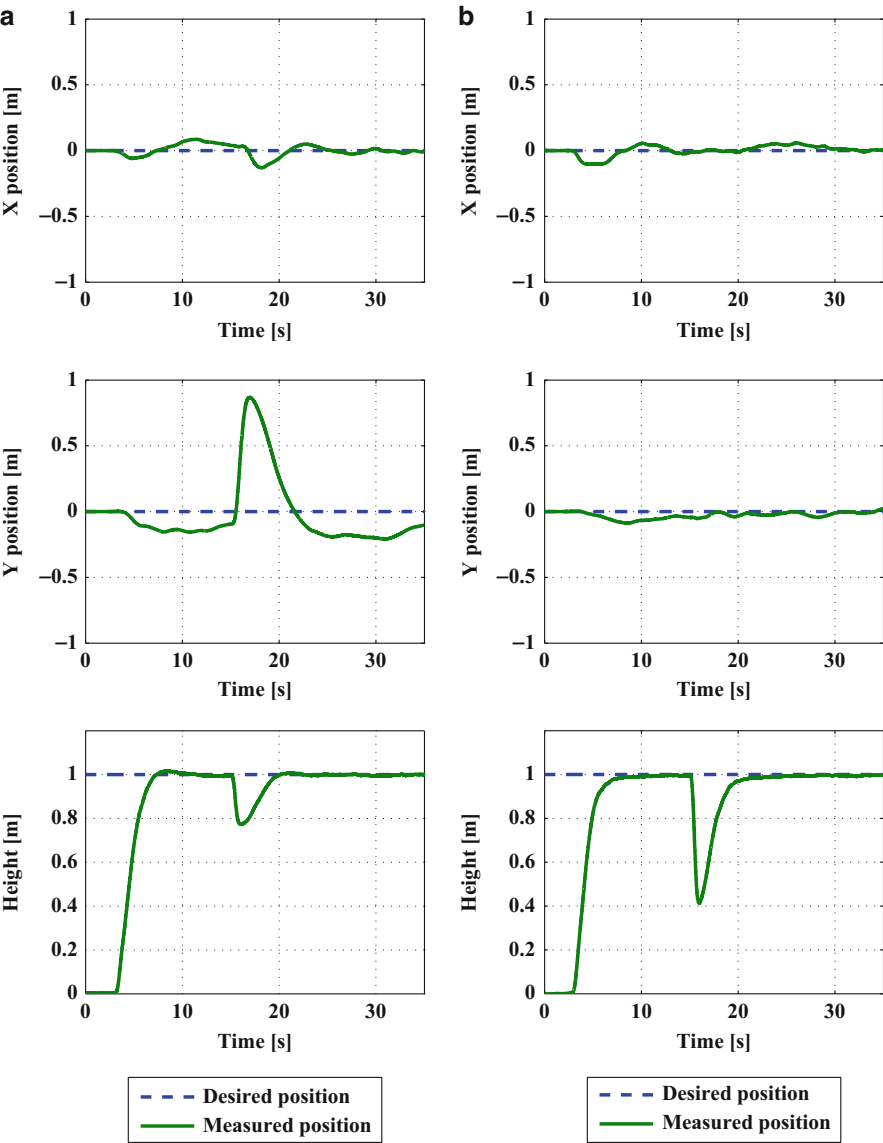


Fig. 45.14 System position along x , y , and z directions for single and simultaneous faults scenarios. (a) Single fault scenario, (b) Simultaneous faults scenario

position. Figure 45.15a shows the system’s behavior in the 3D space upon fault injection. The PWM inputs to the four motors are illustrated in Fig.45.16a. It is clear that before fault (up to 15 s), all the four PWM inputs are almost the same. After fault injection in the third motor, the baseline controller reacts by automatically increasing the third PWM input to compensate the occurred fault. The estimations

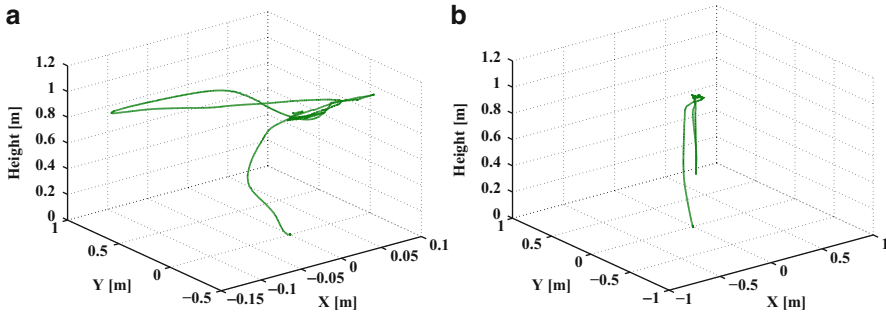


Fig. 45.15 System's 3D position for single and simultaneous faults scenarios, (a) Single fault scenario, (b) Simultaneous faults scenario

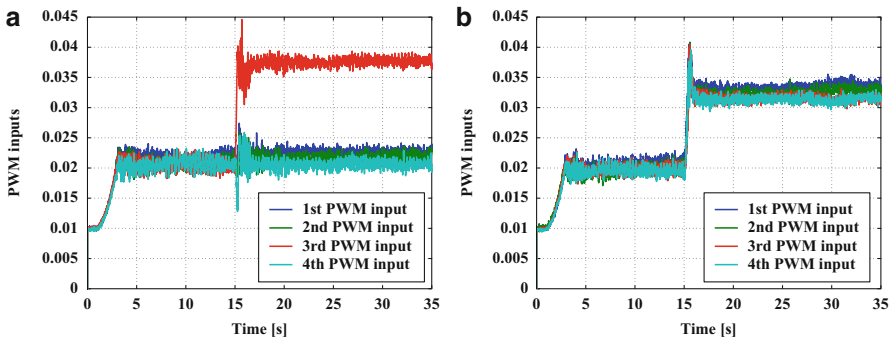


Fig. 45.16 The PWM inputs generated by the controller for single and simultaneous faults scenarios, (a) Single fault scenario, (b) Simultaneous faults scenario

of the actuator fault provided by the UKF, the TSKF, and the ATSKF are given in Figs. 45.17a, 45.18a, and 45.19a, respectively. The experimental application shows a fast and precise estimation of the fault amplitude despite model uncertainties. The estimates of w_1 , w_2 , and w_3 remain close to zero whereas that of w_3 converges to a value close to 0.45.

45.4.2 Simultaneous Faults Scenario: Control Effectiveness Loss in All Motors

In the second scenario, a loss of control effectiveness of 40 % is simulated in all motors (i.e., $w_i = 0.4$ for $i = 1, \dots, 4$). Unlike the previous case, this fault does not affect the system position in the x and y directions but results in a larger loss in altitude due to the feature of the faults, where the system drops to 0.4 m (see Fig. 45.14b). Figure 45.15b shows the system's behavior in the 3D space upon fault injection. The PWM inputs to the four motors are illustrated in Fig. 45.16b. Up to

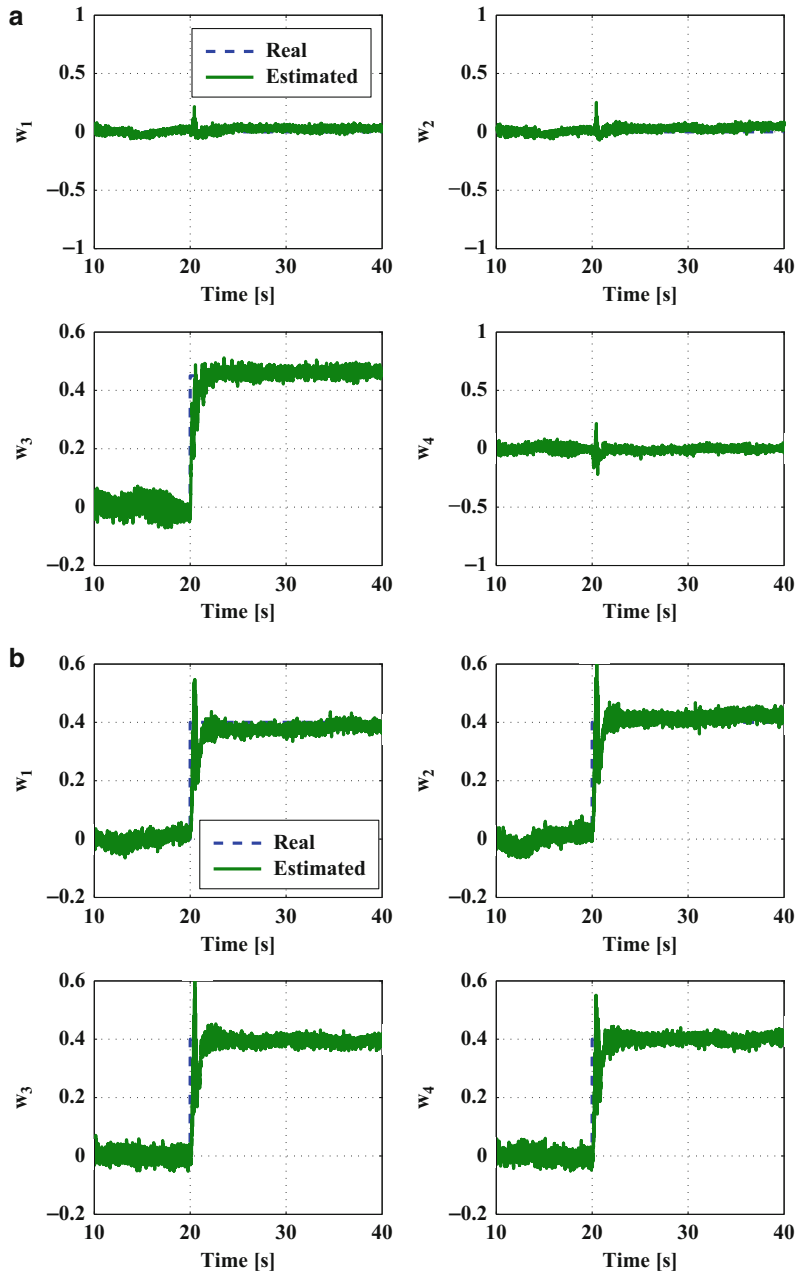


Fig. 45.17 UKF control effectiveness factors estimation for single and simultaneous faults scenarios. (a) Single fault scenario, (b) Simultaneous faults scenario

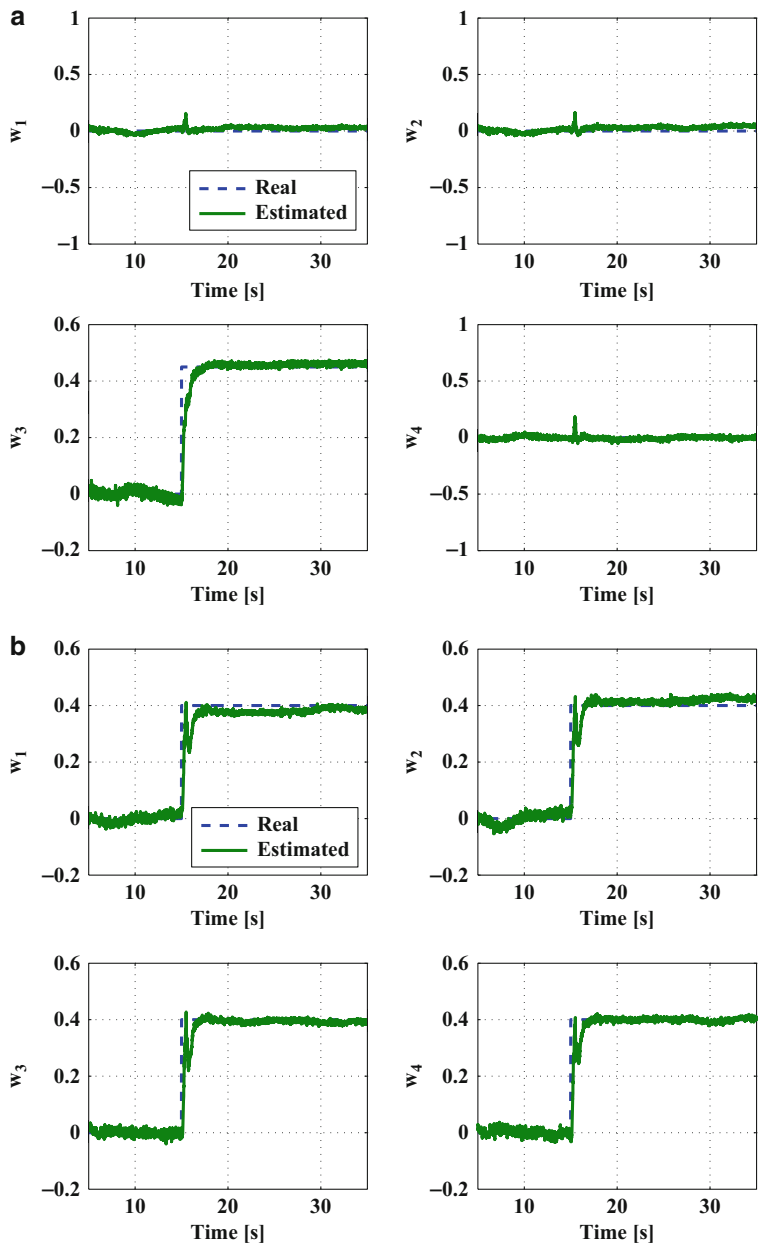


Fig. 45.18 TSKF control effectiveness factors estimation for single and simultaneous faults scenarios, (a) Single fault scenario, (b) Simultaneous faults scenario

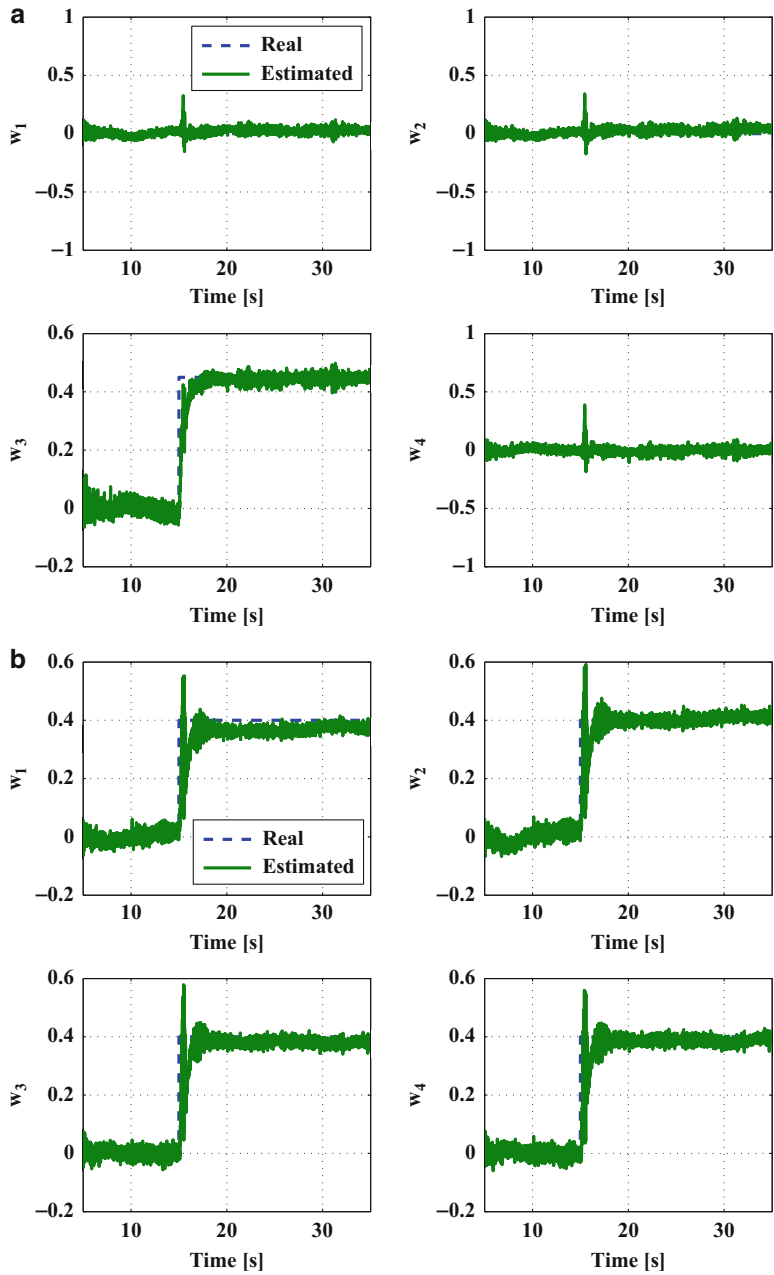


Fig. 45.19 ATSKF control effectiveness factors estimation for single and simultaneous faults scenarios. (a) Single fault scenario, (b) Simultaneous faults scenario

Table 45.1 Mean of execution times (in seconds) for 100 runs

| Fault case | UKF | TSKF | ATSKF |
|---------------------|-------------------------|-------------------------|-------------------------|
| Single fault | 2.8439×10^{-4} | 3.4478×10^{-4} | 7.2427×10^{-4} |
| Simultaneous faults | 2.8782×10^{-4} | 3.5102×10^{-4} | 7.3627×10^{-4} |

15 s and before fault injection, all the four PWM inputs are almost the same. After fault injection, the baseline controller automatically increases the PWM inputs to compensate the occurred faults. The estimation of the actuator faults is given in Figs. 45.17b, 45.18b, and 45.19b. Once again, the experimental application shows a fast and good estimation of all fault amplitudes despite model uncertainties.

45.4.3 Analysis of the Obtained Results

The application of UKF, TSKF, and ATSKF to the Qball-X4 has shown to be effective in estimating actuator faults. A comparison of the obtained results does not allow to draw an obvious superiority of one approach over the others. Moreover, fine-tuning the filter parameters can always improve the results, making them dependent on the gains selection. This is why a comparison which is based solely on Figs. 45.17–45.19 may not be sufficient. The ultimate objective of the present work is to implement the filters onboard the Qball-X4. Due to the limited calculation capabilities of the onboard microcomputer, the computational complexity of the filters is an important factor to investigate. In the current situation, the Gumstix embedded computer runs with a frequency of 200 Hz which is equivalent to a sample time of 0.005 s. In real-time systems, the timing behavior is an important property of each task. It has to be guaranteed that the execution of a task does not take longer than a specified amount of time. Thus, a knowledge about the maximum execution time of programs is of utmost importance (Puschner and Koza 1989). It turns out that each of the filters above is expected to execute in 0.005 s for every new set of data. For comparison purpose, the filters are implemented on a desktop computer running Intel Core i5 CPU with 2.67 GHz processing speed and 3 GB of RAM. Moreover, 100 different runs are carried out to reduce possible effects of programs running in the background. Table 45.1 gives a quantitative comparison of the mean time (in seconds) taken by each filter to execute. One can see that UKF is the fastest to execute with slight superiority over the TSKF mainly because UKF only considers the parameter estimation problem. On the other hand, the ATSKF is twice slower to execute than the TSKF. Figure 45.20 shows the execution times for each of the filters in the single fault case. The left-hand side of the figure illustrates the mean of the execution times for the 100 runs. The right-hand side shows one of the 100 runs and the execution time for each sample time.

45.5 Conclusion

This chapter considered the FDD and FTC problems for unmanned aerial vehicles. It presented a detailed review on the existing experimental works in these domains. The considered test beds can be categorized into two types: the fixed-wing as well

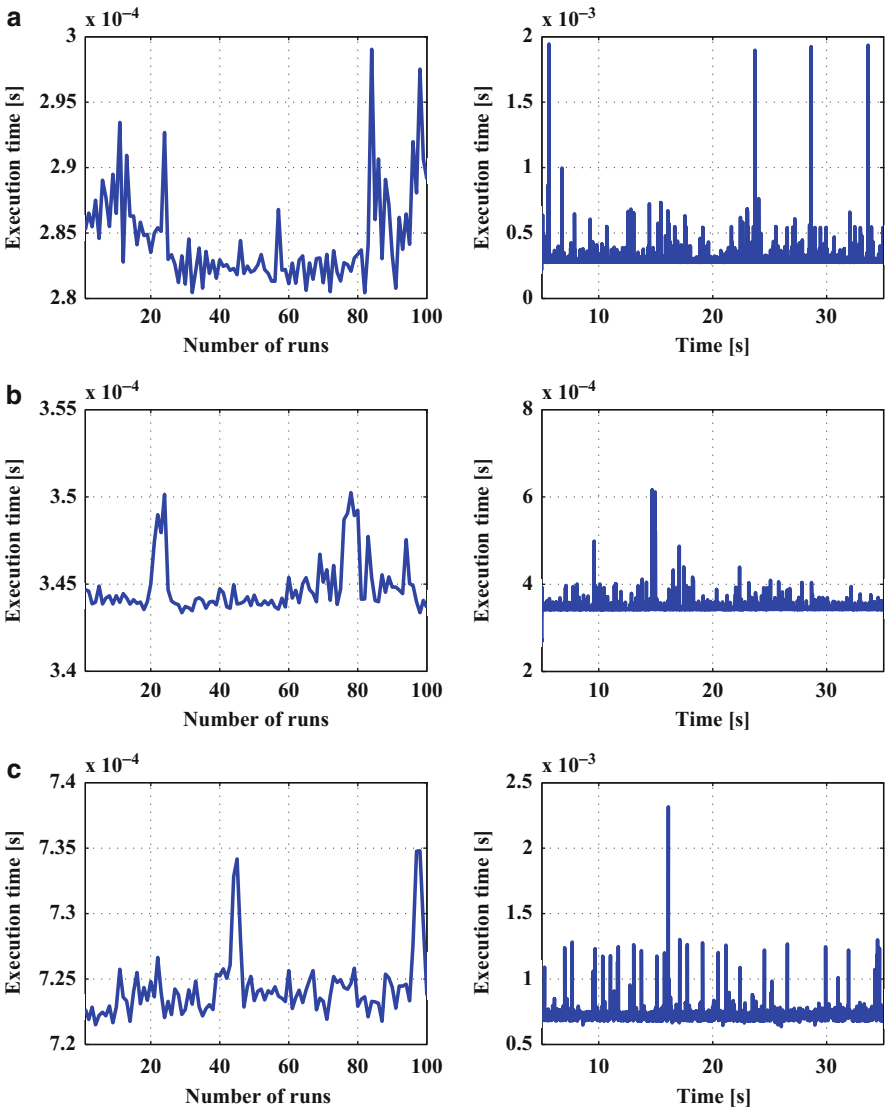


Fig. 45.20 Execution times of UKF, TSKF, and ATSKF in the single fault case, (a) The UKF execution time, (b) The TSKF execution time, (c) The ATSKF execution time

as the rotary-wing systems including the single-rotor and the multiple-rotor systems (also known as quadrotors). The chapter also presents some experimental results on the actuator FDD problem carried out at Concordia University and applied to a cutting-edge quadrotor test bed. The considered approaches are the UKF, the TSKF, and the ATSKF which have shown to be effective in estimating the actuator control effectiveness factors. A quantitative comparison is also given on the filter's execution times and their computational complexity to judge on their possible implementation onboard the quadrotor.

Acknowledgments This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship (PDF) program to the first author and the NSERC Strategic Project Grant and Discovery Project Grant led by the third author. Support from Quanser Inc. and colleagues from Quanser Inc. for the development of the Qball-X4 UAV test bed is also highly appreciated.

References

- AACLAB, online. Active Adaptive Control Laboratory, <http://aacclab.mit.edu/~aacclab/r/index.php>, [cited 15 May 2012]
- Adaptive Flight, online, www.adaptiveflight.com, [cited 15 May 2012]
- T. Bresciani, Modelling, identification and control of a quadrotor Helicopter. Master's thesis, Lund University, Sweden, 2008
- A. Chamseddine, LQR vs C-MRAC with a damage of 12% of propeller 2011a, www.youtube.com/watch?v=d6fnbZvcdmI, [cited 15 May 2012]
- A. Chamseddine, LQR vs C-MRAC with a damage of 16% of propeller 2011b, www.youtube.com/watch?v=NhI-UBi01jo, [cited 15 May 2012].
- A. Chamseddine, I. Sadeghzadeh, Fault-tolerance of the modified quadrotor helicopter based on control re-allocation 2011, www.youtube.com/watch?v=Kj3-pvKWiNk, [cited 15 May 2012]
- G. Chowdhary, S. Srinivasan, E. Johnson, Frequency domain method for real-time detection of oscillations. *AIAA J. Aerosp. Comput. Inf. Commun.* **8**(2), 42–52 (2011)
- R.V. der Merwe, E. Wan, The square-root unscented Kalman filter for state and parameter estimation, in *Proceedings of the International Conference on Acoustics, Speech and Signal Process*, Maui, HI, 7–11 May 2001, pp. 3461–3464
- S.X. Ding, *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools* (Springer, Berlin/New York, 2008)
- G.R. Drozeski, A fault-tolerant control architecture for unmanned aerial vehicles. Ph.D. thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2005
- G.R. Drozeski, B. Saha, G.J. Vachtsevanos, A fault detection and reconfigurable control architecture for unmanned aerial vehicles, in *IEEE Aerospace Conference*, Big Sky, MT, USA, 5–12 March 2005, pp. 1–9
- Z. Dydek, Adaptive control of a quadrotor UAV (2010a), www.youtube.com/watch?v=FhgMy4ss0bw, [cited 15 May 2012]
- Z. Dydek, Altitude regulation for quadrotor UAVs (2010b), www.youtube.com/watch?v=Ti239r5MquI, [cited 15 May 2012]
- FTC, Fault-tolerant flight control with neural networks (2011), www.youtube.com/watch?v=K12Fm5s6YCA, [cited 15 May 2012]
- Georgia Tech, Airplane flies with 50% of wing missing (2009a), www.youtube.com/watch?v=iGxpXNi9Xh0, [cited 15 May 2012]
- Georgia Tech, Airplane flies with 50% of wing missing, take 2 (2009b), www.youtube.com/watch?v=4kUKdTsaKJK, [cited 15 May 2012]

- Y. Gu, Design and flight testing actuator failure accommodation controllers on WVU YF-22 research UAVs. Ph.D. thesis, College of Engineering and Mineral Resources, West Virginia University, 2004
- G. Heredia, A. Ollero, M. Bejar, R. Mahtani, Sensor and actuator fault detection in small autonomous helicopters. *Mechatronics* **18**(2), 90–99 (2008)
- IFCS, online. The Intelligent Flight Control System Project, www.nasa.gov/centers/dryden/news/FactSheets/FS-048-DFRC.html, [cited 15 May 2012]
- R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance* (Springer, Berlin/New York, 2006)
- S.J. Julier, J.K. Uhlmann, A new extension of the Kalman filter to nonlinear systems, in *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, Orlando, FL, USA, 20–25 April 1997, pp. 182–193
- J.Y. Keller, M. Darouach, Optimal two-stage Kalman filter in the presence of random bias. *Automatica* **33**(9), 1745–1748 (1997)
- L. Ma, Y.M. Zhang, DUKF-based fault detection and diagnosis for GTM UAV using nonlinear and LPV models, in *Proceedings of the 6th ASME/IEEE International Conference on Mechatronic & Embedded Systems & Applications*, Qingdao, P. R. China, 15–17 July 2010
- MAST, online. The Micro Autonomous Systems Technologies Project, http://www.grasp.upenn.edu/research/micro_autonomous_system_technologies_mast, [cited 15 May 2012]
- M.R. Napolitano, YF22 model with on-board on-line learning microporcessors-based neural algorithms for autopilot and fault-tolerant flight control systems. Tech. rep., West Virginia University Research Corporation, April 2002
- NAV, online. The networked autonomous vehicles laboratory, <http://users.encs.concordia.ca/~ymzhang/UAVs.htm>, [cited 15 May 2012]
- Office of the Secretary of Defense, Unmanned aerial vehicles roadmap 2002–2027. Tech. rep., Washington, DC 20301, 2002
- J.E. Parkum, N.K. Poulsen, J. Holst, Recursive forgetting algorithms. *Int. J. Control* **55**, 109–128 (1992)
- P. Puschner, C. Koza, Calculating the maximum execution time of real-time programs. *Real-Time Syst.* **1**, 159–176 (1989). doi:10.1007/BF00571421, <http://dx.doi.org/10.1007/BF00571421>
- J.T. Qi, J.D. Han, Fault diagnosis and fault-tolerant control of rotorcraft flying robots: a survey. *CAAI Trans. Intell. Syst.* **2**(2), 31–39 (2007)
- J. Qi, D. Song, L. Dai, J. Han, Design, implement and testing of a rotorcraft UAV system, in *Aerial Vehicles*, ed. by T.M. Lam (InTech Open Access Publisher, Rijeka, 2009), pp. 537–554
- Quanser, Quanser Qball-X4 User Manual, document number 829 (2010)
- RESTORE, online. Reconfigurable Control for Tailless Fighter Aircraft Program, www.nasa.gov/centers/dryden/multimedia/imagegallery/X-36/X-36_proj_desc.html, [cited 15 May 2012]
- Rockwell Collins, Rockwell Collins successfully controls and lands damaged UAV (2008), www.youtube.com/watch?v=QJkIONTzbNM, [cited 15 May 2012]
- Rockwell Collins, Rockwell Collins damage tolerance phase II (2009), www.youtube.com/watch?v=dGiPNV1TR5k, [cited 15 May 2012]
- Rockwell Collins, Damage tolerance flight test: 60% wing loss and stabilizer losses (2010a), www.youtube.com/watch?v=PTMpq.8SSCI, [cited 15 May 2012]
- Rockwell Collins, Damage tolerance flight test: 80% wing loss (2010b), www.youtube.com/watch?v=xN9f9ycWkOY, [cited 15 May 2012]
- Rockwell Collins, DARPA Shadow UAV damage-tolerant flight control test (2010c), www.youtube.com/watch?v=hJ2vJGTesgc, [cited 15 May 2012]
- I. Sadeghzadeh, Y.M. Zhang, Review on fault tolerant control for unmanned aerial vehicles (UAVs), in *AIAA InfoTech@Aerospace 2011: Unleashing Unmanned Systems*, St. Louis, MO, USA, 29–31 March 2011
- D. Shore, M. Bodson, Flight testing of a reconfigurable control system on an unmanned aircraft, in *Proceeding of the 2004 American Control Conference*, Boston, MA, June 30–July 2, 2004, pp. 3747–3452

- STARMAC, online. The Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control Project, <http://hybrid.stanford.edu/~starmac/project.htm>, [cited 15 May 2012]
- SWARM, online. UAV SWARM Health Management Project, <http://vertol.mit.edu>, [cited 15 May 2012]
- M.C. VanDyke, J.L. Schwartz, C.D. Hall, Unscented Kalman filtering for spacecraft attitude state and parameter estimation, in *Proceedings of the 14th AAS/AIAA Space Flight Mechanics Meeting* Maui, HI, 8–12 Feb 2004
- A. Vitale, F. Corraro, M. Bernard, G.D. Matteis, Unscented Kalman filtering for reentry vehicle identification in the transonic regime. *J. Aircr.* **46**(5), 1649–1659 (2009)
- N.E. Wu, Y.M. Zhang, K.M. Zhou, Control effectiveness estimation using an adaptive Kalman estimator, in *Proceedings of IEEE ISIC/CIRA/ISAS Joint Conference*, Gaithersburg, MD, USA, 14–17 Sept 1998, pp. 181–186
- N.E. Wu, Y.M. Zhang, K.M. Zhou, Detection, estimation, and accommodation of loss of control effectiveness. *Int. J. Adapt. Control Signal Process. Special Issue: Fault Detection and Isolation* **14**(7), 775–795 (2000)
- Y.M. Zhang, A. Chamseddine, Fault Tolerant Flight Control Techniques with Application to a Quadrotor UAV Testbed, in *Automatic Flight Control Systems-Latest Developments* ed. by T. Lombaerts (InTech Open Access Publisher, Rijeka, 2011), pp. 119–150.
- Y.M. Zhang, J. Jiang, An active fault-tolerant control system against partial actuator failures. *IEE Proc. Control Theory Appl.* **149**(1), 95–104 (2002)
- Y.M. Zhang, J. Jiang, Bibliographical review on reconfigurable fault-tolerant control systems. *IFAC Annu. Rev. Control* **32**(2), 229–252 (2008)
- Y.M. Zhang, N.E. Wu, Fault diagnosis for a ship propulsion benchmark: part I, in *Preprints of the 14th IFAC World Congress*, Beijing, China, 5–9 July 1999

Fault Detection and Diagnosis for NASA GTM UAV with Dual Unscented Kalman Filter

46

Youmin M. Zhang

Contents

| | | |
|--------|---|------|
| 46.1 | Introduction | 1158 |
| 46.2 | Nonlinear GTM and LPV Model of Nonlinear Longitudinal Motion of the GTM | 1160 |
| 46.2.1 | Brief Description of the Nonlinear Dynamic Model of GTM | 1160 |
| 46.2.2 | LPV Model of the GTM | 1161 |
| 46.3 | Unscented Kalman Filter (UKF) and Dual Unscented Kalman Filter (DUKF) | 1163 |
| 46.3.1 | Unscented Transformation (UT) | 1164 |
| 46.3.2 | State Estimation | 1166 |
| 46.3.3 | Parameter Estimation | 1167 |
| 46.3.4 | The DUKF Estimation | 1168 |
| 46.4 | Fault Detection and Diagnosis Based on Dual Unscented Kalman Filter | 1170 |
| 46.4.1 | Brief Introduction to FDD | 1170 |
| 46.4.2 | Residual Generation | 1170 |
| 46.4.3 | Residual Evaluation/Decision-Making | 1172 |
| 46.5 | Simulation Results | 1174 |
| 46.5.1 | FDD Results Based on Nonlinear Model | 1174 |
| 46.5.2 | FDD Results Based on LPV Model | 1177 |
| 46.5.3 | Comparison of DUKF with TSKF/ATSEKF | 1177 |
| 46.6 | Conclusion | 1179 |
| | References | 1180 |

Abstract

This chapter investigates a simultaneous state and parameter estimation-based fault detection and diagnosis (FDD) scheme for application to a realistic nonlinear six degree-of-freedom fixed-wing unmanned aerial vehicle (UAV) model: the NASA GTM (generic transport model). The NASA GTM UAV model was developed in the MATLAB/Simulink environment under the NASA Aviation

Y.M. Zhang

Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

e-mail: Youmin.Zhang@concordia.ca; ymzhang@encs.concordia.ca

Safety Program (AvSP). By introducing partial loss faults in actuators into the NASA GTM, a dual unscented Kalman filter (DUKF) algorithm is applied for the purpose of online estimation of both flight states and fault parameters. A Bayesian rule is then used for detection and isolation decision-making. The developed FDD scheme has been implemented in both the nonlinear GTM and the linear parameter-varying (LPV) representation of the nonlinear GTM. Simulation results show satisfactory results for detecting and diagnosing the control effectors faults.

46.1 Introduction

Loss of control has become one of the main causes of airplane crashes and crash-related fatal accidents worldwide (see Belcastro 2011 and the references therein). Thus, it is necessary to enable airplanes to increase the fault tolerance capability when unexpected faults occur in the airplane during flight. When a fault occurs in an airplane, the first and main problem to be solved is to detect what and where the fault is, to determine the severity level (magnitude) of the fault, and then to provide a solution for handling or compensating for the fault-induced changes of the airplane to guarantee at least a safe landing. This has been the motivation and topics for fault detection and diagnosis (FDD) and fault-tolerant control (FTC) in aviation industry since the 1980s (Zhang and Jiang 2008; Sadeghzadeh and Zhang 2011). A good FDD scheme should be able to report detailed information for the post-fault system as quick and accurate as possible. As one of most recent efforts by NASA on the purpose of investigating flight dynamics, control, and behavior of aircraft under upset conditions, NASA has built a test bed, which is known as the AirSTAR Flight Test Facility, where an integral part of which is the GTM (generic transport model) unmanned aerial vehicle (UAV) (Jordan et al. 2006). The NASA GTM UAV is a 5.5 % dynamically scaled, turbine powered fixed-wing civil transport aircraft. Associated with the NASA Aviation Safety Program (AvSP), researches on adaptive control, fault-tolerant control, health monitoring and fault diagnosis, safe envelope assessment, and loss-of-control analysis have been carried out recently based on the GTM software simulation environment and real flight tests. A few examples of many relevant works are given in Pandita et al. (2009), Ruschmann et al. (2009), Ma and Zhang (2010a, b), Guo and Tao (2011, 2012), Kwatny et al. (2009), Gregory et al. (2010), Hovakimyan et al. (2011), Belcastro (2011), and Xu and Liu (2011). In this chapter, however, development and application of an FDD scheme is mainly presented based on the work carried out in Ma and Zhang (2010a, b) and Ma (2011), which is based on the nonlinear and linear parameter varying (LPV) model (Shin 2000; Marcos and Balas 2004; Pandita et al. 2009) of the GTM UAV in the event of actuator faults/failures. The FDD scheme used is a dual unscented Kalman filter (DUKF) (Julier and Uhlmann 1997; Wan and van der Merwe 2000) with real-time fault parameter estimation based on the measured outputs of the sensors and the control inputs to actuators.

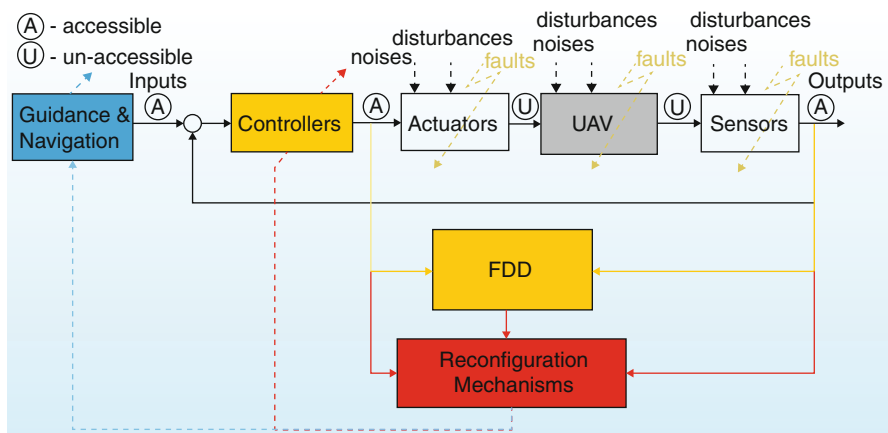


Fig. 46.1 A general structure of an active fault-tolerant control system

For a high performance FDD/FTC scheme, when a fault/failure/damage occurs in either an actuator, a sensor, or a component of the airplane/UAV, the FDD scheme should detect and diagnose the source and the magnitude of the fault timely and precisely. The reconfiguration scheme for FTC will design a reconfigurable/fault-tolerant controller based on the information provided by the FDD scheme to adapt post-fault airplane to an appropriate new operating condition under the fault/failure/damage. In the case of a new flight path for a safe landing to a nearest airport or landing area is needed, reconfiguration for guidance and navigation law (or reconfigurable guidance and navigation law (RGNL)) should also be needed. Therefore the entire airplane/UAV can still maintain its stability and achieve probably degraded but acceptable level of performance. Figure 46.1 depicts the general structure of such an active fault-tolerant control system (AFTCS) with the functional block diagrams of four main modules (instead of traditional three main modules as outlined in Zhang and Jiang (2008)—with insertion of a new RGNL block in the overall AFTCS). In this chapter, however, only the FDD functional block is considered with the overall content arrangement of this handbook. More specifically, an FDD scheme based on a dual unscented Kalman filter (DUKF) algorithm is described with application to the NASA GTM UAV.

The chapter is organized as follows. In Sect. 46.2, the nonlinear GTM and the LPV model of the nonlinear longitudinal GTM model will be briefly introduced. In Sect. 46.3, the UKF and DUKF are presented. In Sect. 46.4, an FDD scheme based on the DUKF is described. In Sect. 46.5, the simulation results for the nonlinear GTM and the LPV model of GTM are presented. Brief conclusion is given in Sect. 46.6.

46.2 Nonlinear GTM and LPV Model of Nonlinear Longitudinal Motion of the GTM

46.2.1 Brief Description of the Nonlinear Dynamic Model of GTM

In this section, brief introduction to the nonlinear NASA GTM UAV model and the LPV model representation of the nonlinear longitudinal motion of the GTM will be given.

The GTM is a dynamically scaled small fixed-wing unmanned aerial vehicle model developed by NASA to investigate modeling and control of large transport vehicles in upset conditions. The UAV is 5.5 % dynamically scaled to realistically simulate characteristics of a full-scale large civil transport jet aircraft.

The nonlinear dynamic equations of the GTM are

$$\dot{u} = [C_{X_{\alpha,\beta}} + C_{X\hat{q}} + C_{X_{\delta e}} + C_{X_{\delta T}} + (C_{X_{\delta r}} + C_{X_{\delta a}})] \frac{S\bar{q}}{m} - g \sin \theta + rv - qw \quad (46.1)$$

$$\dot{w} = [C_{Z_{\alpha,\beta}} + C_{Z\hat{q}} + C_{Z_{\delta e}} + C_{Z_{\delta T}} + (C_{Z_{\delta r}} + C_{Z_{\delta a}})] \frac{S\bar{q}}{m} + g \cos \theta \sin \phi - ru + pw \quad (46.2)$$

$$\begin{aligned} \dot{q} = & [C_{m_{\alpha,\beta}} + C_{m\hat{q}} + C_{m_{\delta e}} + C_{m_{\delta T}} + (C_{m_{\delta r}} + C_{m_{\delta a}})] \frac{S\bar{q}\bar{c}}{I_{yy}} + rp(I_{zz} - I_{yy}) \\ & + \frac{I_{xz}(p^2 - r^2)}{I_{yy}} \end{aligned} \quad (46.3)$$

$$\dot{v} = [C_{Y_{\alpha,\beta}} + C_{Y\hat{p}} + C_{Y_{\delta r}} + C_{Y_{\delta a}} + (C_{Y_{\delta e}} + C_{Y_{\delta \hat{r}}})] \frac{S\bar{q}}{m} + g \cos \theta \cos \phi + qu - pv \quad (46.4)$$

$$D = [C_{l_{\alpha,\beta}} + C_{l\hat{p}} + C_{l_{\delta r}} + C_{l_{\delta a}} + (C_{Y_{\delta e}} + C_{Y_{\delta \hat{r}}})] S\bar{q}b + rp(I_{zz} - I_{yy}) + I_{xz} \quad (46.5)$$

$$E = [C_{m_{\alpha,\beta}} + C_{m_{\hat{p}}} + C_{m_{\delta r}} + C_{m_{\delta a}} + (C_{m_{\delta e}} + C_{m_{\delta \hat{r}}})] S\bar{q}b + pq(I_{yy} - I_{xx}) + I_{xz} \quad (46.6)$$

$$\dot{p} = \frac{I_{yy}I_{zz}D + I_{xz}I_{yy}E}{I_{yy}I_{xx} + I_{xz}^2I_{yy}} \quad (46.7)$$

$$\dot{r} = \frac{I_{yy}I_{xz}D + I_{xx}I_{yy}E}{I_{yy}I_{xx} + I_{xz}^2I_{yy}} \quad (46.8)$$

$$\dot{\phi} = p + (q \sin \phi \sin \theta + r \cos \phi \sin \theta) / \cos \theta \quad (46.9)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (46.10)$$

$$\dot{\psi} = (q \sin \phi + r \cos \phi) / \cos \theta \quad (46.11)$$

In order to simplify the dynamic equations, the following assumptions have been made:

1. $I_{xy} = 0$ and $I_{yz} = 0$.
 2. Mass is assumed to remain constant.
 3. The center of gravity is assumed to be constant and fixed at the nominal value.
- Lift, drag, and pitching moment coefficients are implemented by lookup tables in the MATLAB/Simulink environment.

The state variables of the six degree-of-freedom UAV are defined as

$$x = [u \ v \ w \ p \ q \ r \ \phi \ \theta \ \psi]^T$$

46.2.2 LPV Model of the GTM

As the emerging approach to control theory, linear parameter-varying (LPV) modeling and control of nonlinear systems have been widely studied in dynamic systems since the early 1990s (Shin 2000; Marcos and Balas 2004; Mohammadpour and Scherer 2012). Applications have been successful in certain cases. An LPV model simulates the actual nonlinear system by using time-varying real parameters like altitude and/or speed of an airplane to obtain smooth semi-linear models. However, there are a few investigations for FDD design with LPV model. Since FDD in this work is a model-based estimation, the LPV model is chosen for the FDD design to the nonlinear model of the GTM as well due to real-time implementation consideration.

For the sake of brevity, the basic definition of an LPV model for a nonlinear system is presented. LPV model is a class of finite-dimensional linear models of a nonlinear system whose state-space matrixes A , B , C , and D depend continuously on a time-varying parameter vector $\xi(t)$. Parameters $\xi(t)$ can be measured at the current time, and their values are constrained a priori to lie in some known, bounded set, and this set is continuous, but cannot be known in advance. The formal definition of an LPV model can be given as follows:

Given a compact subset $\mathcal{P} \subset \mathcal{R}^s$, the parameter variation set $\mathcal{F}_{\mathcal{P}}$ denotes the set of all piecewise continuous functions mapping \mathcal{R}^+ (time) into ξ with a finite number of discontinuities in any interval.

Given continuous functions, $A : \mathcal{R}^s \rightarrow \mathcal{R}^{n \times n}$, $B : \mathcal{R}^s \rightarrow \mathcal{R}^{n \times n_u}$, $C : \mathcal{R}^s \rightarrow \mathcal{R}^{n_y \times n}$, $A : \mathcal{R}^s \rightarrow \mathcal{R}^{n_y \times n_u}$, an n th-order LPV model of a given nonlinear system is defined as

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\xi(t)) & B(\xi(t)) \\ C(\xi(t)) & D(\xi(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \quad (46.12)$$

where $\xi(t) \in \mathcal{F}_{\mathcal{P}}$.

Now the key task is to select an adequate $\xi(t)$ such that the above LPV model is able to capture the nonlinearities of the system. So far, there are three techniques for

obtaining LPV models from a nonlinear system (Marcos and Balas 2004; Pandita et al. 2009; Kwatny et al. 2009).

- *Jacobian linearization approach.* It is implemented at a number of selected equilibrium points. The Jacobian linearization approach uses first-order Taylor series expansion of nonlinear model to create an LPV model which approximates the nonlinear system with respect to selected equilibrium points. The drawback of this approach is that it is easy to be divergent and hard to obtain the transient behavior of the nonlinear system.
- *State transformation approach.* This approach is proposed based on exact state transformations at a number of selected equilibrium points. It requires that $\xi(t)$ must be available in real time for measurement. The disadvantage of this approach is that the existence of trim map for the entire flight envelope of interest for a particular combination of the scheduling variables is not guaranteed.
- *The function substitution approach.* This approach is to obtain an LPV model at a unique trim point by decomposing the nonlinear function. The main drawback of this approach is the lack of theoretical validation.

For developing the LPV model based on the nonlinear GTM in longitudinal motion, the following nonlinear equations of motion of the GTM can be given as (Pandita et al. 2009)

$$V_{EAS} = \frac{1}{m} (F_x \cos \alpha + F_z \sin \alpha) \quad (46.13)$$

$$\dot{\alpha} = \frac{1}{m V_{EAS}} (-F_x \sin \alpha + F_z \cos \alpha) + q \quad (46.14)$$

$$\dot{\theta} = q \quad (46.15)$$

$$\dot{q} = \frac{M_y}{I_{yy}} \quad (46.16)$$

These equations contain transcendental functions and aerodynamic data which are obtained through wind tunnel testing and flight tests. The transcendental functions can be approximated by third-order Taylor series expansion as follows:

$$\sin x = x - \frac{1}{6}x^3 \quad (46.17)$$

$$\cos x = 1 - \frac{1}{2}x^2 \quad (46.18)$$

The aerodynamic data which are obtained by using lookup tables in the nonlinear model of the GTM can be approximated by polynomial equations (Ruschmann et al. 2009).

The LPV model of the longitudinal motion of the GTM has state variables $x = [\text{EAS } \alpha \ \theta \ q]^T$ with equivalent airspeed (EAS), angle of attack (α), pitch angle (θ), and pitch angular rate (q). The input vector u is given by $u = [\delta_e \ \delta_{th}]^T$, with δ_e

representing elevator deflection and δ_{th} representing throttle deflection. Therefore aerodynamic forces F_x and F_z and moment M_y are obtained through the following equations:

$$F_x = \bar{q} S_{\text{ref}} [C_x(\alpha) + C_x(\alpha, \delta_e) + C_x(\alpha, \hat{q})] + 2T_x(\delta_{th}) - mg \sin \theta \quad (46.19)$$

$$F_z = \bar{q} S_{\text{ref}} [C_z(\alpha) + C_z(\alpha, \delta_e) + C_z(\alpha, \hat{q})] + 2T_z(\delta_{th}) + mg \cos \theta \quad (46.20)$$

$$M_y = \bar{q} S_{\text{ref}} \bar{c} [C_m(\alpha) + C_m(\alpha, \delta_e) + C_m(\alpha, \hat{q})] + 2\Delta Z_{\text{ENG}} T_x(\delta_{th}) \quad (46.21)$$

where the aerodynamic coefficients of angle of attack α , pitch rate q , and elevator deflection δ_e are obtained through lookup tables.

For FDD purpose, partial loss of control effectiveness fault in elevator has been implemented in this work from the original LPV model of the GTM (Pandita et al. 2009; Ruschmann et al. 2009).

46.3 Unscented Kalman Filter (UKF) and Dual Unscented Kalman Filter (DUKF)

For nonlinear aircraft systems, the extended Kalman filter (EKF) has been widely applied (Zhang and Jiang 2008; VanDyke et al. 2004). The EKF is the nonlinear version of the Kalman filter (KF), and it linearizes nonlinear system about the current equilibrium point with the characteristics of random state variables described by mean and covariance. Hence, the EKF can only preserve the first-order system statistics and may quickly diverge if the system is not modeled correctly also due to the use of linearized model of the nonlinear system. The UKF is an improvement of the EKF to replace the EKF in nonlinear filtering problems. UKF was originally developed by Wan and van der Merwe (2000) and Julier and Uhlmann (1997), and it has been mainly used to nonlinear systems estimation, identification, and training of neural networks, etc.

The UKF picks a minimal set of sample points which are called *sigma points* around the mean by the unscented. These sigma points are then propagated through nonlinear functions and the covariance of the estimate is then recovered. In the UKF, the probability density is approximated by the nonlinear transformation of a random variable, which returns much more accurate results than the first-order Taylor expansion of the nonlinear functions used in the EKF. The approximation utilizes a set of sample points (*sigma points*), which captures the posterior mean and covariance accurately to the third-order Taylor series expansion for any nonlinearity (Wan and van der Merwe 2001). Therefore, the UKF captures both the first-order and second-order statistics of a nonlinear system. It has been demonstrated that the UKF has better filter performance compared with EKF and is equivalent to the performance of second-order EKF (Wan and van der Merwe 2000; Julier and

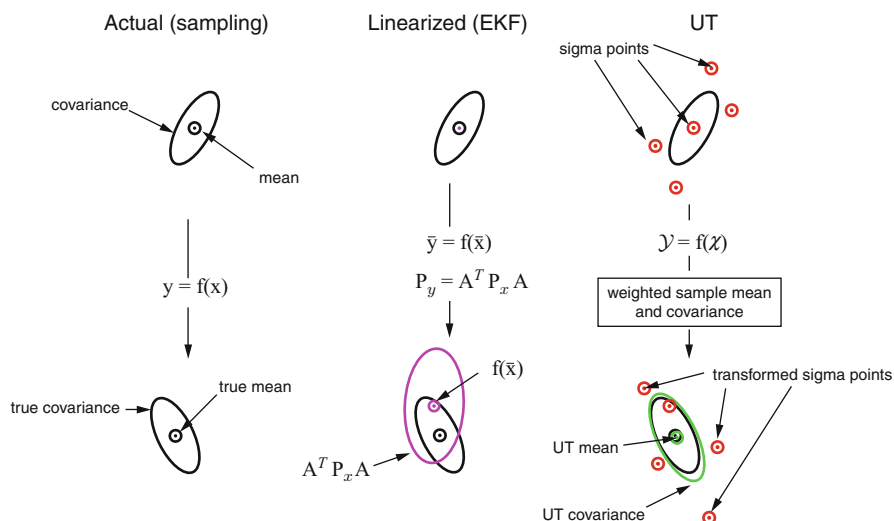


Fig. 46.2 Example of the UT for mean and covariance propagation (Wan and van der Merwe 2001)

Uhlmann 1997). Figure 46.2 shows an example of the unscented transformation (UT) for mean and covariance propagation.

From Fig. 46.2, one can see that the result of UKF is more accurate to capture the true mean and covariance than other Kalman filters can do.

The main motivation for using UKF to FDD application resides in the advantages and the superiority of this approach over the EKF and more specifically the better convergence characteristics and greater accuracy in the case of nonlinear systems (VanDyke et al. 2004). However, to provide good estimation to the unknown parameters in the system due to, for example, fault-induced parameter changes, a separate estimation to the unknown parameters is needed. In other words, an estimation scheme for simultaneous states and parameters based on UKF is needed. Inspired with EKF for such a simultaneous states and parameter estimation issue, a dual unscented Kalman filter (DUKF) scheme was proposed in Wan and van der Merwe (2001). This in fact motivated the research for using such a DUKF to the GTM fault detection and diagnosis application (Ma and Zhang 2010a; Ma 2011).

46.3.1 Unscented Transformation (UT)

The unscented transformation is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation (Wan and van der Merwe 2001). It is built upon the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function. The approach is illustrated in Fig. 46.3.

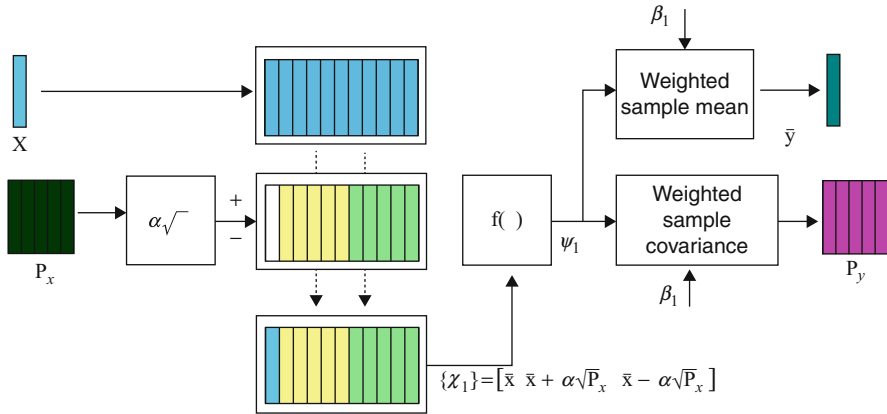


Fig. 46.3 The principle of the unscented transform (Wan and van der Merwe 2001)

Consider a nonlinear model $y = f(x)$, and the random variable x whose dimension is L , and assume x has mean \bar{x} and covariance P_x . The basic steps of UT are:

- Choose a set of points (sigma points) so that their sample mean and sample covariance are \bar{x} and P_x .
- Apply each sigma point in turn to the nonlinear function to yield a cloud of transformed points, and \bar{y} and P_y are the statistics of the transformed points.

The L -dimensional random variable x with mean \bar{x} and covariance P_x is approximated by $2L + 1$ sigma points. In order to calculate the statistics of y , χ , which is the matrix of $2L + 1$ sigma vectors χ_i , is formed:

$$\chi_0 = \bar{x} \quad (46.22)$$

$$\chi_i = \bar{x} + \left(\sqrt{(L + \lambda) P_x} \right)_i, \quad i = 1, \dots, L \quad (46.23)$$

$$\chi_i = \bar{x} - \left(\sqrt{(L + \lambda) P_x} \right)_i, \quad i = L + 1, \dots, 2L \quad (46.24)$$

The weights for the state and covariance are given by

$$\sigma_0^{(m)} = \frac{\lambda}{L + \lambda} \quad (46.25)$$

$$\sigma_0^{(c)} = \frac{\lambda}{L + \lambda} + (1 - \rho^2 + \sigma) \quad (46.26)$$

$$\sigma_i^{(m)} = \sigma_i^{(c)} = \frac{1}{2(L + \lambda)}, \quad i = 1, \dots, 2L \quad (46.27)$$

where $\lambda = \rho^2(L + \kappa) - L$ is a scaling parameter. ρ determines the spread of the sigma points around \bar{x} . The κ is a secondary scaling parameter, and σ is to incorporate prior knowledge of the distribution of x . $\left(\sqrt{(L + \lambda) P_x}\right)_i$ is the i th column of the matrix square root. In this book chapter, ρ , σ , and κ are equal to 1, 2, and $3 - L$, respectively. χ_i is propagated through the nonlinear function:

$$\mu_i = f(\chi_i), \quad i = 0, \dots, 2L \quad (46.28)$$

$$\bar{y} \approx \sum_{i=0}^{2L} \sigma_i^{(m)} \mu_i \quad (46.29)$$

$$P_y \approx \sum_{i=0}^{2L} \sigma_i^{(m)} (\mu_i - \bar{y}) (\mu_i - \bar{y})^T \quad (46.30)$$

where \bar{y} is the mean for y and P_y is the covariance of the posterior sigma points. Figure 46.3 presents the implementation of UT.

46.3.2 State Estimation

Consider a nonlinear transform of a random variable: $y = f(x)$

$$\text{Given: } \bar{x} = E[x], P_x = E[(x - \bar{x})(x - \bar{x})^T]$$

$$\text{Find: } \bar{y} = E[y], P_y = E[(y - \bar{y})(y - \bar{y})^T].$$

A set of $2L + 1$ sigma points is derived as following equations from the augmented state and covariance where L denotes the dimension of the augmented state.

$$\chi_{k-1|k-1}^0 = \chi_{k-1|k-1}^\rho \quad (46.31)$$

$$\chi_{k-1|k-1}^i = \chi_{k-1|k-1}^\rho + \left(\sqrt{(L + \lambda) P_{k-1|k-1}^\rho}\right)_i, \quad i = 1, \dots, L \quad (46.32)$$

$$\chi_{k-1|k-1}^i = \chi_{k-1|k-1}^\rho - \left(\sqrt{(L + \lambda) P_{k-1|k-1}^\rho}\right)_i, \quad i = L + 1, \dots, 2L \quad (46.33)$$

where $\left(\sqrt{(L + \lambda) P_{k-1|k-1}^\rho}\right)_i$ is the i th column of the matrix square root of $\sqrt{(L + \lambda) P_{k-1|k-1}^\rho}$. Using the definition, square root A of matrix B satisfies $B \equiv AA^T$. χ is the matrix of $2L + 1$ sigma vectors. The complete state estimation of the UKF is given in Table 46.1.

Table 46.1 The UKF state estimation algorithm

– Initialize the state vector and covariance with

$$\hat{x}_0 = E\{x_0\}$$

$$\hat{x}^\rho = E\{x^\rho\} = [\hat{x}_0^T \ 0 \ 0]^T$$

$$P_0 = E\{(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\}$$

$$P_0^\rho = E\{(x_0^\rho - \hat{x}_0^\rho)(x_0^\rho - \hat{x}_0^\rho)^T\} = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & R^v & 0 \\ 0 & 0 & R^n \end{bmatrix}$$

– For $k \in \{1, \dots, N\}$, calculate the *sigma points* from the a priori state vector mean and covariance

$$\chi_{k-1}^\rho = [\hat{x}_{k-1}^\rho \quad \hat{x}_{k-1}^\rho + \eta\sqrt{P_{k-1}^\rho} \quad \hat{x}_{k-1}^\rho - \eta\sqrt{P_{k-1}^\rho}]$$

– For $k \in \{1, \dots, N\}$, calculate the *time update* of the state vector and covariance

$$\chi_{k|k-1}^x = F(\chi_{k-1}^x, u_{k-1}, \chi_{k-1}^n)$$

$$\mathcal{Y}_{k|k-1} = H(\chi_{i,k|k-1}^x, \chi_{k-1}^n)$$

$$\hat{x}_k^- = \sum_{i=0}^{2L} \sigma_i^{(m)} \chi_{i,k|k-1}^x$$

$$\hat{y}_k^- = \sum_{i=0}^{2L} \sigma_i^{(m)} \mathcal{Y}_{i,k|k-1}$$

$$P_k^- = \sum_{i=0}^{2L} \sigma_i^{(c)} (\chi_{i,k|k-1}^x - \hat{x}_k^-) (\chi_{i,k|k-1}^x - \hat{x}_k^-)^T$$

– For $k \in \{1, \dots, N\}$, calculate the *measurement update* of the state vector and covariance

$$P_{d_k \tilde{d}_k} = \sum_{i=0}^{2L} \sigma_i^{(c)} (\mathcal{D}_{i,k|k-1} - \hat{d}_k) (\mathcal{D}_{i,k|k-1} - \hat{d}_k)^T + R_k^e$$

$$P_{x_k d_k} = \sum_{i=0}^{2L} \sigma_i^{(c)} (\chi_{i,k|k-1}^x - \hat{x}_k^-) (\mathcal{D}_{i,k|k-1} - \hat{d}_k)^T$$

$$\mathcal{K}_k = P_{x_k d_k} P_{d_k \tilde{d}_k}^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + \mathcal{K}_k (d_k - \hat{d}_k)$$

$$P_k = P_k^- - \mathcal{K}_k P_{d_k \tilde{d}_k} \mathcal{K}_k^T$$

where $\eta = \sqrt{L + \lambda}$, $x^\rho = [x \ v \ n]^T$, $\chi^\rho = [(\chi^x)^T \ (\chi^v)^T \ (\chi^n)^T]^T$, λ is the composite scaling parameter, L is the dimension of augmented states, N is the total number of sample data, R^v is process noise covariance matrix, R^n is measurement noise covariance matrix, and σ_i is the weights as calculated in Eq. (46.27)

46.3.3 Parameter Estimation

The parameter estimation problem involves determining a nonlinear mapping $y_k = G(x_k, \gamma)$ where x_k is the input, y_k is the output, and the nonlinear map G is parameterized by the parameter vector γ . Typically, a training set is provided with sample pairs consisting of known inputs and desired outputs, $\{x_k, d_k\}$, where $k = 1, \dots, N$ and N is the total number of sample data. The error is defined as $e_k = d_k - G(x_k, \gamma)$, and the goal of estimation involves solving for the parameters γ in order to minimize some given function of the error (der Merwe and Wan 2001). Parameters can be efficiently estimated online by writing a new state-space representation as follows:

$$\gamma_k = \gamma_{k-1} + r_{k-1}^\gamma \quad (46.34)$$

$$d_k = G(x_k, \gamma_k) + e_k \quad (46.35)$$

Table 46.2 The UKF parameter estimation algorithm

| | |
|---|--|
| – Initialize the parameter vector and covariance with | |
| $\hat{\gamma}_0 = E\{\gamma_0\}$ | |
| $P_{\gamma_0} = E\{(\gamma_0 - \hat{\gamma}_0)(\gamma_0 - \hat{\gamma}_0)^T\}$ | |
| – For $k \in \{1, \dots, N\}$, calculate the <i>sigma points</i> from the a priori parameter vector mean and covariance | |
| $\hat{\gamma}_k^- = \hat{\gamma}_{k-1}^-$ | |
| $\Upsilon_{k k-1} = \begin{bmatrix} \hat{\gamma}_k^- & \hat{\gamma}_k^- + \eta\sqrt{P_{\gamma_k}^-} & \hat{\gamma}_k^- - \eta\sqrt{P_{\gamma_k}^-} \end{bmatrix}$ | |
| – For $k \in \{1, \dots, N\}$, calculate the <i>time update</i> of the parameter vector and covariance | |
| $\hat{d}_k = \sum_{i=0}^{2L} \sigma_i^{(m)} \mathcal{D}_{i,k k-1}$ | |
| $P_{\gamma_k}^- = P_{\gamma_{k-1}} + R_{k-1}^r$ | |
| Option 1: $\mathcal{D}_{k k-1} = G(x_k, \Upsilon_{k k-1})$ | |
| Option 2: $\hat{d}_k = G(x_k, \hat{\gamma}_k^-)$ | |
| – For $k \in \{1, \dots, N\}$, calculate the <i>measurement update</i> of the parameter vector and covariance | |
| $P_{\hat{d}_k \hat{d}_k} = \sum_{i=0}^{2L} \sigma_i^{(c)} (\mathcal{D}_{i,k k-1} - \hat{d}_k) (\mathcal{D}_{i,k k-1} - \hat{d}_k)^T + R_k^e$ | |
| $P_{\gamma_k d_k} = \sum_{i=0}^{2L} \sigma_i^{(c)} (\Upsilon_{i,k k-1} - \hat{\gamma}_k^-) (\mathcal{D}_{i,k k-1} - \hat{d}_k)^T$ | |
| $\mathcal{K}_k = P_{\gamma_k d_k} P_{\hat{d}_k \hat{d}_k}^{-1}$ | |
| $\hat{\gamma}_k = \hat{\gamma}_k^- + \mathcal{K}_k (d_k - \hat{d}_k)$ | |
| $P_{\gamma_k} = P_{\gamma_k}^- - \mathcal{K}_k P_{\hat{d}_k \hat{d}_k} \mathcal{K}_k^T$ | |

where $\eta = \sqrt{L + \lambda}$, λ is the composite scaling parameter, L is the dimension of parameters to be estimated, N is the total number of sample data, R^r is process noise covariance matrix, and R^e is measurement noise covariance matrix

where γ_k corresponds to a stationary process, driven by process noise r_{k-1}^γ , and the output d_k corresponds to a nonlinear observation on γ_k . The complete parameter estimation of the UKF is given in Table 46.2.

46.3.4 The DUKF Estimation

The dual estimation problem, referred to as dual unscented Kalman filter (DUKF), consists of simultaneously estimating the states x and the unknown parameters γ in a coupled sequential way from the measured noisy data sequence y_k . The DUKF estimation solves this problem through performing the state and parameter estimation simultaneously, however, in a slightly different way as in the two-stage Kalman filter (TSKF) (Wu et al. 2000; Zhang and Jiang 2002). In reviewing the currently available simultaneous estimation of state and parameter techniques that can be applied to any Kalman filter, three methods are classified in this book chapter. These techniques include the following: joint, separated, and dual

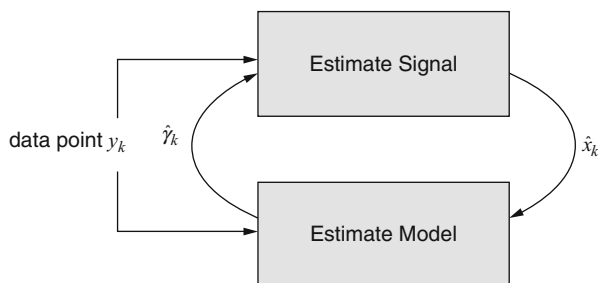


Fig. 46.4 Sequential approach of DUKF designed to pass over the data one point at a time for simultaneous state and parameter estimation (Wan and van der Merwe 2001)

estimation. They all use an analogous filter to estimate the parameters concurrently with the states. However, the joint method (e.g., the EKF) is the simplest of the three to conceptualize: the parameter vector of interest is simply appended onto the original state vector. The time update for the parameter vector portion of the augmented state vector allows no change beyond the effects of process noise, but the entire augmented covariance matrix is propagated as one (Cui and Chen 2009). Different from the joint method, the separated method (e.g., the TSKF) separates estimation of states and parameters with a reduced order of the two coupled sequential estimation processes for the states and parameters separately, therefore, reducing significantly the computational requirement comparing with the joint method. Certain speeding-up techniques, such as forgetting factors commonly used in system identification applications, can also be used for achieving fast convergence of parameter estimation, as the adaptive two-stage Kalman filter (ATSKF) proposed in Wu et al. (2000) and Zhang and Jiang (2002) for linear systems and the adaptive two-stage extended Kalman filter (ATSEKF) proposed in Zhang and Wu (1999) for nonlinear systems. The dual estimation method intertwines a pair of distinct sequential filters, one estimating the original/true states and the other estimating the parameters (Wan and van der Merwe 2001; VanDyke et al. 2004). The dual estimation technique is investigated in this work. The primary benefit of dual estimation is the ability to temporarily decouple the parameter filter from the state filter as needed. Decoupling can prevent erratic behavior due to poor measurements or initial estimate of the parameter estimation from causing the state filter to diverge (VanDyke et al. 2004).

At every time instant with new measurement, the UKF state filter estimates the states using the current parameter estimates $\hat{\gamma}_k$ obtained from Table 46.2, while the UKF parameter filter estimates the parameters based on the current state estimates \hat{x}_k obtained from Table 46.1. Such a sequential estimation scheme is shown in Fig. 46.4.

46.4 Fault Detection and Diagnosis Based on Dual Unscented Kalman Filter

46.4.1 Brief Introduction to FDD

The problem of fault detection and diagnosis (FDD) in dynamic systems has drawn a lot of attention since the 1960s. Many different techniques have been developed and successfully applied to various practical systems, as can be seen in several published comprehensive survey papers and books listed in Zhang and Jiang (2008) and more recent applications to UAVs in Amoozgar et al. (2013) and Zhang et al. (2013). Among the existing techniques, model-based approaches play an important role. Most model-based FDD methods rely on the concept of analytical redundancy. The essence of this concept is the comparison of the actual outputs of the monitored system with the outputs obtained from a mathematical model of the monitored system. The discrepancies, called as *residuals*, are used as an indication of faults. The diagnostic procedure usually consists of two steps: (1) residuals generation/generator and (2) residuals evaluation/evaluator (decision-making), as shown in Fig. 46.5. Typically, the FDD scheme has three tasks: (1) *fault detection* indicates that something has gone wrong in the monitored system, i.e., fault detection determines the occurrence of a fault and the time of the fault occurrence; (2) *fault isolation* determines the location and type of the fault (which component is faulty); and (3) *fault identification* determines the magnitude (size) of the fault. Fault isolation and identification are usually referred to as *fault diagnosis* in the literature (Isermann and Balle 1997; Zhang and Jiang 2008).

In engineering systems, including UAVs, faults can occur in actuators, sensors, and other system components. Faults can appear in the form of slowly developing (incipient) changes or abrupt changes. Further, depending on the physical properties of the fault, a fault can be added on the system in an additive or multiplicative way. Therefore, an effective FDD scheme should be able to fulfil the following basic tasks: (1) detect, isolate, and identify faults in actuators, sensors, and components; (2) detect, isolate, and identify abrupt faults as well as incipient faults and (3) detect, isolate, and identify additive and multiplicative faults.

46.4.2 Residual Generation

Based on the fundamental principle used in each technique, the existing residual generation algorithms can be classified into four categories: (1) state estimation based, (2) parameter estimation based, (3) parity space based, (4) as well as the combination of these three, as shown in Fig. 46.6. Different FDD schemes, based on different model representations of the systems, fulfil different tasks and possess different performance in FDD. For example, some of schemes can only fulfil the first two tasks: fault detection and isolation (FDI). Therefore, FDI, instead of FDD, is often used in the literature.

Fig. 46.5 Two-stage procedure of FDD schemes: residual generation and residual evaluation

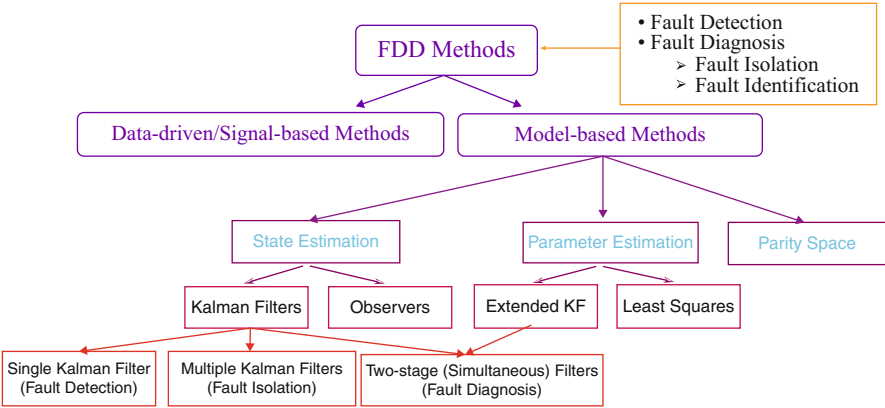
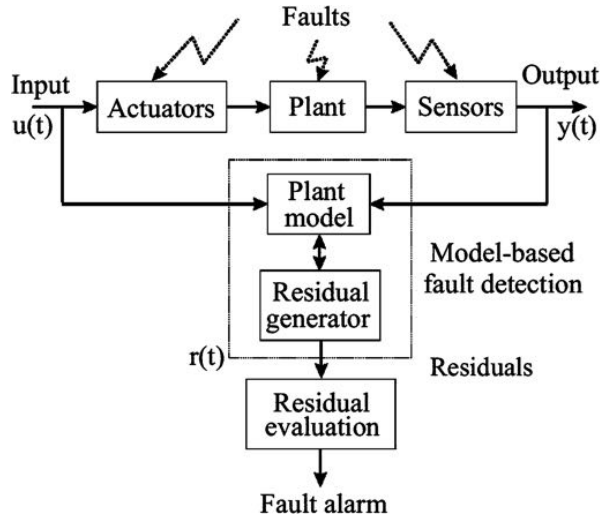


Fig. 46.6 Classification of FDD methods for residual generation

In state estimation-based approaches, the model of the system and the input-output data are utilized to synthesize an estimator (an observer in deterministic case and a Kalman filter in stochastic case). If the system operates normally, the residual of the estimator should be very small (or a white noise process in Kalman filter case). Otherwise, it will be large (or nonwhite). Therefore, such a residual signal can be used for the purpose of fault detection.

In parameter estimation-based approaches, the parameters of the system are estimated online in real time based on input-output data. The estimated parameters are then compared with the nominal ones for the purpose of FDD. In the absence of a system fault, the difference between these two parameter sets should be very small.

However, if a fault has occurred in the system, the difference will be relatively large. Therefore, FDD can be carried out simply by checking the parameter discrepancies.

Parity equations are a relatively simple and straightforward approach to build residuals that indicate faults. More sophisticated schemes with the ability to detect different faults can be gained from multi-input and multi-output processes. Parity equations require the knowledge of a fixed parameterized model that serves as a reference for the measured behavior. They are closely related to observer methods but their design is easier. A direct design from state-space equations or directly from the Laplace-transformed differential equations of the process is possible. Parity equations are especially suitable for additive faults. However, they are sensitive to noise. Parity equations do not need permanent excitation and require less computational effort than parameter estimation, but do not give the same deep insight into the process as parameter estimation. To combine the advantages of parameter estimation and parity equations, it is proposed to supervise the process online with parity equations and to perform parameter estimation after a fault is detected.

A common feature among these methods is to generate a residual signal which should be sensitive to the occurrence of the fault and robust to disturbances and model uncertainties. Because different residual generating mechanisms have been used in different methods, their performance will be different even under the same fault scenario. Subsequently, a very important question arises: which method is most appropriate for a given application? With consideration of using FDD for providing necessary information on the status of post-failure system for control (even guidance and navigation) reconfiguration, the states and system parameter information are necessary for maintaining system/UAV stability and acceptable performance after fault occurrence. Therefore, the three functions in FDD for detection, isolation/localization, and identification are all needed (Zhang and Jiang 2006, 2008). These considerations and requirements are in fact the motivation of this book chapter and the works carried out at (Ma and Zhang 2010a, b).

In particular, the proposed FDD scheme is on the basis of the abovementioned DUKF for residual generation using the simultaneous state and fault parameter estimation.

46.4.3 Residual Evaluation/Decision-Making

Finally, to perform fault detection and isolation, many rules can be used to evaluate the estimated parameters γ^i such as the two-stage statistical hypothesis tests developed in Zhang and Jiang (2002).

In the first stage, the statistical quantities of the system under the normal operation, such as mean values and variances, are determined, and then the same quantities during the continuous system operation are calculated. By defining an appropriate statistical detection variable to accentuate the deviation in the statistical quantities from their normal values, the detection and isolation of the reduction/loss of control effectiveness can be achieved recursively.

Stage I. Under the normal condition, one has

$$\hat{\mathbf{y}}_k \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_{\hat{\mathbf{y}}_0}, \boldsymbol{\sigma}_{\hat{\mathbf{y}}_0}^2) \quad (46.36)$$

where $\hat{\mathbf{y}}_k \in \mathfrak{R}^l$ is the estimate of the reduction of control effectiveness and it can also be used as the residual vector for fault detection. $\bar{\boldsymbol{\mu}}_{\hat{\mathbf{y}}_0} = [\bar{\mu}_{\hat{y}_0^1} \ \bar{\mu}_{\hat{y}_0^2} \ \dots \ \bar{\mu}_{\hat{y}_0^l}]^T$ and $\boldsymbol{\sigma}_{\hat{\mathbf{y}}_0}^2 = \text{diag}[\sigma_{\hat{y}_0^1}^2 \ \sigma_{\hat{y}_0^2}^2 \ \dots \ \sigma_{\hat{y}_0^l}^2]$ are the mean and variance, respectively. These values are obtained, for $i = 1, \dots, l$, by

$$\bar{\mu}_{\hat{y}_0^i} = \frac{1}{N_1} \sum_{j=1}^{N_1} \hat{y}_j^i \quad (46.37)$$

$$\sigma_{\hat{y}_0^i}^2 = \frac{1}{N_1 - 1} \sum_{j=1}^{N_1} [\hat{y}_j^i - \bar{\mu}_{\hat{y}_0^i}]^2 \quad (46.38)$$

The sample size N_1 is chosen to ensure the sufficiency in statistics.

Stage II. Define the following moving window-based statistical quantities:

$$\begin{aligned} \bar{\mu}_{\hat{y}_k^i} &= \frac{1}{N_2} \sum_{j=k-N_2+1}^k \hat{y}_j^i \\ &= \bar{\mu}_{\hat{y}_{k-1}^i} - \frac{1}{N_2} [\hat{y}_{k-N_2}^i - \hat{y}_k^i] \end{aligned} \quad (46.39)$$

$$\begin{aligned} \sigma_{\hat{y}_k^i I}^2 &= \frac{1}{N_2 - 1} \sum_{j=k-N_2+1}^k [\hat{y}_j^i - \bar{\mu}_{\hat{y}_0^i}]^2 \\ &= \sigma_{\hat{y}_{k-1}^i I}^2 - \frac{1}{N_2 - 1} \left[(\hat{y}_{k-N_2}^i)^2 - (\hat{y}_k^i)^2 - 2\bar{\mu}_{\hat{y}_0^i} \varepsilon_k^i \right] \end{aligned} \quad (46.40)$$

$$\begin{aligned} \sigma_{\hat{y}_k^i II}^2 &= \frac{1}{N_2 - 1} \sum_{j=k-N_2+1}^k [\hat{y}_j^i - \bar{\mu}_{\hat{y}_k^i}]^2 \\ &= \sigma_{\hat{y}_{k-1}^i II}^2 + \frac{1}{N_2 - 1} \left[2\varepsilon_k^i \bar{\mu}_{\hat{y}_{k-1}^i} - (\hat{y}_{k-N_2}^i)^2 + (\hat{y}_k^i)^2 - \frac{1}{N_2 - 1} (\varepsilon_k^i)^2 \right] \end{aligned} \quad (46.41)$$

where $\varepsilon_k^i = \hat{y}_{k-N_2}^i - \hat{y}_k^i$ and N_2 is the length of the moving window.

Consequently, a fault in the i th actuator is declared at time k if the following detection index

$$d_k^i = \frac{\sigma_{\hat{y}_k^i I}^2}{\sigma_{\hat{y}_0^i}^2} - \ln \frac{\sigma_{\hat{y}_k^i II}^2}{\sigma_{\hat{y}_0^i}^2} - 1, \quad i = 1, \dots, l \quad (46.42)$$

exceeds a preset threshold η^i .

$$d_k^i \underset{H_0}{\overset{H_i}{\geq}} \eta^i \quad (46.43)$$

where $H_0 = \{\text{No significant reduction in the effectiveness of the } i\text{th actuator}\}$ and $H_i = \{\text{There is significant reduction in the effectiveness of the } i\text{th actuator}\}$. Here, “significant reduction” is a design parameter and should be reflected in the selection of the threshold η^i . The selection of the window length, N_2 , and the threshold, η^i , represents some trade-off between the probability of false alarm and the probability of missed detection.

46.5 Simulation Results

Simulation results and analysis of the nonlinear GTM and the LPV model of GTM using the above DUKF-based FDD scheme are presented in this section. For simulations, it is assumed that the collective elevator actuator encounters a partial loss fault while others remain healthy although the DUKF-based FDD does not have any limitation for handling multiple, simultaneous, and consecutive faults. Consideration of only elevator fault in this work is mainly due to the fact that the longitudinal motion of the GTM UAV has been considered in the simulation validation of the FDD scheme where the only available redundant actuator in UAV longitudinal motion is the engine throttle. The throttle of the GTM UAV is assumed to be kept constant at its trim setting throughout the maneuver. The response to the UAV is captured through equivalent airspeed, angle of attack, pitch angle, and pitch angular rate.

For nonlinear model of GTM, two scenarios are simulated: (1) a 50 % loss of control effectiveness fault in elevator at 5 s and (2) a 20 % loss of control effectiveness fault in elevator at 6 s. To reduce the execution time, the measurement interval of $T = 0.02$ s is used.

The UKF parameters are listed as follows:

$$\rho = 1, \sigma = 2, \kappa = 3 - L, L = 15.$$

where L is the dimension of the augmented states.

46.5.1 FDD Results Based on Nonlinear Model

Assume that at time instant 5 s, elevator encounters a 50 % (50 % of the original normal value) partial loss fault in control effectiveness. The estimation results for states and fault parameters are shown in Figs. 46.7 and 46.8, respectively.

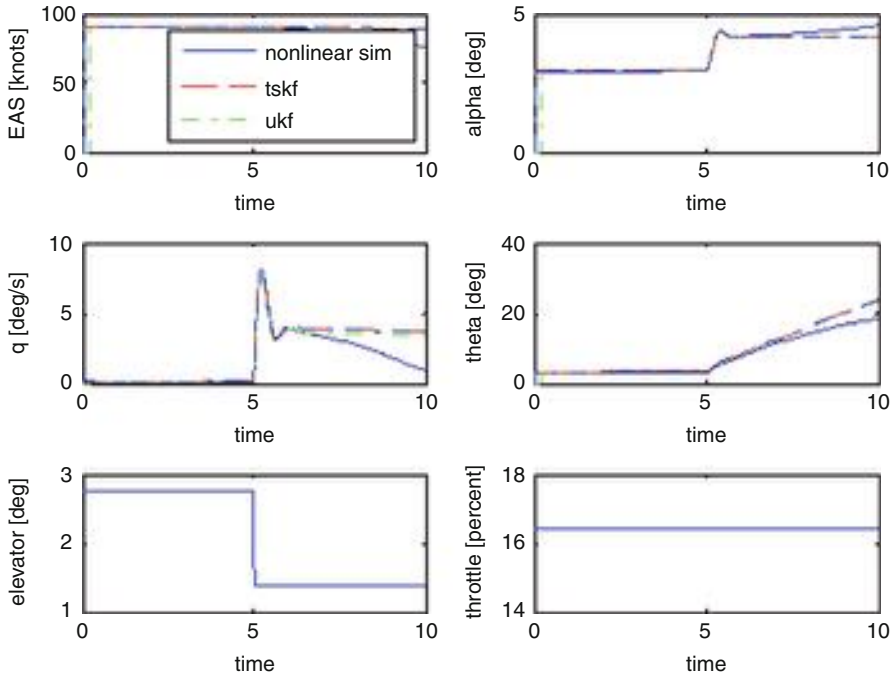


Fig. 46.7 Results of estimated states under 50 % loss of control effectiveness

As shown in the figures, the two-stage Kalman filter (TSKF) and adaptive two-stage extended Kalman filter (ATSEKF) developed in Zhang and Wu (1999), Wu et al. (2000), and Zhang and Jiang (2002) have also been implemented in GTM for comparison.

From Figs. 46.7 and 46.8, one can see that DUKF can correctly estimate all the states and fault magnitudes within the given time limits in the nonlinear model. It is easy to implement DUKF in the nonlinear model since it does not need to linearize the model. Compared with the TSKF, almost the same estimate for fault parameter has been achieved. However, the DUKF is computationally more expensive.

Figures 46.9 and 46.10 show the simulation results for a 20 % loss of control effectiveness fault (i.e., 80 % of the normal value) in elevator occurred at 6 s. It can be seen that good and almost same performance have also been achieved by both DUKF and TSKF.

In the above simulation cases, the actuator faults have been detected at 5.3 s and 6.3 s, respectively, with 0.3 s time delay. It can be viewed that similar parameter estimation accuracy has been achieved by both DUKF and TSKF.

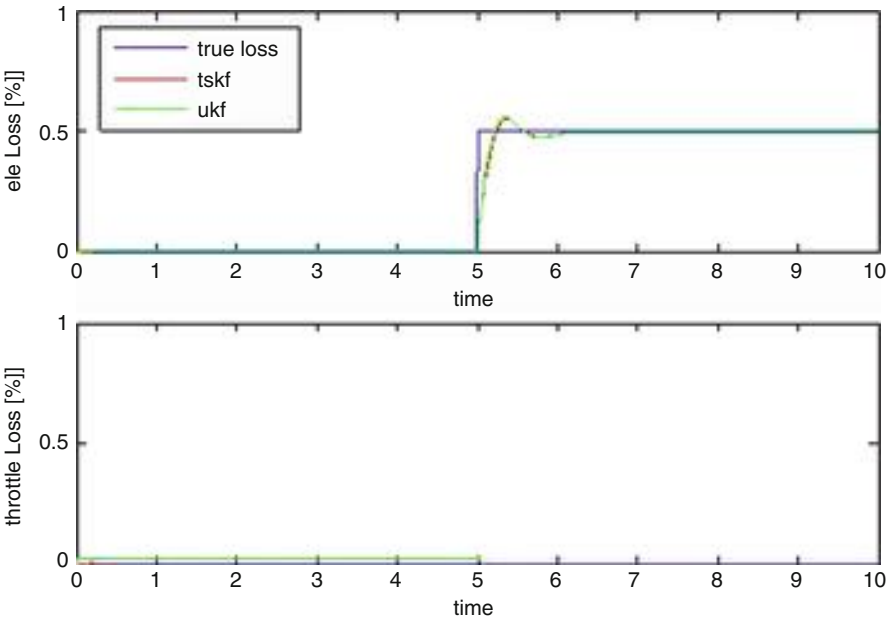


Fig. 46.8 Results of estimated parameters under 50 % loss of elevator effectiveness

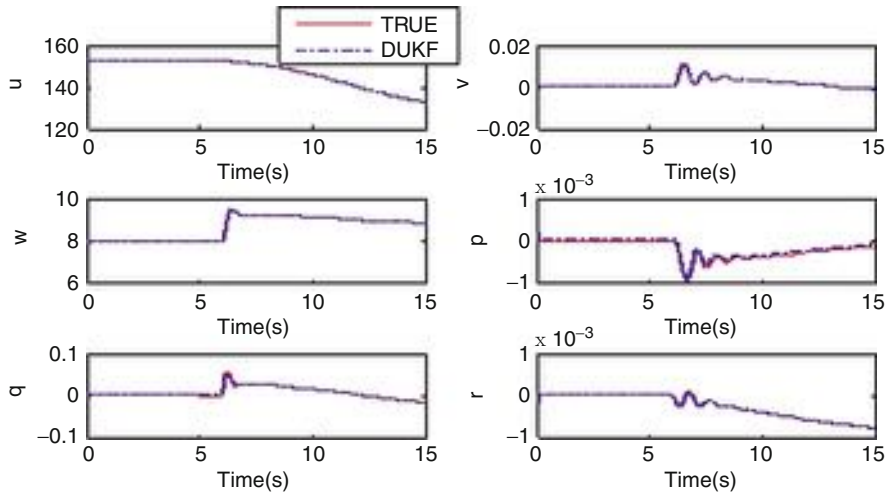


Fig. 46.9 Results of estimated states under 20 % loss of elevator effectiveness

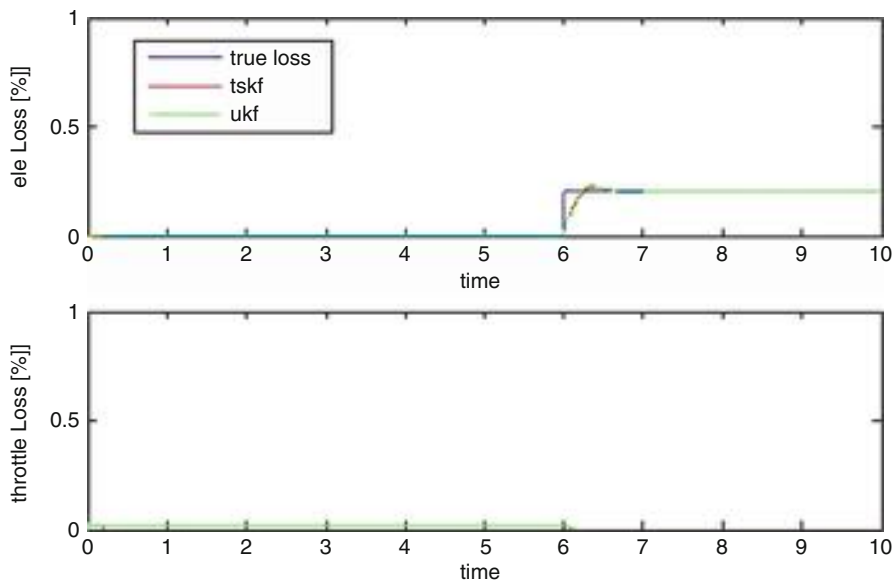


Fig. 46.10 Results of estimated parameters under 20 % loss of elevator effectiveness

46.5.2 FDD Results Based on LPV Model

In this case, DUKF and ATSEKF use the measured outputs and outputs of the LPV model, a 20 % loss of control effectiveness fault in elevator at 6 s is considered. The measurement interval is $T = 0.01$ s.

In Fig. 46.11, the estimation outputs of DUKF and ATSEKF are close to the LPV output at the beginning; however, with the time growing, the DUKF and ATSEKF cannot follow the output of LPV model. For fault parameter estimation, as shown in Fig. 46.12, the estimation results demonstrate that DUKF can also detect and identify the fault precisely based the LPV model when a 20 % partial loss fault occurs at 6 s in elevator, with similar performance as that of ATSEKF.

Figures 46.11–46.12 show that the states estimation of the DUKF and ATSEKF in the GTM LPV environment matched well with the measured outputs and also it filtered the measurement noises and two Kalman algorithms detect and identify the fault precisely when the elevator encounters a partial loss.

46.5.3 Comparison of DUKF with TSKE/ATSEKF

The simulation results based on the GTM six degree-of-freedom nonlinear simulation environment show that DUKF provides similar estimation performance

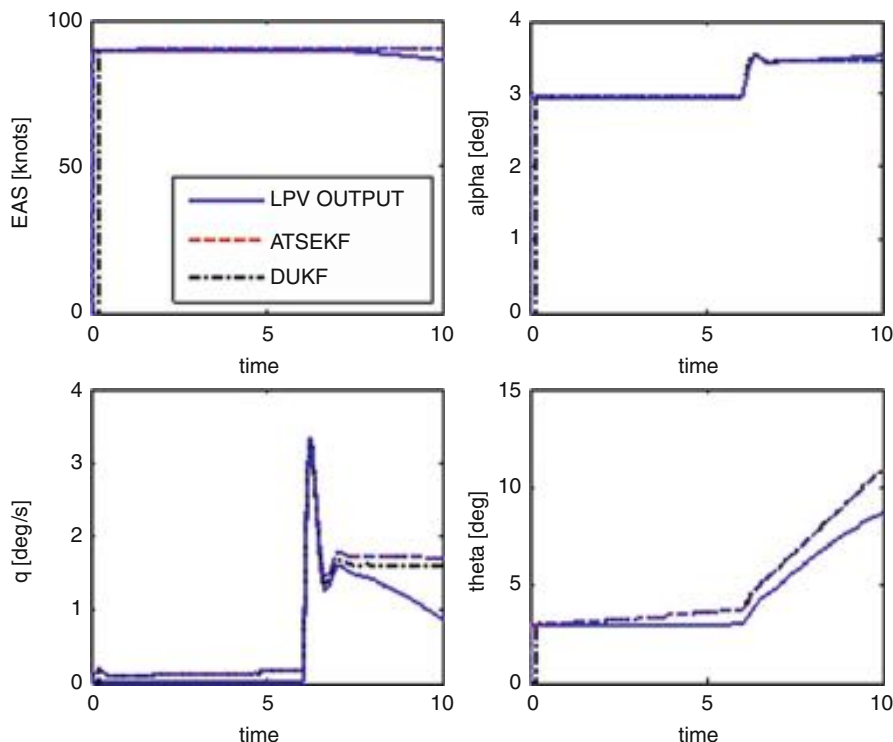


Fig. 46.11 Results of estimated states under 20 % loss of elevator effectiveness

compared with TSKF. DUKF is computationally more expensive, especially for nonlinear models whose coefficients are implemented by lookup tables in the MATLAB/Simulink. The reason is that DUKF algorithm requires the $2L + 1$ sigma points, where L is the dimension of the augmented state vector. These $2L + 1$ sigma points join the measurement update and time update, so it makes the computation $2L + 1$ times larger than that of TSKF/ATSEKF. Since ATSEKF linearizes the system at each sample point, it could lead to a poor performance by introducing unknown errors due to linearization. Such a linearization operation at each time step makes also the implementation more demanding comparing with DUKF in which nonlinear model of the system can be directly used for estimation with no need to linearize the system.

Experimental validation of FDD schemes based on a parameter estimation alone UKF, two-stage Kalman filter, and an adaptive two-stage Kalman filter with applications to a quadrotor helicopter UAV test bed available at the author's Networked Autonomous Vehicles Lab (NAVL) in Concordia University are presented in another chapter in this handbook. Interested readers are referred to Chamseddine et al. (2013) for details.

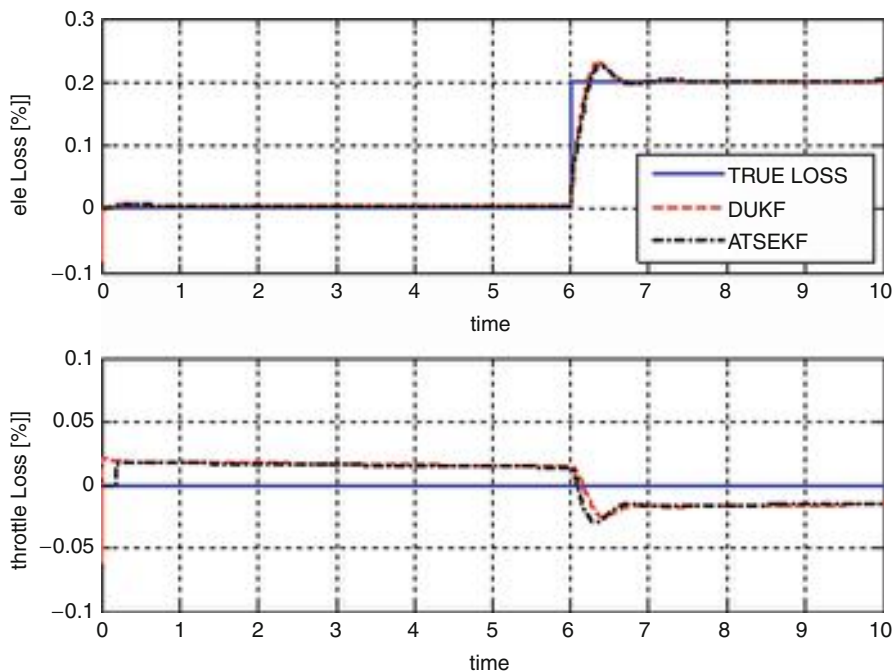


Fig. 46.12 Results of estimated parameters under 20 % loss of elevator effectiveness

46.6 Conclusion

An online FDD scheme has been presented in this chapter. The proposed FDD scheme is based on the DUKF for simultaneous state and fault parameter estimation. Simulation results indicate that DUKF can correctly estimate all the states and fault parameters within the given time limit in the NASA nonlinear GTM UAV model and the LPV model of the nonlinear UAV model. Convergence of parameter estimation is also very fast. However, the DUKF is also computationally more expensive, especially for nonlinear model whose coefficients are implemented by lookup tables in MATLAB/Simulink.

For the nonlinear model and LPV model of the NASA GTM UAV, simulation results show that DUKF and TSKF/ATSEKF provide similar parameter estimation accuracy and FDD performance. In this work, only actuator faults have been considered. In practice, faults can also occur in sensors and system components such as wing damages. Furthermore, only partial loss type faults have been considered. Future works include consideration of actuator stuck failures and wing damages, improvement of robustness and performance of DUKF-based FDD algorithm, and integration with fault-tolerant control to form a complete active fault-tolerant control system for the GTM UAV.

In conclusion, DUKF has the advantage that it separates state estimation and parameter estimation. Such a feature benefits fault parameter estimation needed in an FDD task. Furthermore DUKF can use the nonlinear model of system directly with no need to linearize the system.

Acknowledgments This work was partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) through a strategic project and a discovery project and also by the NASA Aviation Safety Program (AvSP) through a project on “Fault Diagnosis and Prognosis and Reliable Flight Envelope Assessment.”

The author would like to thank Ms. L. Ma for her simulation work carried out in this book chapter through her master thesis at Concordia University, Dr. D. Cox of NASA for providing the open-loop GTM in MATLAB/Simulink through the AvSP project, Dr. G. Balas and Mr. R. Pandita of University of Minnesota for providing LPV model and modifying the open-loop GTM into the closed-loop GTM required by this work, Dr. N. E. Wu and Mr. M. C. Ruschman of Binghamton University for their help on LPV model implementation, and Dr. J. Y. Shin of Gulfstream for his help during regular discussions of the project through Telcon.

References

- M.H. Amoozgar, A. Chamseddine, Y.M. Zhang, Experimental test of a two-stage Kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter. *J. Intell. Robot. Syst.* **70**(1), 107–117 (2013)
- C.M. Belcastro, Aircraft loss of control: analysis and requirements for future safety-critical systems and their validation, in *8th Asian Control Conference*, Kaohsiung, May 2011, pp. 399–406
- A. Chamseddine, M.H. Amoozgar, Y.M. Zhang, Experimental validation of fault detection and diagnosis for unmanned aerial vehicles, in *Handbook of Unmanned Aerial Vehicles*, ed. by K. Valavanis, G.J. Vachtsevanos (Springer, Dordrecht, 2013)
- C.K. Cui, G. Chen, *Kalman Filtering with Real-Time Applications* (Springer, Berlin/Heidelberg, 2009)
- R.V. der Merwe, E. Wan, The square-root unscented Kalman filter for state and parameter estimation, in *Proceedings of the International Conference on Acoustics, Speech and Signal Process*, Maui, 7–11 May 2001, pp. 3461–3464
- I.M. Gregory, E. Xargay, C. Cao, N. Hovakimyan, Flight test of an L_1 adaptive controller on the NASA AirSTAR flight test vehicle, in *AIAA Guidance, Navigation and Control Conference*, Toronto, Aug 2010
- J. Guo, G. Tao, A feedback-based fault detection scheme for aircraft systems with damage, in *8th Asian Control Conference*, Kaohsiung, May 2011, pp. 1431–1436
- J. Guo, G. Tao, Discrete-time adaptive control of a nonlinear aircraft flight dynamic system (NASA GTM) with damage, in *51st IEEE Conference on Decision and Control*, Maui, Dec 2012
- N. Hovakimyan, C. Cao, E. Kharisov, E. Xargay, I.M. Gregory, L_1 adaptive control for safety-critical systems. *IEEE Control Syst. Mag.* **31**(5), 54–104 (2011)
- R. Isermann, P. Balle, Trends in the application of model based fault detection and diagnosis of technical processes. *Control Eng. Pract.* **5**(5), 709–719 (1997)
- T.L. Jordan, J.V. Foster, R.M. Bailey, C.M. Belcastro, AirSTAR: a UAV platform for flight dynamics and control system testing, in *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, San Francisco (AIAA, 2006), pp. 2006–3307
- S.J. Julier, J.K. Uhlmann, A new extension of the Kalman filter to nonlinear systems, in *Proceedings of AeroSense: the 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, Orlando, 20–25 Apr 1997, pp. 182–193

- H.G. Kwatny, J.-E. Dongmo, B.-C. Chang, G. Bajpa, M. Yasar, C. Belcastro, Aircraft accident prevention: loss-of-control analysis, in *AIAA Guidance, Navigation and Control Conference*, Chicago, 2009
- L. Ma, Development of fault detection and diagnosis techniques with applications to fixed-wing and rotary-wing UAVs. Master's thesis, Concordia University, Montreal, 2011
- L. Ma, Y.M. Zhang, DUKF-based fault detection and diagnosis for GTM UAV using nonlinear and LPV models, in *Proceedings of the 6th ASME/IEEE International Conference on Mechatronic & Embedded Systems & Applications*, Qingdao, 15–17 July 2010a
- L. Ma, Y.M. Zhang, Fault detection and diagnosis for GTM UAV with dual unscented Kalman filter, in *AIAA Guidance, Navigation, and Control Conference*, Toronto, 2–5 Aug 2010b
- A. Marcos, G.J. Balas, Development of linear-parameter-varying models for aircraft. *J. Guid. Control Dyn.* **27**(2), 218–228 (2004)
- J. Mohammadpour, C.W. Scherer, *Control of Linear Parameter Varying Systems with Applications* (Springer, New York, 2012)
- R. Pandita, A. Chakraborty, P. Seiler, G. Balas, Reachability and region of attraction analysis applied to GTM dynamic flight envelope assessment, in *AIAA*, Chicago, 2009. *AIAA* 2009–6258
- M. Ruschmann, J. Huang, N.E. Wu, Probing the NASA generic transport aircraft in real-time for health monitoring, in *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, Dec 2009, pp. 4920–4929
- I. Sadeghzadeh, Y.M. Zhang, Review on fault tolerant control for unmanned aerial vehicles (UAVs), in *AIAA InfoTech@Aerospace 2011: Unleashing Unmanned Systems*, St. Louis, 29–31 Mar 2011
- J.-Y. Shin, Worst-case analysis and linear parameter-varying gain-scheduled control of aerospace systems. Ph.D. dissertation, University of Minnesota, 2000
- M.C. VanDyke, J.L. Schwartz, C.D. Hall, Unscented Kalman filtering for spacecraft attitude state and parameter estimation, in *Proceedings of the 14th AAS/AIAA Space Flight Mechanics Meeting*, Maui, 8–12 Feb 2004
- E.A. Wan, R. van der Merwe, The unscented Kalman filter for nonlinear estimation, in *Proceedings of Symposium on Adaptive Systems, Signal Processing, Communication, and Control*, Lake Louise, Oct 2000
- E.A. Wan, R. van der Merwe, The unscented Kalman filter, in *Kalman Filtering and Neural Networks*, ed. by S.S. Haykin (Wiley, New York, 2001), pp. 221–280
- N.E. Wu, Y.M. Zhang, K.M. Zhou, Detection, estimation, and accommodation of loss of control effectiveness. *Int. J. Adapt. Control Signal Process. Spec. Issue Fault Detect. Isol.* **14**(7), 775–795 (2000)
- G.X. Xu, H.H.T. Liu, Trajectory tracking design for a transport aircraft using model reference adaptive control, in *CASI 58th Aeronautics Conference*, Montreal, Apr 2011
- Y.M. Zhang, J. Jiang, An active fault-tolerant control system against partial actuator failures. *IEE Proc. Control Theory Appl.* **149**(1), 95–104 (2002)
- Y.M. Zhang, J. Jiang, Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems, in *Preprints of the 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, Aug 2006, pp. 1513–1524
- Y.M. Zhang, J. Jiang, Bibliographical review on reconfigurable fault-tolerant control systems. *IFAC Annu. Rev. Control* **32**(2), 229–252 (2008)
- Y.M. Zhang, N.E. Wu, Fault diagnosis for a ship propulsion benchmark: Part I, in *Preprints of the 14th IFAC World Congress*, Beijing, 5–9 July 1999
- Y.M. Zhang, A. Chamseddine, C. Rabbath, B. Gordon, C.-Y. Su, S. Rakheja, C. Fulford, J. Apkarian, P. Gosselin, Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed. *J. Frankl. Inst.* **350**(9), 2396–2422 (2013)

Fault Diagnosis of Skew-Configured Inertial Sensor System for Unmanned Aerial Vehicles

47

Seungho Yoon, Seungkeun Kim, Jonghee Bae, Youdan Kim, and Eung Tai Kim

Contents

| | | |
|--------|---|------|
| 47.1 | Introduction | 1184 |
| 47.2 | Algorithms for Model-Free Hybrid FDI | 1186 |
| 47.2.1 | Parity Space Method | 1186 |
| 47.2.2 | In-Lane Monitoring Method Based on the Discrete Wavelet Transform | 1190 |
| 47.2.3 | Model-Free Hybrid Fault Detection and Isolation | 1193 |
| 47.3 | Hardware Configuration for the Flight Test | 1195 |
| 47.3.1 | Skew-Configured Inertial Measurement Unit | 1195 |
| 47.3.2 | Fault Signal Generation Board | 1198 |
| 47.3.3 | Experimental Unmanned Aerial Vehicle System | 1198 |
| 47.4 | HILS Test and Flight Experiments | 1199 |
| 47.4.1 | HILS Test Results | 1199 |
| 47.4.2 | Flight Experimental Result | 1200 |
| 47.4.3 | Analysis | 1204 |
| 47.5 | Conclusion | 1210 |
| | References | 1210 |

S. Yoon (✉) • J. Bae • Y. Kim

School of Mechanical and Aerospace Engineering, Seoul National University, Seoul,
Republic of Korea

e-mail: aeroyooun@gmail.com; jhbae23@snu.ac.kr; ydkim@snu.ac.kr

S. Kim

Department of Aerospace Engineering, Chungnam National University, Daejeon, South Korea
e-mail: skim78@cnu.ac.kr

E.T. Kim

Flight Control and Avionics Team, Korea Aerospace Research Institute, Daejeon, South Korea

Abstract

A fault detection and isolation scheme is presented to handle three successive faults in the skew-configured inertial sensors of an unmanned aerial vehicle. The skew-configured inertial sensors are composed of the primary inertial measurement unit and the redundant secondary inertial measurement unit. Since small unmanned aerial vehicles are restricted by cost and payload space, the secondary small and low-cost inertial measurement unit is installed with a skewed angle in addition to the primary inertial measurement unit. In the hybrid fault detection and isolation scheme, a parity space method and an in-lane monitoring method are combined to increase system tolerance to the occurrence of multiple successive faults during flight. The first and second faults are detected and isolated by the parity space method. The third fault is detected by the parity space method and isolated by the in-lane monitoring method based on the discrete wavelet transform. Hardware-in-the-loop tests and flight experiments with a fixed-wing unmanned aerial vehicle are performed to verify the performance of the proposed fault diagnosis scheme.

47.1 Introduction

Modern flight control systems are designed to tolerate various faults in sensors and actuators during flight; these faults must be detected and isolated as soon as possible to allow the overall system to continue its mission. A fault means an unpermitted deviation of characteristic properties or parameters of the system from the normal condition, and a failure means a permanent interruption of partial systems (Patton et al. 1989; Isermann and Balle 1997). If any successive faults in a subsystem are not resolved, a serious failure might be caused not only in the subsystem but also in the whole system. A widely used precaution for these types of malfunctions is the fault detection and isolation (FDI) technique, which is based on a redundancy management system. Here, fault detection refers to the decision of whether or not a fault occurs, and fault isolation is defined as the process of finding and excluding the faulty component. Generally, the FDI technique is categorized into hardware redundancy and analytic redundancy managements. Hardware redundancy management (Gilmore and McKern 1970; Potter and Deckert 1972; Ray and Desai 1984) involves multiple redundant sensors or actuators and makes the system heavy and complicated. On the other hand, since the analytic redundancy management uses a mathematical model of the system redundancy, the typically used model-based FDI methods (e.g., observers (Patton et al. 2000), parity space (Gertler 1998) and parameter estimation (Isermann and Balle 1997; Isermann 2006)) are prone to modeling errors. In addition, a related robustness problem should be addressed as defined in Chen and Patton (1999). Recently, an in-lane monitoring method based on signal processing has been widely used for diagnosing the sensors or actuators of various systems as embedded computing technology develops (Patton 1997; Zhang and Yan 2001).

The triple or quadruple hardware redundancies generally used for large conventional aircraft or high-performance military aircraft are not suitable for small aircraft or unmanned aerial vehicle (UAV) systems because small and inexpensive systems are limited by cost and payload space. Therefore, this study considers a limited redundancy system consisting of a primary inertial measurement unit (IMU) and an additional small and low-cost IMU. A parity space approach and an in-lane monitoring technique are used in concert to increase tolerance to multiple successive faults in the IMU during flight. If an additional sensor is mounted in the same direction as the primary sensor, the first fault on each axis can be detected, but cannot be isolated because there are only two sensors for each axis; therefore, it is impossible to decide which sensor is functioning properly. To overcome this limitation, a skew-configured IMU is considered in this study. The skewed configuration refers to the configuration of the sensors that are mounted in orientations other than the orthogonal axes, provided that the measurements in three degrees of freedom can be expressed as independent linear combinations of the orthogonal components (Kim et al. 2004, 2007). This sensor configuration has been used to facilitate proper fault diagnosis for the IMUs of space shuttles and rockets (Sudano et al. 1988; Potter and Suman 1977; Jankovitz and Perdsock 1988; DeAngelis 1990; Krogmann 1990; McClary and Walborn 1994; Titterton and Weston 1997). However, designing a skew-configured IMU with peripheral devices such as low-level individual inertial sensors, accelerometers, and gyroscopes is costly and time-consuming because it involves optimal alignment and calibration problems. To cope with this difficulty, this study proposes that a low-cost off-the-shelf IMU can be mounted to a plane relatively inclined to the plane of the primary IMU, which is aligned with aircraft body coordinates. Because this idea is easy and practical, it can be applied to various unmanned systems regardless of whether the mission takes place over ground, sea, or air.

The previous works (Kim et al. 2004, 2007) proposed a model-free fault detection and an isolation technique for a sensor system with limited redundancy using the parity equation and wavelet transform. The basic idea of that technique is also applicable to the successive triple faults of the skew-configured dual IMU system. The first and second faults can be detected and isolated by the parity space approach (Gilmore and McKern 1970), and the third fault can be detected and isolated by the in-lane monitoring method based on the discrete wavelet transform (Kim et al. 2004, 2007; Bogges and Narcowich 2001; Daubechies 1992; Mallat 1999; Li 1999; Santoso et al. 1996; Mallat 1989; Dohono and Johnstone 1994). Finally, hardware-in-the-loop simulation (HILS) experiments and flight tests were performed to verify the performance and practicability of the proposed FDI scheme.

The rest is organized as follows: Sect. 47.2 summarizes the model-free hybrid FDI method composed of the parity space method and the discrete wavelet transform. Section 47.3 describes peripheral device configuration and alignment/calibration process of the proposed skew-configured IMU. Section 47.4 presents the results and analysis of the HILS and flight test. Conclusion is given in Sect. 47.5.

47.2 Algorithms for Model-Free Hybrid FDI

The analytic redundancy management is categorized into a model-based FDI and a signal-based FDI. The model-based FDI requires a mathematical model of a system, whereas the signal-based FDI does not. Therefore, the accuracy of the mathematical model affects the FDI performance of the model-based FDI. The hybrid FDI technique proposed in this study consists of two methods: the parity space method and the in-lane monitoring method based on the discrete wavelet transform. The parity space method is a model-based FDI, and the in-lane monitoring method is a signal-based FDI. In the parity space method, the fault in inertial sensors is detected and isolated using the geometrical relationship between skew-configured sensors. If the geometrical model is perfectly known, the parity space method is independent of the aircraft motion, which is model-free. However, the parity space method cannot isolate a certain level of successive faults due to the limited independent relationship between healthy sensors. For instance, in the case of a cone configuration of six sensors, the third fault can be detected but cannot be isolated by the parity space method alone. To deal with this problem, the parity space method is combined with the in-lane monitoring method based on the discrete wavelet transform; therefore, successive faults in the inertial sensors can be isolated by the in-lane monitoring method.

Figure 47.1 describes a schematic of the proposed FDI technique. When the first fault occurs in Sensor 1, FDI is accomplished by a three-dimensional parity vector as shown in line A of Fig. 47.1. The second fault in Sensor 2 is also detected and isolated using a two-dimensional parity vector as shown in line B of Fig. 47.1. Lastly, the successive third fault that could occur in Sensors 3–6 is detected by a one-dimensional parity vector and isolated by the discrete wavelet transform as shown in line C of Fig. 47.1. The parity space method is explained in Sect. 47.2.1, and the in-lane monitoring method is presented in Sect. 47.2.2. The detailed procedure of the FDI is described in Sect. 47.2.3.

47.2.1 Parity Space Method

The parity space method is extended from a static parity relationship that is widely applied to various FDI problems, as described in several papers (Chow and Willsky 1984; Patton and Chen 1994; Gertler 1997; Ding et al. 1999; Hwang et al. 2010). To address the parity relation in a sensor fault problem, the discrete state-space aircraft model is considered:

$$x(k+1) = Ax(k) + Bu(k) + v(k) \quad (47.1)$$

$$y(k) = Cx(k) + Du(k) + w(k) + f(k) \quad (47.2)$$

where $x(k) \in R^n$ is a state vector, $u(k) \in R^q$ is an input vector, $y(k) \in R^m$ is an output vector, $v(k) \in R^n$ and $w(k) \in R^m$ are noise or disturbance vectors,

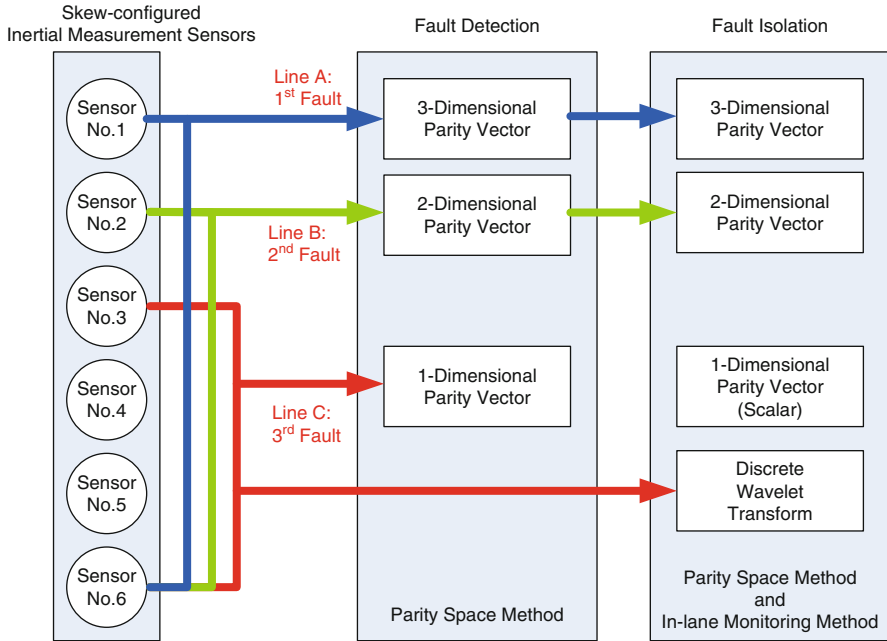


Fig. 47.1 Hybrid fault detection and isolation of three successive faults in inertial sensors

$f(k) \in R^m$ is a fault vector of the sensor, $A \in R^{n \times n}$ is a system matrix, $B \in R^{n \times q}$ is an input matrix, $C \in R^{m \times n}$ is a measurement matrix, and $D \in R^{m \times q}$ is an input-to-output matrix. The $v(k)$ and $w(k)$ are zero-mean white noise vectors with the Gaussian distribution. Each element of the fault vector $f_i(k)$ ($i = 1, 2, \dots, m$) corresponds to a fault signal of each sensor. The state-space model can be transformed to the input-output equation as

$$y(z) = G_u(z)u(z) + C(zI - A)^{-1}v(z) + w(z) + f(z) \quad (47.3)$$

where

$$G_u(z) = C(zI - A)^{-1}B + D \quad (47.4)$$

Patton and Chen (1994) showed that, in the static parity relation approach, a residual vector is generally defined as

$$r(z) = H_u(z)u(z) + H_y(z)y(z) \quad (47.5)$$

where H_u and H_y are parity matrices with respect to the input vector and the output vector, respectively. The residual vector is defined to be insensitive to the input vector as follows:

$$H_u(z) = -H_y(z)G_u(z) \quad (47.6)$$

$$r(z) = H_y(z) \left[C(zI - A)^{-1} v(z) + w(z) + f(z) \right] \quad (47.7)$$

Note that the residual vector in Eq. (47.7) is independent from the input vector $u(z)$ and the state vector $x(z)$. When a sensor fault does not occur, the residual vector is a zero-mean vector as shown in Eq. (47.8). When a sensor fault occurs, the residual vector is a nonzero vector as shown in Eq. (47.9). Consequently, the residual vector in the static parity relation is sensitive to the sensor fault and insensitive to the input and the state.

$$E[r(z)|_{f(z)=0}] = E\left[C(zI - A)^{-1} v(z) + w(z)\right] \approx 0 \quad (47.8)$$

$$E[r(z)|_{f(z) \neq 0}] = E\left[C(zI - A)^{-1} v(z) + w(z) + f(z)\right] \approx E[f(z)] \quad (47.9)$$

The parity vector $p(k)$ is introduced similar to the residual vector in Eq. (47.5) as

$$p(k) = H_u(k) u(k) + H_y(k) y(k) \quad (47.10)$$

$$H_u(k) = -H_y(k) D \quad (47.11)$$

Then, the primary residual vector $v(k) \in R^m$ is represented as (Hwang et al. 2010)

$$v(k) = y(k) - Du(k) = C(k)x(k) + w(k) + f(k) \quad (47.12)$$

Finally, the parity vector is defined as

$$p(k) = H_y(k) v(k) \quad (47.13)$$

In this study, $x(k) \in R^3$ is an angular rate vector in the aircraft body coordinate: roll rate, pitch rate, and yaw rate. Initial $y(k) \in R^6$ is measured with skew-configured inertial sensors which are composed of two IMUs. The initial measurement matrix is $C(k) \in R^{6 \times 3}$ which transforms the angular rates in the body coordinate to the angular rates in the sensor coordinate. Therefore, initial $p(k)$ is a three-dimensional vector, and initial $H_y(k)$ is a 3-by-6 matrix with the following orthonormal properties (Potter and Deckert 1972; Ray and Desai 1984):

$$H_y(k) C(k) = O_{3 \times 3} \quad (47.14)$$

$$H_y(k) H_y^T(k) = I_{3 \times 3} \quad (47.15)$$

where $O_{3 \times 3}$ is a zero matrix, $I_{3 \times 3}$ is a identity matrix, and the parity matrix $H_y(k)$ is the null space matrix of $C(k)$. In detail, $H_y(k)$ is a transpose matrix of the null space of $C^T(k)$. The null space is calculated using the singular value decomposition (Junkins and Kim 1993). Because the rank of $C^T(k) \in R^{3 \times 6}$ is always three due to the orthogonal configuration of the conventional IMU, $C^T(k)$ is factored as

$$C^T(k) = U \Sigma V^T = [U_1 \ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \quad (47.16)$$

where $U \in R^{3 \times 3}$ and $V \in R^{6 \times 6}$ are normalized unitary matrices, $\Sigma \in R^{3 \times 6}$, and Σ_1 is a diagonal matrix of three singular values of $C^T(k)$. The first three columns of V are orthonormal vectors spanning the column space of $C^T(k)$, and the last three columns of V are orthogonal vectors spanning the null space of $C^T(k)$. Note that all the orthogonal vectors v_i lying in the null space of $C^T(k)$ satisfy $C^T(k) v_i = 0$; therefore, $H_y(k)$ is the transpose of an orthogonal set of the null space of $C^T(k)$ as follows:

$$C^T(k) [v_1 \ v_2 \ v_3] = C^T(k) V_2 = 0 \quad (47.17)$$

$$H_y(k) = V_2^T \quad (47.18)$$

Using Eqs. (47.12)–(47.14), the parity vector can be represented as

$$p(k) = H_y(k) [C(k)x(k) + w(k) + f(k)] = H_y(k)w(k) + H_y(k)f(k) \quad (47.19)$$

This relationship indicates that the parity vector is a function of the noise signal $w(k)$ and the fault signal $f(k)$, not the system states $x(k)$ and the control input $u(k)$. In this study, a fault detection function is defined as the norm of the parity vector as

$$F_D(k) = \|p(k)\| = \sqrt{p(k)^T p(k)} \quad (47.20)$$

When all sensors are normal, $F_D(k)$ is close to zero as long as all sensor signals are unbiased with zero-mean noise. Once a fault occurs in a certain sensor, the term $H_y(k)f(k)$ in Eq. (47.19) becomes nonzero. As a result, $F_D(k)$ becomes larger than a predetermined threshold for a given confirmation time t_c , then it is declared that the fault occurred. Here, the threshold values are chosen as $3 - \sigma$ values for each fault case as follows:

$$T_1 = k_1 (3 \times \sqrt{3} \times \sigma_{\max}), \quad T_2 = k_2 (3 \times \sqrt{2} \times \sigma_{\max}), \quad T_3 = k_3 (3 \times \sigma_{\max}) \quad (47.21)$$

where T_i is a threshold value for the i -th fault occurrence, k_i is a safety factor, and σ_{\max} is a maximum standard deviation among all inertial sensors. The safety factor makes the threshold value larger than $E[H_y(k)w(k)]$ and smaller than $E[H_y(k)w(k) + H_y(k)f(k)]$; this can be further tuned by trial and error. Note that $\sqrt{3}$ and $\sqrt{2}$ are related to the dimensions of the matrix $H_y(k)$.

A fault isolation function of the j -th sensor is defined as an inner product of the j -th normalized column vector in $H_y(k)$ and the parity vector $p(k)$ as follows:

$$F_{I,j}(k) = h_j^T p(k) / \|h_j\| \quad (47.22)$$

where h_j is the j -th column vector of $H_y(k)$. The parity vector indicates the direction of the fault vector $f(k)$ in the space of $H_y(k)$ because $p(k)$ is a linear transformation of $f_i(k)$ by $H_y(k)$ in Eq. (47.19). If all sensors are normal, then all fault isolation values are zero. If a fault occurs in the j -th inertial sensor, then the fault isolation function value of the j -th inertial sensor becomes equal to $h_j^T h_j f(k) / \|h_j\|$. In a geometrical sense, the parity vector including the fault signal indicates the direction of the column vector h_j of the faulty sensor. Therefore, the j -th inertial sensor having the largest fault isolation value should be isolated, and then a new parity vector is reconstructed using the remaining healthy sensors.

In this study, six inertial sensors are initially considered. The first and second faults in any inertial sensors will be detected by Eq. (47.20) and isolated by Eq. (47.22). However, the third fault can be detected but cannot be isolated because (i) the parity vector is not a vector with a specific direction but rather a scalar and (ii) the fault signal that is transformed into the column space of $H_y(k)$ is not always dominant in that column space.

47.2.2 In-Lane Monitoring Method Based on the Discrete Wavelet Transform

In the parity space method described in Sect. 47.2.1, the third fault in any of the four remaining inertial sensors cannot be isolated due to the lack of the parity vector. In other words, the parity space method isolates the fault only if sufficient numbers of inertial sensors are provided. On the other hand, the in-lane monitoring method is based on the discrete wavelet transform of each sensor signal. Not only the first and second faults but also the third fault in inertial sensors can be detected and isolated by the in-lane monitoring method. However, the in-lane monitoring method might give a false alarm when an aircraft is maneuvering rapidly because any sudden change in the signal could be detected by the discrete wavelet transform. Because of this, the in-lane monitoring method should be combined with the parity space approach (Gilmore and McKern 1970). Note that the parity space method is not affected by a sudden change in the signal as long as the parity vector is not changed. Consequently, this study proposes the first and second faults of inertial sensors be detected and isolated with the parity space method, and the third fault in one of the four remaining inertial sensors be detected and isolated using a combination of the parity space method and the in-lane monitoring method.

Fourier transform has been widely used for analyzing stationary signals; however, it is less useful for analyzing nonstationary signals because of its time-varying frequency contents (Bogges and Narcowich 2001). On the other hand, the wavelet transform provides time-frequency localization (Daubechies 1992) in the sense that it can zoom in on the high-frequency signal for a short duration or zoom out on slow oscillations (Mallat 1999; Li 1999). Furthermore, in terms of the computational load, the wavelet transform is better than the windowed Fourier transform because the discrete wavelet transform requires $\log_2 N$ computational steps, whereas the fast Fourier transform requires $N \log_2 N$ steps for a discrete signal of given length N .

Recently, multi-resolution signal decomposition based on the orthonormal properties of the discrete wavelet transform has been widely used to divide the sensor signal into two parts, an approximated signal and a detailed signal (Mallat 1989). These signals are obtained by convolving the sensor signal with a type of low-pass filter for “the approximations” and a type of high-pass filter for “the details.” This also helps to reduce the computational load of the next step’s decomposition by decreasing the number of samples. Note that the detailed signal contains noise or fault signals such as sharp edges, transitions, jumps at the bias, drifts, and spikes.

Consider a discrete-time sensor signal $c_j(n)$ at the discrete-time step n . This is divided into an approximated part $c_{j+1}(n)$ and a detailed part $d_{j+1}(n)$ using the following relationship:

$$c_{j+1}(n) = \sum_{k=-\infty}^{\infty} h(k-2n) c_j(n) \quad (47.23)$$

$$d_{j+1}(n) = \sum_{k=-\infty}^{\infty} g(k-2n) c_j(n) \quad (47.24)$$

where $h(n)$ and $g(n)$ are the low-pass and high-pass filter coefficients, respectively, which can be determined from the following two-scale relationship between the scaling function $\phi(t)$ and the wavelet function $\psi(t)$.

$$\phi(t) = \sqrt{2} \sum_{n=-\infty}^{\infty} h(n) \phi(2t-n) \quad (47.25)$$

$$\psi(t) = \sqrt{2} \sum_{n=-\infty}^{\infty} g(n) \phi(2t-n) \quad (47.26)$$

where

$$g(n) = (-1)^{1-n} h(1-n) \quad (47.27)$$

If the scaling function $\phi(t)$ is defined as the signal basis, the low-pass filter coefficients $h(n)$ can be obtained by Eq. (47.25). Note that because the design of the scaling function and the wavelet basis is beyond the scope, Daubechies wavelets are selected for the application in this study. A detailed description of the wavelet selection will be discussed further at the end of this section. The low-pass filter coefficients $g(n)$ and the wavelet function $\psi(t)$ can be successively designed in Eq. (47.26). Figure 47.2 describes the process of signal decomposition into the approximated signal and the detailed signal.

The following example is provided to illustrate the mechanism of the discrete wavelet transform. If the low-pass filter is $h(0) = 1/\sqrt{2}$, $h(1) = 1/\sqrt{2}$, then the high-pass filter can be computed using Eq. (47.27) as

$$g(0) = -1/\sqrt{2}, \quad g(1) = 1/\sqrt{2} \quad (47.28)$$

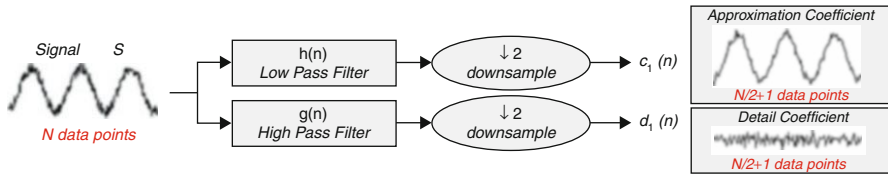


Fig. 47.2 Multi-resolution signal decomposition

Consider the following discrete-time sensor signal:

$$c_0 = (0.9, 1.1, 1.5, 1.3) \quad (47.29)$$

This is divided into the approximated part and the detailed part using Eqs. (47.23) and (47.24).

$$c_1 = (1.0, 1.4), \quad d_1 = (-0.1, 0.1) \quad (47.30)$$

The multi-resolution signal decomposition technique can localize and discriminate a high-frequency signal from the original signal, and therefore it can be applied to detect a sensor fault included in the high-frequency signal. For this reason, the choice of a wavelet basis is important for fault diagnosis. The first consideration for selecting a wavelet basis is a vanishing moment. If a wavelet function satisfies the following condition, it has n vanishing moments:

$$\int t^k \psi(t) dt = 0 \text{ for } k = 0, 1, \dots, n-1 \quad (47.31)$$

This means that the larger its vanishing moment is, the larger the order polynomials it can suppress will be. In other words, it can make the wavelet transform more localized by removing smooth functions. Another consideration is the support size. The shorter the support size, the better time localization it has. Daubechies wavelets have good time localization and orthogonal properties because they have the shortest compact support for a given number of vanishing moments. Therefore, Daubechies wavelets are considered as the wavelet basis in this study. It is well-known that short and fast transient singularities can be easily detected using a Daubechies 2 or 4 wavelet, whereas slow-moving anomalies, such as drift, can be detected using Daubechies 8 or 10. In this study, a Daubechies 2 wavelet was adopted to detect the fault signal because sensor faults usually have fast transient characteristics caused by instantaneous physical shock or electromagnetic interference. The high-pass filter coefficients $h(n)$ of the Daubechies 2 wavelet are shown in the following equations (Mallat 1999):

$$\begin{aligned}
h[0] &= 0.482962913145 \\
h[1] &= 0.836516303738 \\
h[2] &= 0.224143868042 \\
h[3] &= -0.129409522551
\end{aligned} \tag{47.32}$$

The threshold for FDI can be set by modifying the threshold of the wavelet shrinkage technique generally used for de-noising as follows (Kim et al. 2007; Dohono and Johnstone 1994):

$$T_4 = k_w \sigma_w \sqrt{2 \ln N_w} \tag{47.33}$$

where N_w is the length of the signal, σ_w is the standard deviation of the measurement noise, and k_w is the safety factor, which can be determined considering the system characteristics and the operational environment.

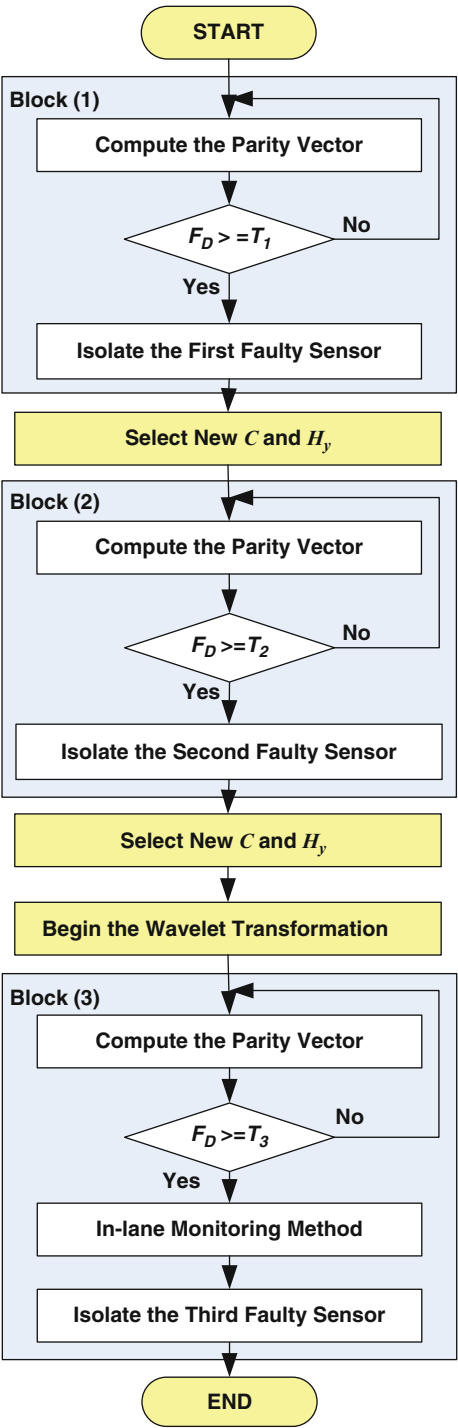
In this study, the discrete wavelet transform is conducted using 32 samples of the sensor data $c_0 = [y(k-31), y(k-30), \dots, y(k-1), y(k)]$, and fault detection is based on the maximum value of the detailed signal $[d_1(k-16), d_1(k-15), \dots, d_1(k-1), d_1(k)]$ to grab the high-frequency faulty signals.

47.2.3 Model-Free Hybrid Fault Detection and Isolation

The model-free hybrid FDI combines the parity space method with the in-lane monitoring method based on the discrete wavelet transform. Successive faults in the skew-configured IMU are detected by the parity space method and then isolated by the in-lane monitoring method as well as the parity space method. This diagnosis is known as a model-free scheme because it does not require any dynamic model of the aircraft. A flowchart of the model-free hybrid FDI technique is shown in Fig. 47.3. Triple faults successively occurring in the inertial sensors are detected and isolated by the following procedures:

1. When the first fault occurs in the i -th inertial sensor, the fault detection function F_D in Eq. (47.20) exceeds the threshold T_1 , and the occurrence of the fault is detected. The fault isolation functions $F_{I,i}$ in Eq. (47.22) for each sensor are calculated, and the sensor with the largest value is isolated from the measurement as shown in Block (1) of Fig. 47.3. The output matrix C and the parity matrix H_y are selected according to the remaining five sensors.
2. When the second fault occurs in the j -th inertial sensor, it can be detected and isolated similar to the above process as shown in Block (2) of Fig. 47.3. Then, the discrete wavelet transform of the remaining four sensor signals begins immediately after the second fault isolation.
3. When the third fault occurs in the k -th sensor, F_D exceeds the threshold T_3 , and the fault occurrence is detected. Then, a sudden change in the k -th sensor is detected by the in-lane monitoring method based on the discrete wavelet transform, and the corresponding sensor is isolated from the inertial measurement system as shown in Block (3) of Fig. 47.3.

Fig. 47.3 Flowchart of hybrid fault detection and isolation



In this way, triple successive faults can be treated during the flight, and the UAV can perform its mission continuously. A measurement matrix C , which is originally 6-by-3, is reduced to a 5-by-3 matrix after the first fault isolation and is downsized to a 4-by-3 matrix after the second fault isolation. A parity matrix H_y , which is originally 3-by-6, is reduced to a 2-by-5 matrix after the first fault isolation and is reset to a 1-by-4 matrix after the second fault isolation.

47.3 Hardware Configuration for the Flight Test

The skew-configured IMU and the fault signal generation board were constructed to verify the performance of the proposed hybrid FDI scheme. The fault signal generation board produces virtual fault signals instead of real fault signals. The hybrid FDI algorithm is processed in the flight control computer. The performance of the proposed FDI is verified through ground and flight experiments. Figure 47.4 describes the system configuration and the test procedure. The details of the design/calibration procedures and functions of the skew-configured IMU, the fault signal generation board, and the fixed-wing UAV are described below.

47.3.1 Skew-Configured Inertial Measurement Unit

The primary IMU of the experimental UAV system is the *Crossbow AHRS400*. The secondary IMU is the *XSENS MTi-68*, which is a small and low-cost IMU compared to the primary IMU. The secondary IMU is mounted on the inclined plane with respect to the aircraft body coordinate, whereas the primary IMU is aligned with the aircraft body coordinate. The inclination angle of the secondary IMU is determined by optimizing the navigation index and the fault detection performance index. A detailed optimization procedure can be found in the references (Park 2004). Figure 47.5 illustrates the designed structure of the mounting surface for the secondary IMU.

As shown in Fig. 47.5, the primary IMU is aligned with the aircraft body coordinate (X_1, Y_1, Z_1) , whereas the secondary IMU is aligned with the slope plane (X_2, Y_2, Z_2) . As a result, six inertial sensors are configured in a cone shape. With this configuration, the measurement matrix $C(k)$ and the parity matrix $H_y(k)$ in Eqs. (47.12) and (47.13) are analytically calculated as follows:

$$C(k) = \begin{bmatrix} -1 & 0 & 0 & 0.6667 & 0.6667 & -0.3333 \\ 0 & 1 & 0 & -0.3333 & 0.6667 & 0.6667 \\ 0 & 0 & -1 & 0.6667 & -0.3333 & 0.6667 \end{bmatrix}^T \quad (47.34)$$

Fig. 47.4 System configuration and test procedure

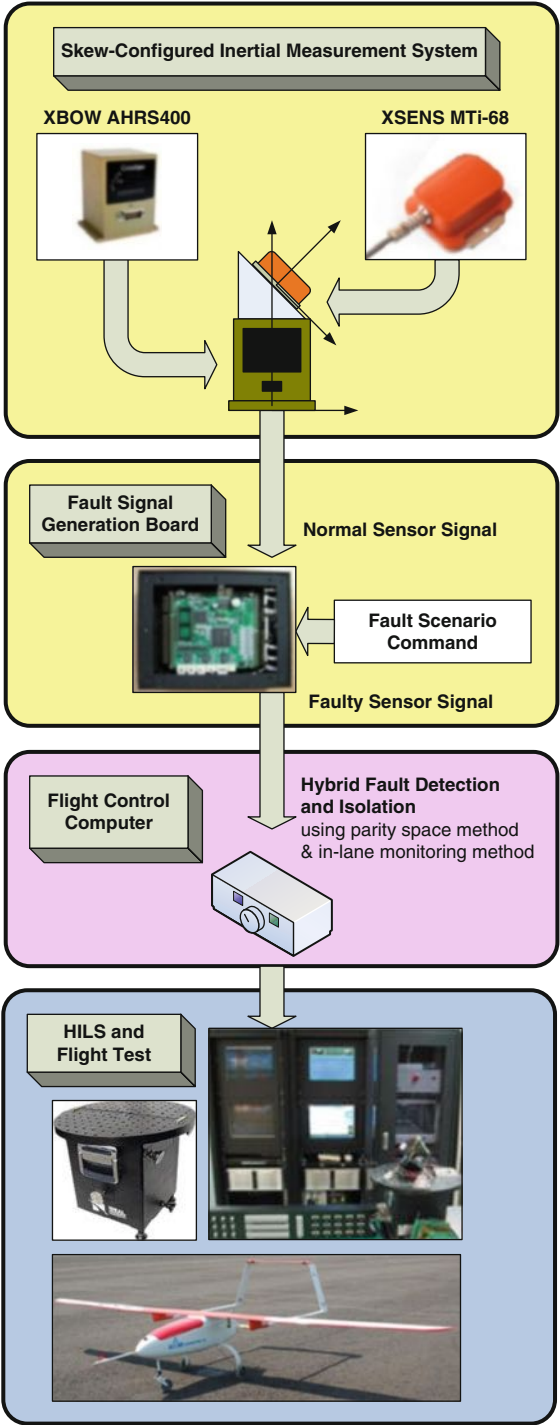
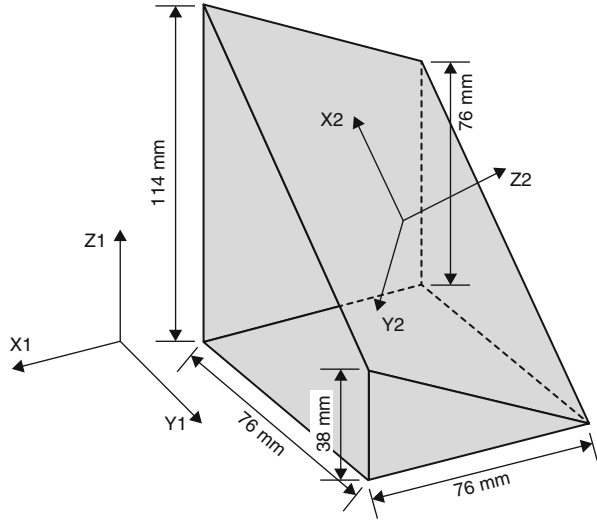


Fig. 47.5 Configuration of two inertial measurement units



$$H_y(k) = \begin{bmatrix} 0.4714 & 0.2357 & 0.4714 & 0.7071 & 0.0000 & 0.0000 \\ 0.4714 & -0.4714 & -0.2357 & 0.0000 & 0.7071 & 0.0000 \\ -0.2357 & -0.4714 & 0.4714 & 0.0000 & 0.0000 & 0.7072 \end{bmatrix} \quad (47.35)$$

In practice, the skew-configured IMU contains small alignment errors in $C(k)$ and $H_y(k)$. Therefore, the assembled IMUs should be calibrated with an initialization experiment, which compensated for the bias of the skew-configured IMU with a 5-min history of each signal in the stationary condition. The sensor system was placed on a single-axis rate table, *Ideal Aerosmith 1291BR*, and was rotated about three different axes (X_1, Y_1, Z_1) with six constant angular velocities: 100, 50, 20, -20, -50, and -100 deg/s. Using the signals from these experiments, the measurement matrix $C(k)$ in Eq. (47.12) with $w(k) = 0$ and $f(k) = 0$ was computed as

$$C(k) = z(k) x_r^\dagger(k) \quad (47.36)$$

where $z(k)$ is a 6-by-18 output matrix of the skew-configured IMU and $x_r^\dagger(k)$ is a pseudo inversion of an 18-by-3 reference input matrix. The columns of $z(k)$ comprise 18 rotating conditions; these are the rotating axis X_1 with 100 deg/s, the rotating axis Y_1 with 100 deg/s, the rotating axis Z_1 with 100 deg/s, and so on. As a result of this initialization experiment, the calibrated $C(k)$ and $H_y(k)$ were obtained as follows:

$$C(k) = \begin{bmatrix} -1.0018 & 0.0031 & -0.0063 & 0.6885 & 0.65889 & -0.3041 \\ -0.0030 & 1.0018 & -0.0121 & -0.3163 & 0.6554 & 0.6865 \\ -0.0021 & -0.0028 & -0.9999 & 0.6555 & -0.3660 & 0.6635 \end{bmatrix}^T \quad (47.37)$$

$$H_y(k) = \begin{bmatrix} 0.4845 & 0.2295 & 0.4613 & 0.7070 & 0.0012 & -0.0001 \\ 0.4656 & -0.4644 & -0.2595 & -0.0002 & 0.7073 & -0.0012 \\ -0.2185 & -0.4809 & 0.4700 & -0.0007 & 0.0017 & 0.7072 \end{bmatrix} \quad (47.38)$$

47.3.2 Fault Signal Generation Board

In practice, it is difficult to make the inertial measurement sensors partially fail during flight. A fault signal generation board is considered to produce virtual fault signals instead of real fault signals. The fault signal generation board receives normal sensor signals from the skew-configured IMU and adds virtual fault signals to normal sensor signals according to the failure scenario. The modified sensor signals are transmitted to a flight control computer that will perform FDI. In this way, inertial measurement sensors can be virtually broken by the fault generation board.

The fault signal generation board was designed with digital signal processing units: *Texas Instrument's TMS320LF2407A* and *TL16C754BPN*. It has 5 serial communication ports and 20 digital input/output ports. The fault signal generation board has a communication port and six light-emitting diodes on the outside. Through its 15-pin communication port, 2 IMUs are connected to the fault signal generation board. In addition, commands from the ground station and the corresponding virtual sensor signals are communicated through this port.

47.3.3 Experimental Unmanned Aerial Vehicle System

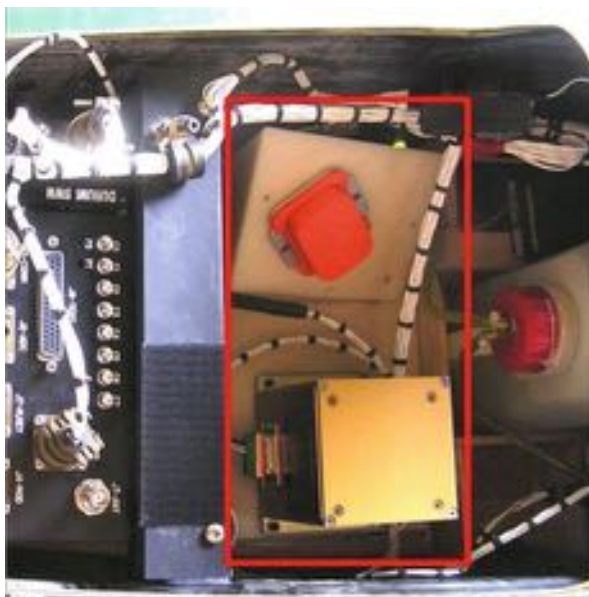
The performance of the proposed FDI approach was verified with an experimental UAV, *Durumi 2*, developed and operated by the *Korea Aerospace Research Institute*. This UAV has performed various flight experiments for real-time system identification and fault-tolerant control systems (Park et al. 2006). Its wing span is 4.8 m, and its body length is 2.7 m as shown in Fig. 47.6. Its total weight is 37 kg, including a maximum payload weight of 12 kg. The maximum speed is 120 km/h, and its stall speed is 55 km/h. In addition, all of the control surfaces are fractioned into two pieces able to move independently for a fault-tolerant mission.

The skew-configured IMU was loaded on the payload of this experimental UAV as shown in Fig. 47.7. A hexahedron in the lower right corner is the primary IMU, the *Crossbow AHRS400*. An upper flat unit on the slope is the redundant secondary IMU, *XSENS MTi-68*, which is smaller than the primary unit. Note that the secondary unit is cheaper and has a poorer resolution. These two IMUs are directly connected to the fault signal generation board. The modified sensor output in the fault signal generation board is transmitted to the flight control computer.

Fig. 47.6 Experimental UAV, Durumi 2



Fig. 47.7 Skew-configured inertial measurement system in the payload



47.4 HILS Test and Flight Experiments

HILS test and flight experiments were performed to evaluate the performance of the proposed hybrid FDI system. Note that the operational inspection of the FDI algorithm was done in advance by a ground test using the single-axis rate table. The fault diagnosis performance of the HILS and the flight test are discussed below.

47.4.1 HILS Test Results

The objective of the ground experiment was to verify the hybrid FDI algorithm and determine the parameters of the algorithm. The bias and misalignment errors of the skew-configured IMU were calibrated with the initialization experiment. The ground test also inspected the function of the fault signal generation board.

Table 47.1 Parameters for the hybrid fault detection and isolation

| Parameter | Variable | Value |
|--|-------------------|-----------------|
| Maximum standard deviation of the inertial sensor signals in Eq. (47.21) | σ_{\max} | 0.81 |
| Safety factors of the fault detection in Eq. (47.21) | (k_1, k_2, k_3) | (1.5, 1.5, 1.3) |
| Confirmation time of the fault detection function | t_c | 0.25 (s) |
| Safety factor of the threshold in Eq. (47.33) | k_w | 1.5 |
| Sampling number for the wavelet transform in Eq. (47.33) | N_w | 32 |

As a result of various ground experiments including constant rotations, oscillations, and random vibrations, the FDI parameters for the parity equation and the maximum absolute value of the wavelet transform at each sampling time (this will be called “the discrete wavelet transform” hereafter for simplicity) were determined as summarized in Table 47.1.

Figure 47.8 shows one of the ground experimental results: the sensor signals, the parity norm, the discrete wavelet transform, and the isolated sensor number, respectively. In this test, 10 deg/s bias faults successively occurred in Sensors 1, 2, and 5. The first fault occurred in Sensor 1 at 6.2 s. As shown in the second plot, the fault detection function $F_D(k)$, the norm of the parity vector, increased when the fault occurred. That is, the first fault was detected when the residual exceeded the prescribed threshold T_1 . Simultaneously, the fault isolation function $F_{I,1}(k)$ declared Sensor 1 to be faulty, as shown in the bottom plot, and therefore, Sensor 1 is isolated from the measurement.

In the same way, the second fault occurring in Sensor 2 at 8.4 s was detected when the residual exceeded the threshold T_2 as shown in the second plot of Fig. 47.8. Among the remaining five sensors, the fault isolation function $F_{D,2}(k)$ declared Sensor 2 to be the faulty sensor as shown in the bottom plot. Then, Sensor 2 was isolated from the measurement. At that time, the wavelet transform of the remaining four sensor signals began as shown in the third plot.

Whereas the first and the second faults are detected and isolated by the parity space method, the third faults are detected by the parity space method and isolated by the in-lane monitoring method. The second plot of Fig. 47.8 shows that the residual exceeded the threshold T_3 when the third fault occurred in Sensor 5 at 11.2 s. Simultaneously, the faulty Sensor 5 was detected and isolated by monitoring the wavelet transform as shown in the third and fourth plots. In brief, the results of the ground HILS test showed the viability of the proposed approach and encouraged the necessity for real flight experiments.

47.4.2 Flight Experimental Result

The fault diagnosis performance of the hybrid FDI system was also verified through the flight test. The command for the type and magnitude of the fault signal is transmitted from the ground station according to the sensor failure scenario.

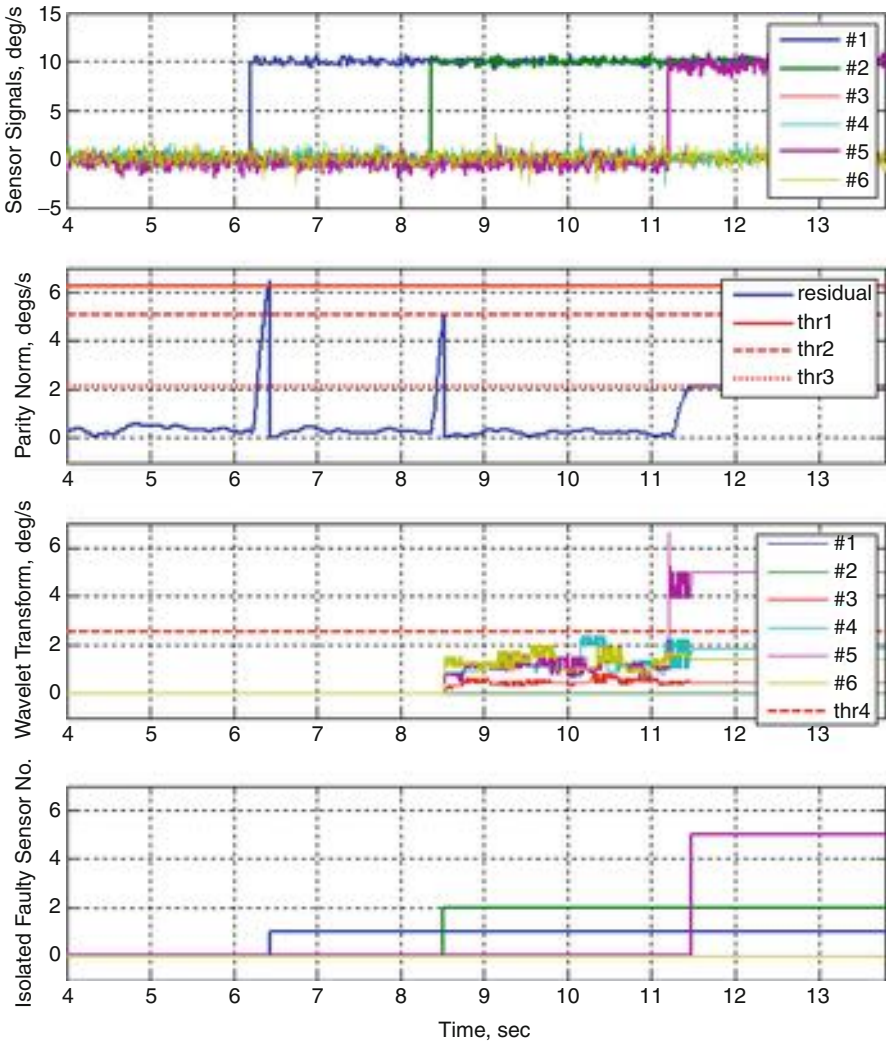


Fig. 47.8 Ground test result of the hybrid fault detection isolation system

In the flight test, two types of sensor fault were considered, a bias fault and a stuck fault. The bias and stuck faults are basic errors that can be generated in sensors due to both hardware and software problems. If the sensor is badly calibrated before or during the flight, the measurement can be easily biased in the decoding process. After hardware failure, the sensor output might be stuck to any last value that was normal or biased in the computing algorithm. The detectable magnitude of the sensor fault is numerically computed according to the noise level and the configuration of the skew-configured IMU. Once the parity matrix $H_y(k)$

in Eq. (47.13) and the threshold T_i in Eq. (47.21) are determined, the fault magnitude, which can be always detected, is computed as

$$\|f(k)\| = T_i / \|h_j\|_{\min} \tag{47.39}$$

where h_j is the j -th column vector of $H_y(k) = [h_1 \ h_2 \ \dots \ h_m]$. When all six inertial sensors are normal, the constantly detectable magnitude is 5.9 deg/s. When five inertial sensors are normal, the constantly detectable magnitude is 6.6 deg/s. When four inertial sensors are normal, the constantly detectable magnitude is 10.9 deg/s. If the corresponding column vector h_j is larger than the denominator of Eq. (47.39) in a certain case, the sensor fault smaller than the constantly detectable magnitude can be detected usually but not always. Note that the detectable fault magnitude increases from 5.9 to 10.9 as the number of normal sensors decreases.

Thirty-six experiments were performed during the flight test according to the failure scenario, including seven sets of successive triple faults. Table 47.2 summarizes the fault types and magnitudes of each sensor failure scenario. Although the parameters of the hybrid FDI system had been shown to be tolerant to 10 deg/s bias errors in the ground test, the fault diagnosis performance in the flight test was superior than the expected performance from the ground test. Therefore, three additional failure scenarios (5 through 7) were tested.

Table 47.3 summarizes the flight experimental results for each sensor failure scenario. In the flight test, the aircraft maneuvered to perform level flights, bank turns, climbs, and descents. The result of the fault isolation is simply displayed as follows: success, partial success, and fail. In failure scenarios 1 through 5, the proposed FDI technique successfully detected and isolated the successive sensor faults under various real flight situations. In failure scenarios 6 and 7, all of the first/second faults were successively detected and isolated by the proposed FDI system, but the third fault was partially isolated in some cases even though all of the third faults were detected by the parity space method without any problems. Those cases that missed the fault isolation will be discussed in detail in the next subsection.

Next, one of the flight experimental results is discussed in detail. The UAV was in a coordinate turn for 23 s. The mean velocity and the mean altitude over the ground were 35 m/s and 170 m, respectively. The successive stuck and bias faults

Table 47.2 Sensor failure scenarios commanded from the ground station

| Scenario number | First fault | | | Second fault | | | Third fault | | |
|-----------------|-------------|-------|--------|--------------|-------|--------|-------------|-------|--------|
| | deg/s | Type | Sensor | deg/s | Type | Sensor | deg/s | Type | Sensor |
| 1 | 15 | Bias | 1 | 15 | Bias | 3 | 15 | Bias | 5 |
| 2 | 15 | Bias | 2 | 15 | Bias | 4 | 30 | Stuck | 6 |
| 3 | 10 | Bias | 1 | 10 | Bias | 3 | 10 | Bias | 5 |
| 4 | 10 | Bias | 2 | 10 | Bias | 4 | 30 | Stuck | 6 |
| 5 | 8 | Bias | 1 | 10 | Stuck | 3 | 8 | Bias | 5 |
| 6 | 8 | Bias | 2 | 10 | Stuck | 4 | 8 | Bias | 6 |
| 7 | 10 | Stuck | 5 | 8 | Bias | 1 | 10 | Stuck | 3 |

Table 47.3 Flight experimental results of the fault detection and isolation

| Scenario number | Flight status | FDI parameters | Fault isolation (success/total) |
|-----------------|------------------|---|---------------------------------|
| 1 | Level flight | $k_1 = 1.8, k_2 = 1.8, k_3 = 1.5, t_c = 0.12$ | Success (2/2) |
| | Bank turn | | Success (2/2) |
| | Climb/descent | | Success (2/2) |
| 2 | Level flight | $k_1 = 1.5, k_2 = 1.5, k_3 = 1.2, t_c = 0.12$ | Success (2/2) |
| | Bank turn | | Success (2/2) |
| | Climb/descent | | Success (2/2) |
| 3 | Level flight | $k_1 = 1.3, k_2 = 1.3, k_3 = 1.1, t_c = 0.12$ | Success (2/2) |
| | Bank turn | | Success (2/2) |
| | Climb/descent | | Success (2/2) |
| 4 | Bank turn | | Success (2/2) |
| 5 | Level flight | $k_1 = 1.3, k_2 = 1.3, k_3 = 1.1, t_c = 0.12$ | Success (2/2) |
| | Bank turn | | Success (3/3) |
| | Climb/descent | | Success (2/2) |
| 6 | Level flight | | Partial success (2/4) |
| | Bank turn | | Success (2/2) |
| | Climb/descent | | Partial success (2/4) |
| 7 | Roll oscillation | | Partial success (1/3) |

were generated in Sensors 5, 1, and 3 according to the failure scenario 7. Figure 47.9 shows the sensor signals, the parity norm, the discrete wavelet transform, and the isolated sensor number, respectively. The first fault took place in Sensor 5 at 1,274.5 s. Its signal was stuck at 10 deg/s as shown in the first plot. The fault detection function $F_D(k)$ increased immediately when the fault occurred, as shown in the second plot. The first fault was detected when this residual exceeded the prescribed threshold T_1 . Then, the fault isolation function $F_{I,5}(k)$ declared that Sensor 5 was the faulty sensor, as shown in the fourth plot. Consequently, Sensor 5 was isolated from the skew-configured inertial measurement system, and the fault diagnosis parameters were updated for a five-sensor configuration.

The second fault, an 8 deg/s bias error in Sensor 1 at 1,283.5 s, was detected in the same way. The wavelet transform of the remaining four inertial sensor signals began after the isolation of the second faulty inertial sensor. When the third fault occurred in Sensor 3, which stuck at 10 deg/s after 1,287.8 s, the residual exceeded the threshold T_3 , as shown in the second plot of Fig. 47.9. The parity space method could no longer identify the faulty sensor because of an insufficient number of remaining healthy sensors. However, the wavelet transform of each sensor signal could identify the timing of the abrupt signal change caused by the fault. As shown in the third plot, the wavelet-transformed signal of Sensor 3 exceeded the predefined threshold T_4 and that sensor was thus successfully isolated. Other flight experiments in Scenarios 1 through 5 showed similarly satisfactory performance; the UAV was able to continue performing its mission with the skew-configured IMU composed of the remaining three sensors.

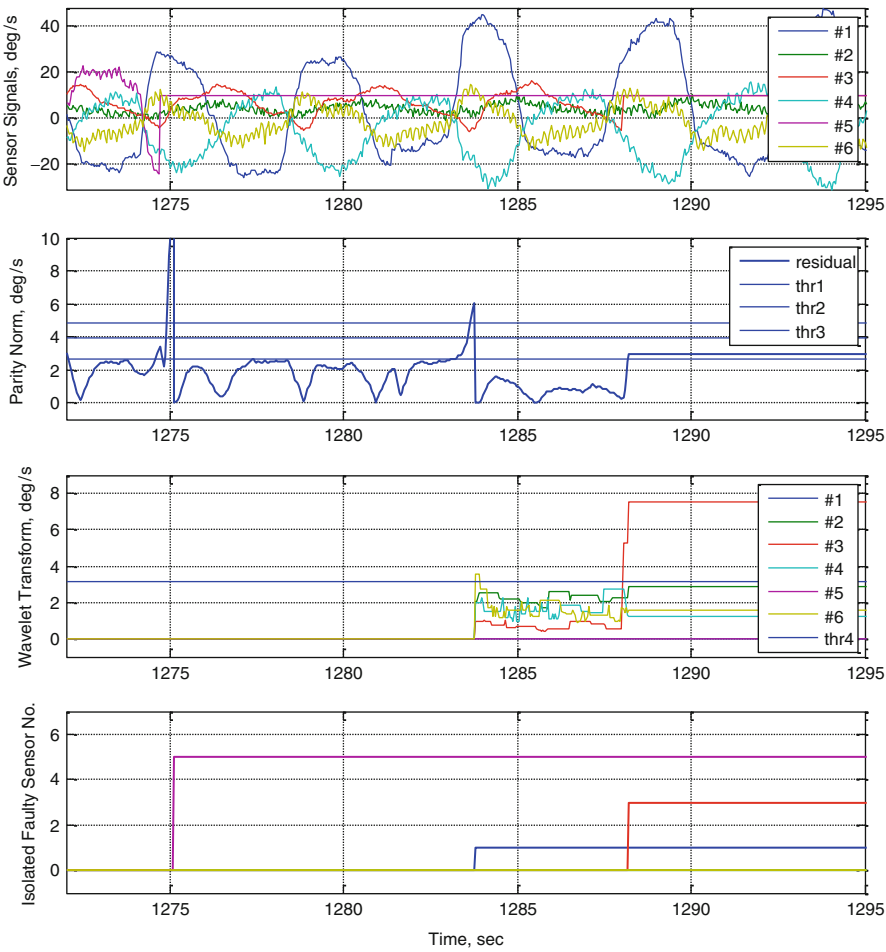


Fig. 47.9 Flight test results of the hybrid fault detection and isolation system

47.4.3 Analysis

47.4.3.1 Comparison of the State Vector with or Without FDI

The performance of the proposed FDI technique against three successive faults can be analyzed by comparing the body-coordinate sensor signals collected during the faulty situation to those collected during a healthy, fault-free situation. The healthy body-coordinate angular rates are calculated from normal sensor signals, and the faulty body-coordinate angular rates are calculated from all six sensor signals.

Figures 47.10 and 47.11 illustrate the difference between the healthy and the faulty angular rates of failure scenario 7 without and with application of the FDI scheme, respectively. As shown in Fig. 47.10, without the application of the

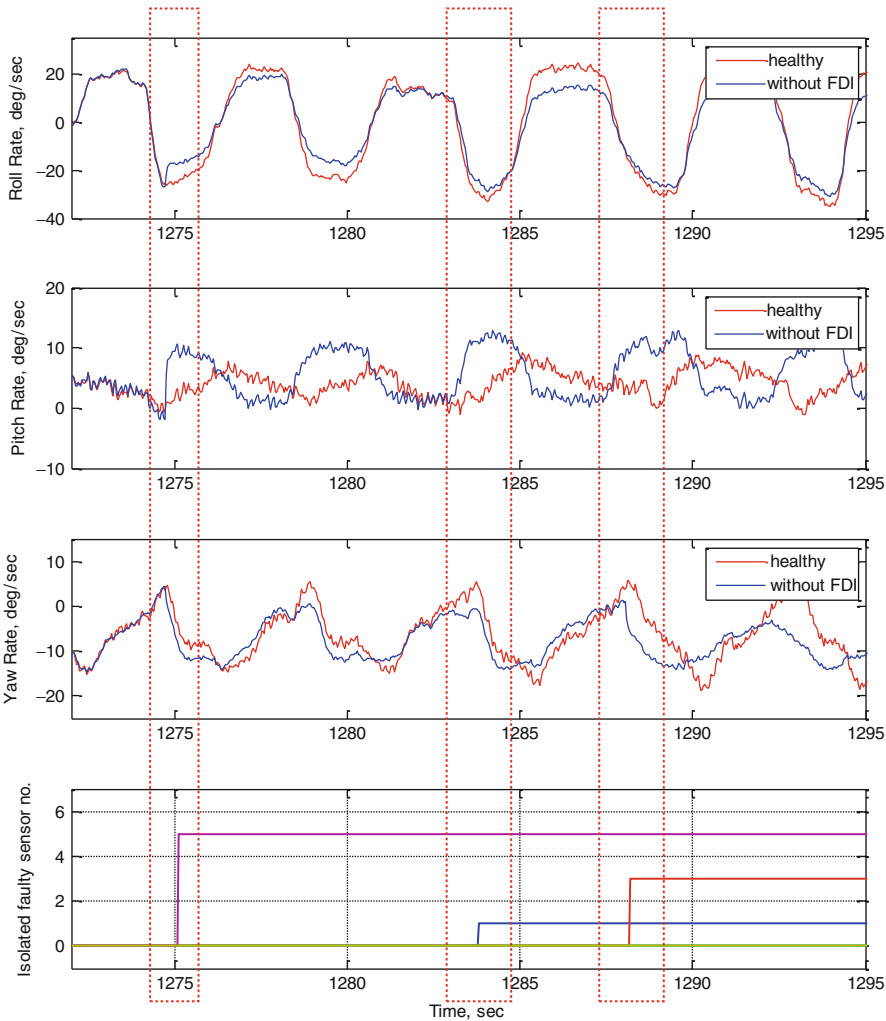


Fig. 47.10 Comparison between healthy and faulty signals without FDI

hybrid FDI, the body-coordinate sensor output (blue line) could not provide accurate signals (red line) when successive faults occurred according to failure scenario 7 (Sensor 5 stuck at 10 deg/s, Sensor 1 bias of 8 deg/s, and Sensor 3 stuck at 10 deg/s, successively). Abrupt changes in the roll, pitch, and yaw rate signals due to the stuck faults in Sensors 5 and 3 can be seen at 1,274.5 and 1,288.1 s, respectively. Notably, the error between the healthy and faulty signals increased suddenly when the first stuck fault occurred. The error became even larger and more remarkable after the second and third faults.

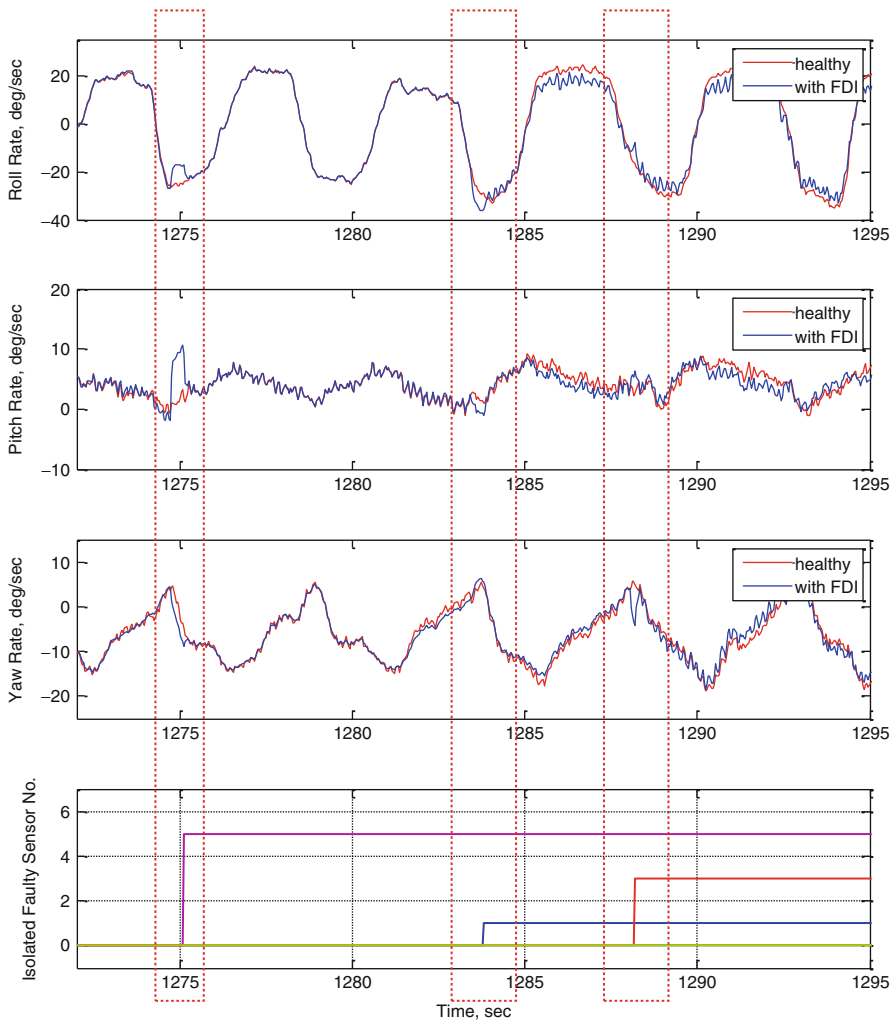


Fig. 47.11 Comparison between healthy and faulty signals with FDI

Figure 47.11 shows that the proposed FDI scheme enhances the accuracy of the body-coordinate sensor output. The computed body-coordinate sensor outputs (blue line) are consistent with the healthy signals (red line) except the short interval when the faults occurred because the FDI algorithm requires almost a half second to detect and isolate the faulty sensor. Table 47.4 presents a quantitative error analysis based on the error norm between the healthy and faulty signals. The errors obtained without FDI decreased by up to 29 % when the proposed FDI is applied. Therefore,

Table 47.4 Error norm analysis between healthy and faulty signals

| Axis | w/o FDI | w/ FDI | Ratio |
|------------|---------|--------|--------|
| Roll rate | 139.02 | 82.68 | 1:0.59 |
| Pitch rate | 121.57 | 32.58 | 1:0.29 |
| Yaw rate | 98.42 | 35.95 | 1:0.37 |

the flight experiments demonstrated that the proposed hybrid FDI technique with the skew-configured IMU can continuously provide healthy sensor outputs despite successive stuck and bias faults in the individual sensors.

47.4.3.2 False Alarm Due to the Smooth Transition

As summarized in Table 47.3, several partially successive tests of failure scenarios 6 and 7 failed to isolate the third fault in the skew-configured IMU. Figure 47.12 shows one of the partially successive FDI results of failure scenario 7. In this scenario, three successive faults occurred: Sensor 5 stuck at 10 deg/s, Sensor 1 experienced an 8 deg/s bias error, and Sensor 3 stuck at 10 deg/s. As depicted in the fourth plot, the first and second faulty sensors were correctly isolated but the third faulty sensor was not isolated because the wavelet-transformed signals did not exceed the threshold.

As can be seen in the first plot of Fig. 47.12, the third fault in Sensor 3, which stuck at 10 deg/s after 1,395.5 s, did not generate a sufficiently remarkable signal change compared to the general sensor noise to be identified by the wavelet transform based on the Daubechies 2 wavelet function. Because the wavelet transform is a signal processing technique based on each sensor output, the fault may not be identified if the magnitude of the fault signal is smaller than the magnitude of the noise signal or if noise characteristics vary over time.

Since the wavelet transform is primarily designed to avoid emergencies when a third fault occurs, the UAV should return to base by homing guidance if the parity space method detects a third fault but the in-lane monitoring system is unable to identify the failed sensor. If the UAV cannot return to base, it may be useful to generate a virtual sensor signal based on model-based propagation after the third fault. Nevertheless, this approach would yield errors if the discrepancy caused by the modeling uncertainties exceeds the fault signal. Alternatively, the wavelet transform could be run in parallel based on the higher scale’s wavelet basis function (e.g., Daubechies 9 or 10) to identify a slow or weak signal change. The capability of the flight control computer should be considered since this method requires a higher computation load.

47.4.3.3 False Alarm Due to the Computational Delay

The computational delay in the wavelet transformation is another cause of the partially successful results. Confirmation of the presence of a faulty signal involves six steps of the fault detection algorithm and requires 0.12 s. A faulty signal that lasts less than 0.12 s is considered temporary noise and may be ignored. Moreover,

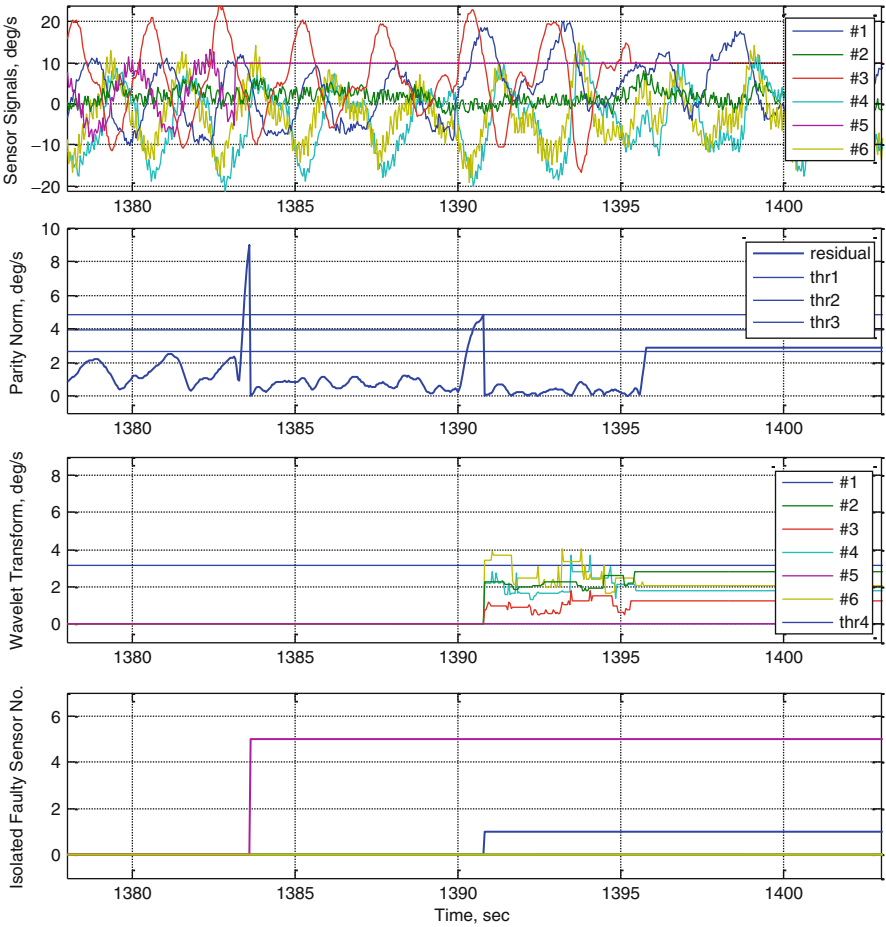


Fig. 47.12 Partially successful result: insensitivity of the wavelet transform to the stuck fault

the discrete wavelet transform monitors signal changes based on 32 previous signal samples. If the change in the faulty signal is small, the parity norm increases slowly. Therefore, the 32 samples of the wavelet transform might not contain the faulty signal because the fault signal is detected by the parity norm. Figure 47.13 shows one of the partially successful FDI results from failure scenario 6. In this scenario, Sensor 2 experienced an 8 deg/s bias error, Sensor 4 stuck at 10 deg/s, and Sensor 6 experienced an 8 deg/s bias error. As shown in the third plot, the wavelet transformation of the signal from Sensor 6 exceeded the threshold due to the third bias fault at 945.9 s. The third fault could not therefore be detected because the fault detection function in the second plot increased slowly and did not guarantee the existence of the faulty sensor. This problem can be easily overcome by considering the previous history of the wavelet-transformed signal.

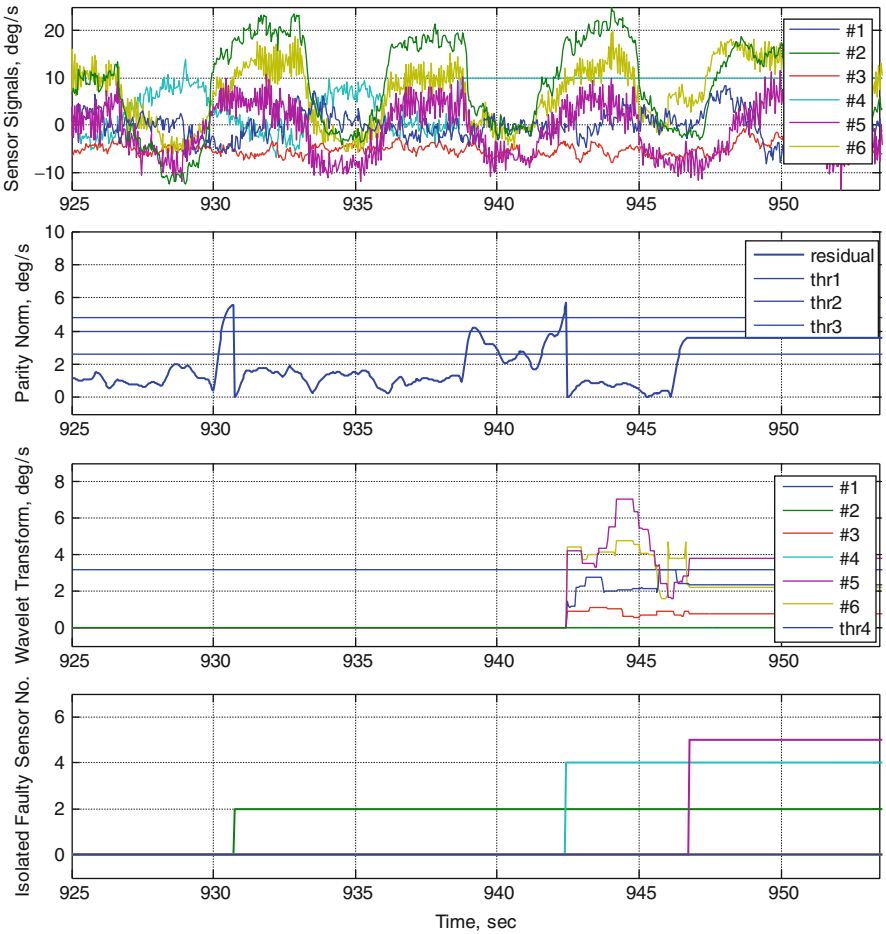


Fig. 47.13 Partially successful result: computational delay in the fault detection function

47.4.3.4 False Alarm Due to the Aircraft Motion

The proposed fault diagnosis method is theoretically independent of aircraft motion or trajectory. Four maneuvers were performed to verify that the aircraft motion does not affect fault diagnosis performance: level flight, bank turn, climb/descent, and oscillating roll motion. As summarized in Table 47.3, the fault diagnosis result was successful regardless of aircraft maneuver when FDI parameter k_i was large. However, false alarms occurred as the FDI parameter decreased because the fault diagnosis became more sensitive to sensor noise or signal change. Even though aircraft states and actuator inputs do not affect the parity vector, a sudden maneuver does affect the time delay between the present aircraft state and the measured states in some inertial sensors. One of the fast maneuvers, an oscillating roll, was performed in flight scenario 7. If the response times of the inertial sensors are not

synchronized, different responses to a fast maneuver might break the orthonormal properties of parity relation and cause a false alarm. Therefore, the redundant secondary IMU should be chosen to have similar response characteristics to those of the primary IMU.

47.5 Conclusion

A model-free hybrid FDI scheme was proposed to provide system tolerance to the occurrence of multiple successive faults in the inertial sensors during flight. The hybrid FDI scheme combines the parity space method with the in-lane monitoring method based on the discrete wavelet transform. This scheme was successfully verified through the hardware-in-the-loop simulation and flight experiments. The skew-configured inertial sensor system was developed with the primary IMU and the redundant secondary IMU. In addition, three successive fault signals were generated virtually with the fault signal generation board during the real flight experiments. The first and the second faults in arbitrary inertial sensors were detected and isolated by the parity space method, and the third sensor fault was detected by the parity space method and isolated by the in-lane monitoring method based on the discrete wavelet transform. The proposed fault diagnosis scheme is applicable to the robust navigation and control of robots, unmanned air/ground vehicles, and satellites.

Acknowledgments The content of this chapter is based on the following journal paper: Yoon et al. (2011). This work was supported by the Korea Aerospace Research Institute under contract 0498-20060013 and was also partially supported by the Smart UAV Development Program, one of the twenty-first Century R&D Programs funded by the Ministry of Knowledge Economy of Korea.

References

- A. Bogges, F.J. Narcowich, *A First Course in Wavelets with Fourier Analysis* (Prentice Hall, Upper Saddle River, 2001)
- J. Chen, R.J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems* (Kluwer Academic, Boston, 1999)
- E.Y. Chow, A.S. Willsky, Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. Autom. Control* **29**(7), 603–614 (1984)
- I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992)
- F.E. DeAngelis, A threshold redundancy management algorithm for skewed sensor arrays, in *Proceedings of IEEE/AIAA/NASA 9th Digital Avionics Systems Conference*, Virginia Beach (1990)
- X. Ding, L. Guo, T. Jeinsch, A characterization of parity space and its application to robust fault detection. *IEEE Trans. Autom. Control* **44**(2), 337–343 (1999)
- D. Dohono, I. Johnstone, Ideal spatial adaptation via wavelet shrinkage. *Biometrika* **81**(3), 425–455 (1994)

- J. Gertler, Fault detection and isolation using parity relations. *Control Eng. Pract.* **5**(5), 653–661 (1997)
- J. Gertler, *Fault Detection and Diagnosis in Engineering Systems* (Marcel Dekker, New York, 1998)
- J.P. Gilmore, R.A. McKern, A redundant strapdown inertial systems mechanization-SIRU, in *Proceedings of the AIAA Guidance, Control, and Flight Mechanics Conference*, Santa Barbara, CA (1970)
- I. Hwang, S. Kim, Y. Kim, C.E. Seah, A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Control Syst. Technol.* **18**(3), 636–653 (2010)
- R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance* (Springer, New York, 2006)
- R. Isermann, P. Balle, Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Eng. Pract.* **5**(5), 709–719 (1997)
- J. Jankovitz, J. Perdsock, Laboratory and flight evaluation of the integrated inertial sensor assembly (IISA), in *Proceedings of IEEE Position, Location and Navigation Symposium*, Orlando, FL (1988)
- J.L. Junkins, Y. Kim, *Introduction to Dynamic and Control of Flexible Structures* (American Institute of Aeronautics and Astronautics, Washington, DC, 1993)
- S. Kim, Y. Kim, C.G. Park, I. Jung, Hybrid fault detection and isolation techniques for aircraft inertial measurement sensors, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Providence, RI (2004)
- S. Kim, Y. Kim, C.G. Park, Failure diagnosis of skew-configured aircraft inertial sensors using wavelet decomposition. *IET Control Theory Appl.* **1**(5), 1390–1397 (2007)
- U.K. Krogmann, Failure management in spatio-temporal redundant, integrated navigation and flight control reference-systems, in *Proceedings of IEEE Position, Location and Navigation Symposium*, Las Vegas, NV (1990)
- X. Li, Discrete wavelet transform for tool breakage monitoring. *Int. J. Mach. Tool Manuf.* **39**(12), 1935–1944 (1999)
- S. Mallat, Theory of multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(7), 674–693 (1989)
- S. Mallat, *A Wavelet Tour of Signal Processing* (Academic, San Diego, 1999)
- C.R. McClary, J.R. Walborn, Fault-tolerant air data inertial reference system development results, in *Proceedings of IEEE Position, Location and Navigation Symposium*, Las Vegas, NV (1994)
- S. Park, Hybrid fault detection and isolation method for TDOF inertial sensors using hardware and analytic redundancy, M.S. Thesis, Department of Mechanical and Aerospace Engineering, Seoul National University, Republic of Korea, 2004
- W. Park, E. Kim, K. Seong, Y. Kim, A study on the parameter estimation of DURUMI-II for the fixed right elevator using flight test data. *J. Mech. Sci. Technol.* **20**(8), 1224–1231 (2006)
- R.J. Patton, Robustness in model-based fault diagnosis: the 1995 situation. *Ann. Rev. Control* **21**, 103–123 (1997)
- R.J. Patton, J. Chen, Review of parity space approaches to fault diagnosis for aerospace systems. *J. Guid. Control Dyn.* **17**(2), 278–285 (1994)
- R.J. Patton, P.M. Frank, R.N. Clark, *Fault Diagnosis in Dynamic Systems: Theory and Application* (Prentice Hall, Englewood Cliffs, 1989)
- R.J. Patton, P.M. Frank, R.N. Clark, *Issues of Fault Diagnosis for Dynamic Systems* (Springer, London, 2000)
- J.E. Potter, J.C. Deckert, Minimax failure detection and identification in redundant gyro and accelerometer systems. *J. Spacecr.* **10**(4), 236–243 (1972)
- J.E. Potter, M.C. Suman, Thresholdless redundancy management with arrays of skewed instruments, in *Integrity in Electronic Flight Control Systems*. Agardograph-224 (NATO, Neuilly sur Seine, 1977), pp. 11–25

- A. Ray, M. Desai, A redundancy management procedure for fault detection and isolation. *J. Dyn. Syst. Meas. Control* **106**(2), 149–156 (1984)
- S. Santoso, E.J. Powers, W.M. Grady, P. Hofmann, Power quality assessment via wavelet transform analysis. *IEEE Trans. Power Deliv.* **11**(2), 924–930 (1996)
- J.J. Sudano, J.R. Preisig, J. Pokotylo, Improved fault detection using a selected grouping of parity equations for advanced flight control systems, in *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference*, Dayton, OH (1988)
- D.H. Titterton, J.L. Weston, *Strapdown Inertial Navigation Technology* (Peter Peregrinus, Lavenham, 1997)
- S. Yoon, S. Kim, J. Bae, Y. Kim, E. Kim, Experimental evaluation of fault diagnosis in a skew-configured UAV sensor system. *Control Eng. Pract.* **19**(2), 158–173 (2011)
- Q. Zhang, Y.A. Yan, Wavelet-based approach to abrupt fault detection and diagnosis of sensors. *IEEE Trans. Instrum. Meas.* **50**(5), 1389–1393 (2001)

Section X

UAV Modeling, Simulation, Estimation and Identification

Lora Weiss

Kimon P. Valavanis and George J. Vachtsevanos

Modeling and simulation techniques are essential and crucial in the design, operation, and performance assessment of UAVs. Together with estimation and identification, they are integral components of the overall UAV design process. They afford the opportunity to develop, test, and evaluate sensing, control, and cooperative control algorithms, among others. Software-in-the-loop and hardware-in-the-loop simulations are typical prerequisites to flight testing. UAV Modeling, Simulation, Estimation and Identification presents a treatment of these topics from first principles to mechanisms that lead to UAV qualification and certification.

► **Flight Dynamics Modeling of Coaxial Rotorcraft UAVs** by Wang, Cui, Chen, and Lee aims at presenting a thorough approach related to systematic modeling of UAV flight dynamics by talking about the UAV model formulation, parameter identification, and model verification. The presented methodology of UAV model formulation and parameter identification is based on modeling two kinds of coaxial helicopters. This is a challenging problem as modeling of coaxial helicopters, despite the common governing kinematic and dynamic principles, deserves special attention due to their distinctive mechanical structure and aerodynamics behavior. The modeling procedures and parameter identification technique presented in this chapter may also serve as a guideline for modeling other types of aerial vehicles.

► **Modeling of a Micro UAV with Slung Payload** by Feng, Rabbath, and Su derives a mathematical and simulation model of a micro UAV carrying a payload.

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

When the load is slung underneath a UAV by cable, the flight dynamics of the UAV are altered, which makes the stability of the UAV disturbed. The unstable oscillation degrades UAV performance, and the accurate placement of the load will be affected. Unlike external disturbances, the negative effects are related to the characteristics of the UAV and the payload. In order for the UAV to be able to adopt the change of the system dynamics and reduce the effects caused by the swing of the load, one modeling method of a micro UAV with single slung payload is addressed. The slung payload is treated as a pendulum-like mass point, and the coupling factors between the UAV and the payload are considered. The derived model may be used to estimate the negative effects acting on the UAV, and it may also be used to estimate the trajectory of the load, which allows for improving the accuracy of placement of the loads.

► **Command and Control of Autonomous Unmanned Vehicles** by Scheidt describes a series of experiments, including 2011 experiments in which 16 fully autonomous unmanned vehicles, including 9 unmanned air vehicles, were used to simultaneously support mounted, dismounted, and maritime users. During these experiments users provided abstract mission-level ISR needs to the “vehicle cloud.” These needs were interpreted by the vehicles, which self-organized and efficiently achieved the user’s objectives. The starting point is using information theory to examine UAV command and control (C2). The information theoretic analysis provides a justification and use cases for *autonomous* UAVs. An autonomous unmanned vehicle system “Organic Persistent Intelligence Surveillance and Reconnaissance” (OPISR) is introduced. OPISR is an autonomous unmanned vehicle system that combines the immediate response to tactical ISR needs provided by organic assets with the time-on-station, minimal logistics provided by persistent unmanned systems. OPISR autonomous vehicles collectively interpret real-time tactical intelligence surveillance and reconnaissance (ISR) objectives submitted by any number of disadvantaged users, gather the required ISR data, and return the needed intelligence directly to the affected user. OPISR is an ad hoc, decentralized system that requires no central base or authority and is capable of functioning in communications-denied environment.

Fei Wang, Jinqiang Cui, Ben M. Chen, and Tong H. Lee

Contents

| | | |
|--------|---|------|
| 49.1 | Introduction | 1218 |
| 49.2 | Working Principle of Coaxial Helicopters | 1220 |
| 49.3 | Model Formulation | 1222 |
| 49.3.1 | Model Overview | 1222 |
| 49.3.2 | Kinematics and Rigid-Body Dynamics | 1224 |
| 49.3.3 | Force and Torque Composition | 1227 |
| 49.3.4 | Force and Torque Formulation of Fixed-Pitch Coaxial Helicopter | 1228 |
| 49.3.5 | Force and Torque Formulation of Variable-Pitch Coaxial Helicopter | 1234 |
| 49.4 | Model-Based Parameter Identification | 1239 |
| 49.4.1 | Direct Measurement | 1239 |
| 49.4.2 | Test Bench Experiment | 1239 |
| 49.4.3 | Flight Test | 1246 |
| 49.5 | Model Verification | 1252 |
| 49.6 | Conclusions | 1255 |
| | References | 1255 |

Abstract

In many unmanned aerial vehicle (UAVs)-related engineering projects, flight dynamics modeling of the controlled platform usually forms the foundation of the whole project development. An accurate mathematical model of the controlled UAV not only makes high-performance model-based control law design possible but also provides insights into the mechanical design of the aerial platform so that radical improvements can be made at the beginning of the development. However, the topic of flight dynamics modeling is somehow not paid enough

F. Wang (✉) • J. Cui • B.M. Chen • T.H. Lee

Control and Simulation Lab, Department of Electrical and Computer Engineering, National University of Singapore, Singapore

e-mail: wangfei@nus.edu.sg; jinqiang@nus.edu.sg; bmchen@nus.edu.sg; eleleeth@nus.edu.sg

attention to across the general engineering audience. This chapter aims to disseminate the knowledge of systematic modeling of UAV flight dynamics by talking about UAV model formulation, parameter identification, and model verification. The methodology of UAV model formulation and parameter identification based on the case study of two kinds of coaxial helicopters will be explained. Modeling of coaxial helicopters, despite the common governing kinematic and dynamic principles, deserves special attention due to their distinctive mechanical structure and aerodynamics behavior. The modeling procedures and parameter identification process presented in this chapter also serve as a guideline for modeling other types of aerial vehicles.

49.1 Introduction

In recent years, unmanned aerial vehicles (UAVs) have been more actively involved in military and civil operations. Possible UAV applications include reconnaissance and intelligence gathering, forest patrol, coastline monitoring, and search and rescue (Tsach et al. 2010). In all kinds of UAV development, the flight dynamics modeling of the controlled platform always forms the cornerstone of the whole project. UAV modeling not only provides an accurate mathematical model so that advanced model-based control law design techniques can be used but also provides insights into the mechanical design of the aerial platform itself. Moreover, in contrast to those conventional manned aerial vehicles, UAV platforms are normally custom-made or largely modified from off-the-shelf products to cope with the payload requirement and the mounting geometry of the onboard avionics. UAV platforms may possess different working principles, ranging from fixed-wing, rotorcraft, flapping-wing to mono-copters like the Samurai (Rosen and Seter 1991). To model some of the special types of aerial platforms, the well-established work for the conventional aerial platforms in literature cannot be applied directly. Hence, it is especially useful to let UAV developers understand how the models are derived and how to identify unknown parameters embedded within the models.

Many works related to flight dynamics modeling and model-based parameter identification of aerial vehicles are conducted in literature. Some of them are done for the ultimate purpose of UAV development. In Heffley and Mnich (1988), a mathematical model for the conventional single-rotor helicopter with adequate complexity was derived, and the report highlighted the formulation of the main rotor thrust generation. In Johnson (1994), a fairly comprehensive and detailed coverage of helicopter working theory and design considerations were provided in aspects including helicopter vertical flight, forward flight, mathematics of rotating systems, rotary wing dynamics and aerodynamics, aeroelasticity, and stability and control. Based on the above works, Cai et al. (2012) obtained a comprehensive nonlinear model of a miniature single-rotor helicopter. This work has also later been extended in a book (Cai et al. 2011) with other UAV-related topics like UAV construction, software development, and controller design.

In more recent years, the development of UAV platforms has entered the era of miniature, microscale, and even nanoscale. The used-to-be efficient conventional fixed-wing or single-rotor helicopter structure may not be an optimum design anymore. The reduced form factor has also resulted in more innovative and unconventional aerodynamic designs, and for these new types of platforms, quite a few new modeling works have been published. In Nonami et al. (2010), the mathematical model of a miniature coaxial helicopter is derived in the transfer function form. The identified linear model is used for optimal controller design. In Bermes (2010), the design and dynamic modeling and the simulation of an autonomous coaxial micro helicopter platform (muFly) are investigated. A modular dynamic model is developed, which incorporates the active and passive flapping characteristics of a hingeless rotor system, the stabilizer bar dynamics, roll-pitch steering by swashplate or displacing center of mass, etc.

Coaxial helicopter is an attractive UAV platform due to its small dimension, high thrust-to-weight ratio, and aerodynamic symmetry. If a coaxial helicopter and a single-rotor helicopter are of the same weight, the size of the coaxial helicopter can be 35–40 % smaller. In addition, the aerodynamic symmetry of coaxial helicopters successfully gets rid of the yaw moment and side forces commonly seen in single-rotor helicopters. Thus, the coaxial helicopter is much more effective in fast forward flights. These advantages make the coaxial helicopter ideal for UAV operation in confined environments such as indoor and cluttered outdoor. There are two types of coaxial helicopters. One has rotor blades with a fixed collective pitch, while the other is with variable collective pitch. In the following content of this chapter, they will be called the fixed-pitch coaxial and the variable-pitch coaxial in short. The dynamic modeling of these two kinds of coaxial helicopters will form the main discussion in this chapter.

A major difference of modeling between the coaxial helicopter and the conventional single-rotor helicopter is the pair of concentric rotors, in which each of them rotates with the induced velocity affected by the other. Such relationship is described in Colin (1997). Detailed studies of the wake dynamics of the two rotors were also documented in Kim and Brown (2006), Rand and Khromov (2010), and Lim et al. (2009). Another specialty of coaxial helicopter is the stabilizer bar attached to the top rotor hub, which passively stabilizes the helicopter. It, however, causes strong influences to the rotor dynamics especially to the fixed-pitch coaxial configuration as the upper rotor is not linked to any servo. As a result, the cyclic pitch control of the upper rotor is solely induced by the stabilizer bar. The stabilizer bar dynamics is commonly modeled as a first-order lag system (Mukherjee and Waslander 2011). In the works shown in Mukherjee and Waslander (2011) and Schafrroth et al. (2010), the tip-path-plane (TPP) dynamics is separated into the upper and lower portion, where only the lower TPP is controlled by the servo inputs.

In a few recent works on the modeling of miniature coaxial helicopter, although fairly complete nonlinear or linear models are obtained, the works lack intuitive explanation of the model formulation. Moreover, their methods of parameter identification are not comprehensive enough. For example, in Neamtu et al. (2010),

the helicopter dynamics were treated as a black box, while the whole system is vaguely identified using the CIPHER (Comprehensive Identification from Frequency Responses) toolkit. To complement the existing work, this book chapter presents the detailed derivation of the nonlinear model for both the fixed-pitch and the variable-pitch coaxial helicopters.

The content of this chapter is organized as follows: Sect. 49.2 briefly gives the working principles of both the fixed-pitch and the variable-pitch coaxial helicopters. Next, Sect. 49.3 will comprehensively formulate the coaxial flight dynamics system, starting from kinematics, rigid-body dynamics, and then force/torque composition and generation. Some parts of the model formulation can be shared by both types of coaxial helicopters, while the differences in rotor thrust generation and rotor flapping dynamics will be separately explained. Model-based parameter identification will be introduced in Sect. 49.4 with the fixed-pitch coaxial as an example. Last but not least, Sect. 49.5 will verify the derived model of the fixed-pitch coaxial case and proves the fidelity of the overall modeling methodology.

49.2 Working Principle of Coaxial Helicopters

Before any rigorous derivation of the flight dynamic models of the aforementioned two types of coaxial helicopters, the general working principles of the two will first be introduced so that it will be easier for readers to understand the later detailed model formulation. First of all, both platforms have a pair of contrarotating rotors (upper and lower) to provide the fundamental lift force for the overall platform. This is indeed where the term “coaxial” comes from.

For the fixed-pitch coaxial helicopter, as shown in Fig. 49.1, the collective pitch of the rotor blades cannot be changed. Hence, the heave and yaw motion of the platform can only be controlled by varying the rotational speed of the rotors, which are linked to two separate motors with step-down gears. In general, the summation of the motor speeds determines the helicopter vertical motion, while the difference of the two determines the yaw motion. Rolling and pitching are accomplished by introducing a cyclic pitch to the lower rotor via a dual-servo-controlled swashplate. In this way, a tilted flapping of the rotor blades can be induced, and the generated rotor thrust becomes non-vertical. In the case of the platform shown in Fig. 49.1, which is called the Esky Big Lama, the cyclic pitch of the upper rotor is not actively controlled. Instead, the rotor hub tethers together with a stabilizer bar. As the stabilizer bar is purposely constructed with relatively high moment of inertia, when the helicopter fuselage suddenly tilts, the stabilizer bar tends to remain rotating at the original level plane. This introduces a cyclic pitch to the upper rotor, and it is purposely designed in a way that the thrust tilting resulted from this cyclic pitch counters the instantaneous motion of the helicopter. Therefore, this kind of fixed-pitch coaxial helicopters is inherently stable with regard to their attitude angles.

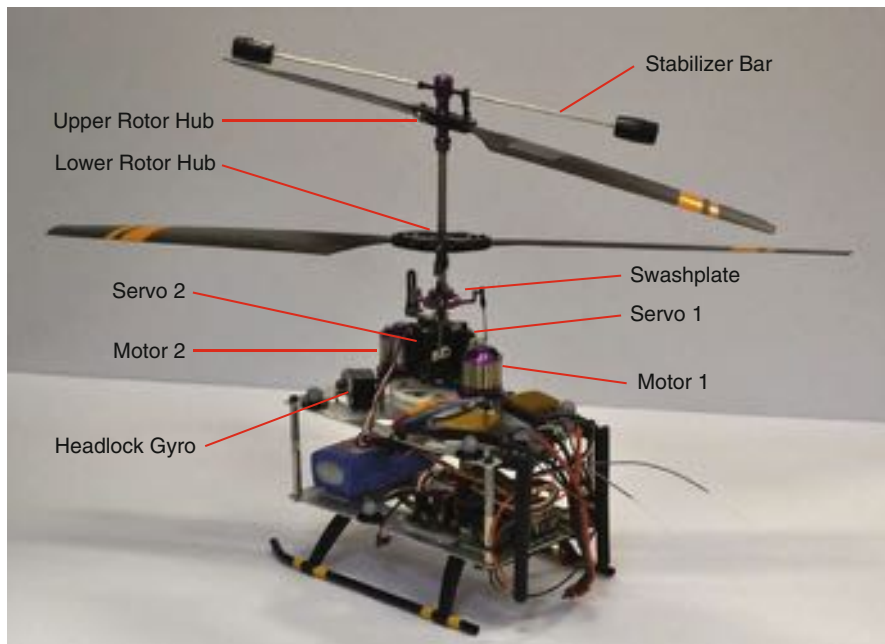


Fig. 49.1 Description of the fixed-pitch coaxial helicopter

On the other hand, the variable-pitch coaxial helicopter, as shown in Fig. 49.2, is a mini coaxial helicopter customized according to the design of full-scale coaxial helicopter from the Kamov design bureau. Its rotor head is equipped with integrated hinges and shock-resistant dampers. This helicopter also consists of two contrarotating rotors. However, unlike the fixed-pitch coaxial, the rotational speeds of the variable-pitch coaxial helicopter's rotors are maintained at the same constant speed in normal flight conditions. The dynamic motion of the helicopter is achieved by actively changing the pitch angles of both upper and lower rotors via the upper swashplate and the lower swashplate, respectively. The pitch angles of both rotors are constituted by collective pitch and cyclic pitch, which are mixed controlled by three servos linked to the lower swashplate. The two swashplates are always parallel to each other since they are circumferentially connected by three rigid linkages. The upper rotor is attached with a Bell-Hiller stabilizer bar which introduces damping to the rotor's cyclic pitch. Collective and cyclic inputs from servos are transferred to the lower swashplate and also induced to the top swashplate, resulting in the dynamic movement of the helicopter in heave direction or pitch-roll direction. The yaw direction control is realized by changing the collective pitch angle of the lower rotor. For this particular platform, the upper rotor and lower rotor are driven by the same brushless direct current (DC) electric motor with the same gear ratio. Hence, the rotational speeds of the upper rotor and the lower rotor are always the same.

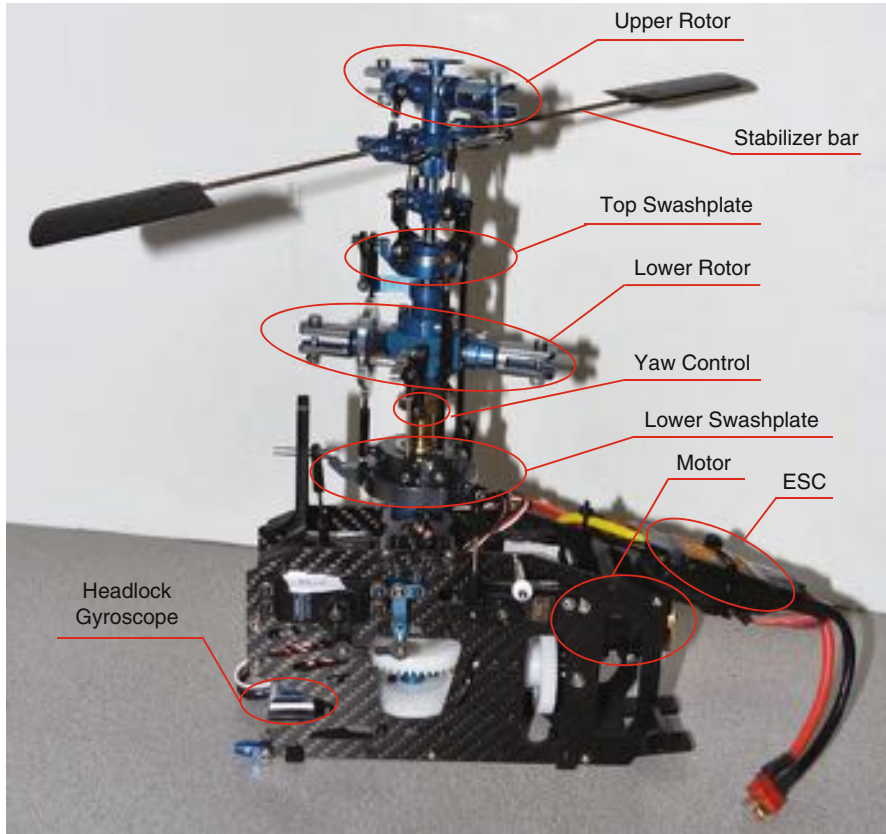


Fig. 49.2 Description of the variable-pitch coaxial helicopter

49.3 Model Formulation

49.3.1 Model Overview

The mathematical model of any continuous physical dynamic system can be expressed in the following compact form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}), \quad (49.1)$$

where \mathbf{x} is the state vector, \mathbf{u} is the input vector, and \mathbf{w} represents external disturbances. For the case of the fixed-pitch coaxial helicopter system, the system state and input vectors can be defined as

$$\mathbf{x} = (x \ y \ z \ u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r \ a_{\text{up}} \ b_{\text{up}} \ a_{\text{dw}} \ b_{\text{dw}} \ \Omega_{\text{up}} \ \Omega_{\text{dw}} \ r_{\text{fb}})^T, \quad (49.2)$$

Table 49.1 Physical meaning of state variables

| Symbol | Physical meaning | Unit |
|----------------------|--|--------------------------------|
| x | | |
| y | \mathbf{p} , position in the ground frame | m |
| z | | |
| u | | |
| v | \mathbf{v}_b , linear velocity in the body frame | m/s |
| w | | |
| ϕ | Roll | rad |
| θ | Pitch | |
| ψ | Yaw | |
| } attitude angle | | |
| p | | |
| q | $\boldsymbol{\omega}$, angular velocity in body frame | rad/s |
| r | | |
| a_{up} | Longitudinal | flapping angle of upper blades |
| b_{up} | Lateral | |
| a_{dw} | Longitudinal | flapping angle of lower blades |
| b_{dw} | Lateral | |
| Ω_{up} | Rotational speed of the upper rotor | |
| Ω_{dw} | Rotational speed of the lower rotor | |
| r_{fb} | Controller state of yaw stability augmentation | NA |

$$\mathbf{u} = (\delta_{\text{ail}} \delta_{\text{ele}} \delta_{\text{thr}} \delta_{\text{rud}})^T, \quad (49.3)$$

$$\mathbf{w} = (\omega_u \omega_v \omega_w)^T. \quad (49.4)$$

For the case of the variable-pitch coaxial helicopter system, the system state and input vectors can be defined as

$$\mathbf{x} = (x \ y \ z \ u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r \ a_{\text{up}} \ b_{\text{up}} \ a_{\text{dw}} \ b_{\text{dw}} \ r_{\text{fb}})^T, \quad (49.5)$$

$$\mathbf{u} = (\delta_{\text{ail}} \delta_{\text{ele}} \delta_{\text{thr}} \delta_{\text{rud}})^T, \quad (49.6)$$

$$\mathbf{w} = (\omega_u \omega_v \omega_w)^T. \quad (49.7)$$

The physical meanings of the state, input, and disturbance variables are listed in Tables 49.1–49.3. It should be noted that the fixed-pitch coaxial helicopter has two additional state variables, Ω_{up} and Ω_{dw} , because its rotor rotational speeds keep changing during normal flights and its motor dynamics is not fast enough to be neglected. On the other hand, the rotors of the variable-pitch coaxial platform always rotate at the same speed. Thus, no dynamics need to be considered.

With regard to the input definition, the conventional radio-controlled (RC) joystick signals, aileron (δ_{ail}), elevator (δ_{ele}), throttle (δ_{thr}), and rudder (δ_{rud}), normalized to $[-1, 1]$ with respect to their corresponding minimum and maximum values, are chosen as the primary system inputs. This kind of input definition makes sure that the modeling methodology is also applicable to other types of

Table 49.2 Physical meaning of input variables

| Variables | Physical meaning | Range |
|----------------------|--|-----------|
| δ_{ail} | Control deflection for lateral cyclic pitch | $[-1, 1]$ |
| δ_{ele} | Control deflection for longitudinal cyclic pitch | $[-1, 1]$ |
| δ_{thr} | Control deflection for collective pitch | $[-1, 1]$ |
| δ_{rud} | Control deflection for collective pitch of tail rotor | $[-1, 1]$ |
| $\bar{\delta}_{rud}$ | Control deflection for yaw-stability-augmentation controller | $[-1, 1]$ |

Table 49.3 Physical meaning of wind disturbance variables

| Variables | Physical meaning |
|------------|---|
| ω_u | Wind velocity in the helicopter x -axis |
| ω_v | Wind velocity in the helicopter y -axis |
| ω_w | Wind velocity in the helicopter z -axis |

aerial platforms. Usually, the helicopter attitude angles (ϕ, θ), heading (ψ), and 3D position (x, y, z) are chosen to be the ultimate controlled outputs as they can comprehensively define the state of the helicopter in the full 6 degree-of-freedom (6DoF) space.

Furthermore, to derive a mathematical model of a complex system, it is preferable to modularize the overall system into subsystems so that the divide-and-conquer strategy can be used. Here, an overview of the two model structures is shown in Figs. 49.3 and 49.4. From the inputs to the state variables, there are numerous blocks representing all the subsystems involved. The two models share quite a few similarities but preserve their own distinctive features as well. On the one hand, the input and output definitions are unified for both, and the same kinematics and dynamics blocks can be used to relate the body-frame forces and moments to the helicopter 6DoF motion. On the other hand, the two models are different in the mechanism in generating the individual forces and moments. This results in two different sets of intermediate model blocks. In the following sections, mechanisms in all these blocks will be explained in detail. The similar blocks between these two types of platforms will be discussed together, while the different blocks will be discussed separately.

49.3.2 Kinematics and Rigid-Body Dynamics

As a common practice of aeronautic analysis, model formulation of a flying vehicle normally assumes that the target platform is a rigid body. Thus, it follows the universal 6DoF kinematics equations and the Newton-Euler dynamics equations. Two main coordinate frames are generally involved to link the equations. One is the north-east-down (NED) frame, and the other is the helicopter body frame. While the NED frame is stationary with respect to a static observer on the ground, the body

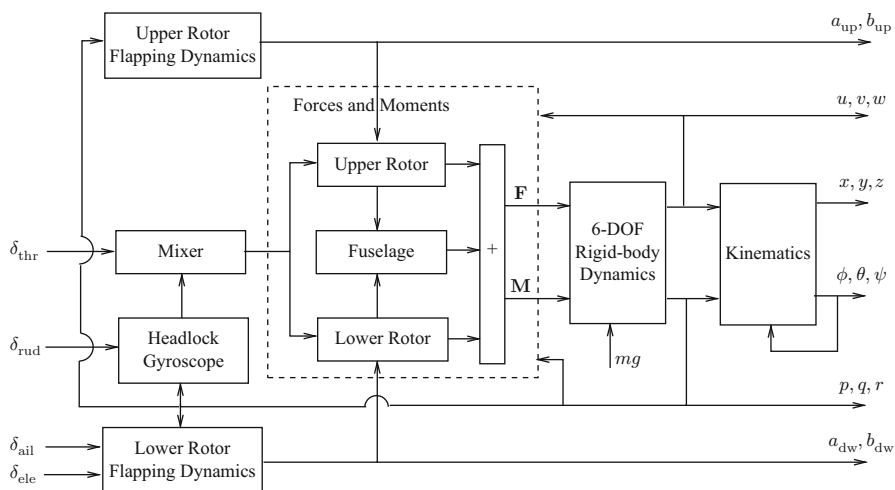


Fig. 49.3 Model structure of fixed-pitch coaxial helicopter

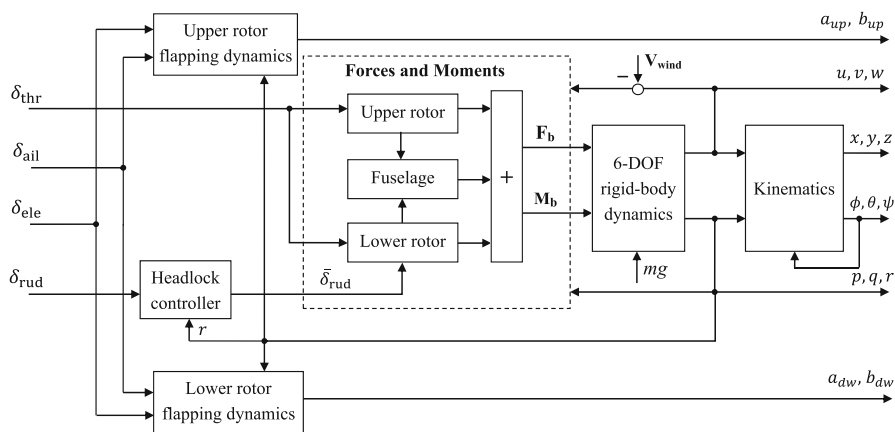


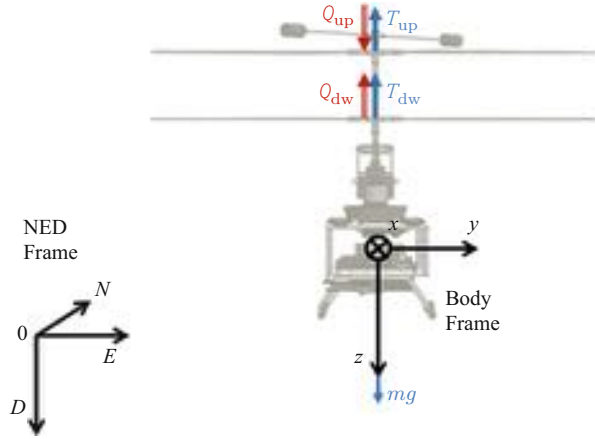
Fig. 49.4 Model structure of variable-pitch coaxial helicopter

frame is placed at the center of gravity (CG) of the coaxial helicopter, where its origin and orientation move together with the helicopter fuselage (see Fig. 49.5). It is worth noting that the listed formulas to describe the kinematics and dynamics of the coaxial helicopter are indeed universal to all other rigid-body vehicles.

The relationship between the helicopter NED-frame position and its body-frame velocity is determined by the following navigation equation:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \quad (49.8)$$

Fig. 49.5 Coordinate frames and various forces and torques



where x, y, z are the NED-frame position components of the helicopter and u, v, w are the body-frame velocity components. ϕ, θ, ψ are the conventional roll, pitch, yaw angles of the helicopter fuselage and s_*, c_* denote $\sin(*), \cos(*),$ respectively. It is also critical to point out that the Euler angle derivatives, $\dot{\phi}, \dot{\theta}, \dot{\psi},$ are not orthogonal to each other. They are related to the body-frame angular rates, $p, q, r,$ by the following equation:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} 1 & s_\phi s_\theta / c_\theta & c_\phi s_\theta / c_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}. \quad (49.9)$$

Note that the above equation has singularity at $\theta = 90^\circ$. If full-envelope flight is required, a quaternion representation is recommended. However, normal maneuvering of a coaxial helicopter will not hit such an extreme condition. Thus, it is still adequate to use this relatively simple equation.

Next, by treating the whole coaxial platform as a rigid mass, the 6DoF dynamics can be described by the following Newton-Euler equations:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \frac{1}{m} \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \quad (49.10)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \mathbf{J}^{-1} \left\{ \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \mathbf{J} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \right\}, \quad (49.11)$$

where F_x, F_y, F_z are projections of the net force, $\mathbf{F},$ onto the helicopter body-frame $x-, y-, z$ -axis and M_x, M_y, M_z are projections of the net torque, $\mathbf{M},$ onto the body-frame $x-, y-, z$ -axis. The compositions of \mathbf{F} and \mathbf{M} come from

various parts of the coaxial helicopter. For the fixed-pitch and variable-pitch coaxial helicopters, although their force and torque compositions are more or less the same (explained in Sect. 49.3.3), the individual force and torque generation will have different formulations because of their different working principles (explained in Sects. 49.3.4 and 49.3.5, respectively). Till now, the unknown parameters that need to be identified are m , the total mass of the platform, and \mathbf{J} , the moment of inertia of the platform, which is in the form of

$$\mathbf{J} = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix}.$$

Since the coaxial helicopters are normally designed to be symmetric in both longitudinal and lateral directions, J_{xy} , J_{xz} , J_{yz} are extremely small and can be assumed to be zero. The identification method of J_{xx} , J_{yy} , J_{zz} will be explained in Sect. 49.4, which talks about various approaches in the problem of model-based parameter identification.

49.3.3 Force and Torque Composition

As mentioned in Sect. 49.3.2, forces and torques acting on the coaxial helicopter come from various mechanical parts. First of all, the helicopter weight exerts a force of mg in the NED-frame z -axis. After converting it to the body frame, the vector is shown as the second term on the right-hand side of Eq. 49.12.

Next, when the rotor blades spin, they generate thrusts, \mathbf{T}_i ($i = \text{up, dw}$), in the direction perpendicular to their respective tip path plane (TPP). When the upper and lower TPPs deviate from their default orientation, the thrust vectors no longer pass through the CG of the helicopter, thus creating rotational torque. The torque vectors caused by the rotor thrusts can be calculated by $\mathbf{l}_{\text{up}} \times \mathbf{T}_{\text{up}}$ and $\mathbf{l}_{\text{dw}} \times \mathbf{T}_{\text{dw}}$, where \mathbf{l}_{up} and \mathbf{l}_{dw} are the displacement vectors from helicopter CG to the upper rotor hub, and the lower rotor hub respectively. The deviation of the TPP can be described by the longitudinal flapping angle a_i and the lateral flapping angle b_i . The thrust decomposition to the body-frame axes can be approximated by Eq. 49.14. Nonzero a_i and b_i also directly result in flapping torque on the rotor hub. This torque can be simplified as the second term on the right-hand side of Eq. 49.13, where K_β is called the effective spring constant, and it has the same value for both the upper and lower rotors provided they are rotating at approximately the same speed.

Furthermore, the rotation of the rotors creates the drag torque, \mathbf{Q}_{up} and \mathbf{Q}_{dw} , around the body-frame z -axis. When the coaxial helicopter hovers without yaw motion, the two torques have the same magnitude, thus canceling each other. Else, if the net drag torque is nonzero, yaw acceleration is generated. In addition, the change of rotational speeds of the rotors also generates the so-called reaction torques on the helicopter body (denoted by $\mathbf{Q}_{\text{r,up}}$ and $\mathbf{Q}_{\text{r,dw}}$). They are described in (Eq. 49.16),

where J_{up} and J_{dw} are the moment of inertia of the upper rotor (with stabilizer bar) and the lower rotor with respect to the rotor shaft. They can be calculated by measuring the mass and dimension of the rotor blades and stabilizer bar and assuming a regular geometric shape.

Lastly, when the helicopter moves in air, its fuselage experiences drag forces, X_{fus} , Y_{fus} , Z_{fus} , due to air resistance. This drag force is usually related to the linear speed and up-front surface area of the aerial vehicle.

Equations (49.12) and (49.13) have summarized all the forces and torques mentioned above, with (49.14)–(49.16) explaining how to evaluate the individual terms:

$$\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = \sum \mathbf{T}_i + mg \begin{pmatrix} -s\theta \\ s_\phi c\theta \\ c_\phi c\theta \end{pmatrix} + \begin{pmatrix} X_{\text{fus}} \\ Y_{\text{fus}} \\ Z_{\text{fus}} \end{pmatrix}, \quad (49.12)$$

$$\begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = \sum \mathbf{l}_i \times \mathbf{T}_i + \sum K_\beta \begin{pmatrix} b_i \\ a_i \\ 0 \end{pmatrix} + \sum \mathbf{Q}_{\text{d},i} + \sum \mathbf{Q}_{\text{r},i}, \quad (49.13)$$

$$\mathbf{l}_i = |\mathbf{l}_i| \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \quad \mathbf{T}_i = |\mathbf{T}_i| \begin{pmatrix} -\sin a_i \\ \sin b_i \\ -\cos a_i \cos b_i \end{pmatrix}, \quad (49.14)$$

$$\mathbf{Q}_{\text{d},\text{up}} = |\mathbf{Q}_{\text{d},\text{up}}| \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{Q}_{\text{d},\text{dw}} = |\mathbf{Q}_{\text{d},\text{dw}}| \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \quad (49.15)$$

$$\mathbf{Q}_{\text{r},\text{up}} = J_{\text{up}} \dot{\Omega}_{\text{up}} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{Q}_{\text{r},\text{dw}} = J_{\text{dw}} \dot{\Omega}_{\text{dw}} \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}. \quad (49.16)$$

49.3.4 Force and Torque Formulation of Fixed-Pitch Coaxial Helicopter

The generation of individual forces and torques on the fixed-pitch coaxial and the variable-pitch coaxial has quite different formulations. In this section, a full formulation of the fixed-pitch coaxial force and torque generation will be provided. That means, by tracing all the formulas listed in this section, the forces and torques exerted on the fixed-pitch coaxial helicopter can be exhaustively related to the four fundamental inputs in a rigorous way.

49.3.4.1 Thrust and Torque from Rotors

Here, the magnitude of the rotor thrust and drag torque, $|\mathbf{T}_i|$ and $|\mathbf{Q}_{d,i}|$, will first be investigated. According to the aerodynamic actuator disk theory (Bramwell et al. 2001), the magnitude of thrust generated by the rotors can be formulated as follows:

$$|\mathbf{T}_i| = \rho C_{T,i} A (\Omega_i R)^2, \quad (49.17)$$

where ρ is the density of air, $C_{T,i}$ is the lift coefficient, A is the rotor disk area, Ω_i is the rotational speed of the rotor, and R is the rotor blade length. Since this is a fixed-pitch coaxial helicopter, $C_{T,i}$, like the other parameters in (49.17), is constant. The only variable is Ω_i . Hence, the equation can be simplified to

$$|\mathbf{T}_i| = k_{T,i} \Omega_i^2, \quad (49.18)$$

where $k_{T,i}$ is a lumped thrust coefficient that needs to be identified. Similar assumptions and formulation can be applied to the relationship between the drag torque and the rotational speed of the rotors:

$$|\mathbf{Q}_{d,i}| = k_{Q,i} \Omega_i^2. \quad (49.19)$$

49.3.4.2 Rotor Flapping Dynamics

For this specific type of coaxial helicopter, the rotor collective pitch is fixed, while the cyclic pitch can be changed. For the lower rotor, the rotor hub is connected to the aileron and the elevator servos via a swashplate. When the swashplate tilts, it teeters the rotor hub and creates a cyclic pitch on the rotor. For every cycle of rotation, the rotor blade will reach the maximum angle of attack at a particular phase angle at which the lift on the blade is largest. This results in the flapping of the rotor disk. The whole mechanism is a combination of gyroscopic precession and aerodynamic precession. For the case of Esky Big Lama, if one observes the rotor blade in a slow motion, the maximum rotor flapping occurs roughly at 45° lag with respect to the occurrence of maximum angle of attack. This explains why the aileron and elevator servos of the off-the-shelf coaxial platform are connected to the swashplate 45° off the body-frame x -, y -axis. In this way, the aileron servo mainly controls the lateral flapping of the lower rotor, and the elevator servo mainly controls the longitudinal flapping. However, the flapping phase lag is not exactly equal to 45° (slightly larger than 45° from test bench observations) due to mechanical modifications to the original RC platform (original rotor blades have been replaced by stiffer ones for larger payload). This results in non-negligible coupling between the servo inputs and the lower rotor longitudinal and lateral flapping angles. As the lower rotor does not have any additional damping mechanism attached, its flapping process is almost instantaneous. By assuming a first-order dynamics, the time constant can be observed via a high-speed camera. The result turns out to be 0.0375 s (see Fig. 49.6), which is very small as compared to dynamics of other parts of the coaxial helicopter

Fig. 49.6 Step response of servo motion attached to the lower rotor (*top*: $t = 0$; *center*, $t \approx 0.0375$ s; *bottom*, $t = \infty$). This figure looks dim because it was taken by a high-speed camera



and thus can be ignored. Hence, the relationship between servo inputs and lower rotor flapping angles can be formulated in a static way:

$$a_{dw} = A_{a,dw} \delta_{ele} + A_{b,dw} \delta_{ail} - A_q q, \quad (49.20)$$

$$b_{dw} = B_{b,dw} \delta_{ail} + B_{a,dw} \delta_{ele} - B_p p, \quad (49.21)$$

where aileron (δ_{ail}) and elevator (δ_{ele}) are the servo inputs, $A_{a,dw}$ and $B_{b,dw}$ are the on-axis steady-state ratio from servo inputs to flapping angles, and $A_{b,dw}$ and $B_{a,dw}$ are the off-axis (coupling) values. The last term which depends on angular rates, p and q , comes from an effect called rotor damping, which was considered in literature either in linear or quadratic form. Here, a linear form is chosen because of its simplicity.

For the upper rotor system, a stabilizer bar is attached to the rotor hub, so that they teeter together. As the stabilizer bar has large moment of inertia, it tends to remain at its original rotating plane. Hence, at the moment when the helicopter body tilts, the stabilizer bar TPP will remain at the level plane, thus creating a cyclic pitch on the upper rotor which leads to blade flapping. The torque generated by this flapping redresses the rotational motion of the helicopter and significantly stabilizes the whole platform attitude. Similar to the lower rotor system, the stabilizer bar is installed at 45° phase lead to the rotor blade. In this way, the maximum flapping happens at the direction that precisely counters the rotational motion of the helicopter. Again, there is coupling between the longitudinal and lateral channels because the flapping phase lag is not exactly 45° . The following equations describe the above mentioned dynamics:

$$\dot{\phi}_{sb} = \frac{1}{\tau_{sb}} (\phi - \phi_{sb}), \quad (49.22)$$

$$\dot{\theta}_{sb} = \frac{1}{\tau_{sb}} (\theta - \theta_{sb}), \quad (49.23)$$

$$a_{up} = A_{a,up} (\theta_{sb} - \theta) + A_{b,up} (\phi_{sb} - \phi) - A_q q, \quad (49.24)$$

$$b_{up} = B_{b,up} (\phi_{sb} - \phi) + B_{a,up} (\theta_{sb} - \theta) - B_p p, \quad (49.25)$$

where ϕ_{sb} and θ_{sb} are the roll and pitch angles of the stabilizer bar TPP, $A_{a,up}$ and $B_{b,up}$ are the on-axis steady-state ratio from the stabilizer bar teetering angles to the upper rotor flapping angles, and $A_{b,up}$ and $B_{a,up}$ are the off-axis (coupling) values. τ_{sb} is the time constant of approximated first-order flapping dynamics. Again, the same rotor damping effects (terms depending on p and q) are considered for the upper rotor flapping dynamics.

49.3.4.3 Fuselage Drag

When the helicopter fuselage moves in air, it experiences drag force acting on the opposite direction of the motion. For the body-frame horizontal directions, the rotor

downwash is deflected by u and v . In the situation when u (or v) is less than v_i (the induced velocity of air at the lower rotor), the downwash effect needs to be taken into account. Otherwise, the downwash effect is relatively weak and can be ignored. The fuselage in all three directions is considered as a flat plate perpendicular to the helicopter motion; thus, the drag coefficient is approximately unity. As such, the horizontal fuselage drag forces are formulated in a quadratic form:

$$X_{\text{fus}} = -\frac{\rho}{2} S_x u \cdot \max(v_i, |u|), \quad (49.26)$$

$$Y_{\text{fus}} = -\frac{\rho}{2} S_y v \cdot \max(v_i, |v|), \quad (49.27)$$

$$v_i = \sqrt{\frac{|T_{\text{dw}}|}{2\rho\pi R^2}}, \quad (49.28)$$

where S_x and S_y are the effective drag area along the body-frame x - and y -axis, respectively.

For the vertical direction, since the fuselage is constantly exposed to the lower rotor downwash, it is commonly formulated in the following form:

$$Z_{\text{fus}} = -\frac{\rho}{2} S_z (w - v_i) |w - v_i|. \quad (49.29)$$

However, as the lift coefficient test for identifying $k_{T,i}$ in Eq. 49.18 was done with the presence of the fuselage (so the term $\frac{\rho}{2} S_z v_i^2$ has already been taken into account), the above equation needs to be compensated as

$$Z_{\text{fus}} = -\frac{\rho}{2} S_z w \cdot \max(v_i, |w|), \quad (49.30)$$

where S_z is the effective drag area along the body-frame z -axis.

49.3.4.4 Motor Speed Dynamics

Two brushless DC motors are used on the Esky Big Lama coaxial platform. Their rotational speed dynamics follows the well-known differential equation of electro motors:

$$J_{\text{mot}} \dot{\omega} = \frac{k_m U - k_m k_e \omega}{R_{\text{mot}}} - d\omega - M_L, \quad (49.31)$$

where J_{mot} is the motor moment of inertia, k_m and k_e are the mechanical and electrical motor constants, U is the input voltage, R_{mot} is the resistance of the circuit, d is the friction coefficient, and M_L is the external torque acting on the motor shaft. Here, M_L is equal to the rotor drag torque $Q_{d,i}$ appeared in Eq. 49.19. If the helicopter operates at a near-hover condition, everything can be approximated as linear. M_L can be assumed to be a combination of a constant trimming value, M_L^* ,

and another term proportional to extra rotational speed as compared to the trimming speed, Ω^* :

$$M_L = M_L^* + k_L(\Omega - \Omega^*). \quad (49.32)$$

By further considering that the rotational speed of rotor, Ω , and the rotational speed of the motor, ω , are perfectly proportional by the gear ratio, the rotor speed dynamics can be simplified to the following first-order equations:

$$\dot{\Omega}_{\text{up}} = \frac{1}{\tau_{\text{mt}}}(m_{\text{up}}\delta_{\text{up}} + \Omega_{\text{up}}^* - \Omega_{\text{up}}), \quad (49.33)$$

$$\dot{\Omega}_{\text{dw}} = \frac{1}{\tau_{\text{mt}}}(m_{\text{dw}}\delta_{\text{dw}} + \Omega_{\text{dw}}^* - \Omega_{\text{dw}}), \quad (49.34)$$

where Ω_{up}^* and Ω_{dw}^* are the trimming values of the rotor rotational speed at hovering, τ_{mt} is the time constant of the motor speed dynamics, and m_{up} , m_{dw} are the steady-state ratio between the change of rotor speeds and the change of motor inputs.

49.3.4.5 Mixer and Headlock Gyro Dynamics

In order to largely decouple the throttle-heave and the rudder-yaw dynamics, the throttle and rudder signals are passed into a hardware mixer and transformed to dual motor control signals:

$$\delta_{\text{up}} = \delta_{\text{thr}} + \bar{\delta}_{\text{rud}}, \quad (49.35)$$

$$\delta_{\text{dw}} = \delta_{\text{thr}} - \bar{\delta}_{\text{rud}}. \quad (49.36)$$

It can be seen that when the throttle signal δ_{thr} increases, inputs to both motors increase; when the rudder signal $\bar{\delta}_{\text{rud}}$ increases, the input to the motor connected to the upper rotor increases while the input to the motor connected to the lower rotor decreases.

Note that the rudder signal in the above mixer equation is not the original signal δ_{rud} . From δ_{rud} to $\bar{\delta}_{\text{rud}}$, there is a hardware headlock gyro which helps refine the rudder signal and acts as the most inner-loop yaw motion stabilizer. Usually, there is a PI controller embedded inside the headlock gyro, and it can be formulated as follows:

$$\dot{r}_{\text{fb}} = K_a\delta_{\text{rud}} - r, \quad (49.37)$$

$$\bar{\delta}_{\text{rud}} = K_P(K_a\delta_{\text{rud}} - r) + K_I r_{\text{fb}}, \quad (49.38)$$

where r_{fb} is the augmented state variable needed by the integral control. At this point, the full dynamics of a coaxial helicopter have been mathematically formulated, but the model parameters are yet to be identified. In Sect. 49.4, the identification methods will be comprehensively given.

49.3.5 Force and Torque Formulation of Variable-Pitch Coaxial Helicopter

For the case of the variable-pitch coaxial helicopter, the formulation of force and torque generation will not be thoroughly listed as some of the formulas are very much like its fixed-pitch counterpart. For example, the formulas about fuselage drag and the headlock gyro PI controller are more or less the same. As such, only formulations that have significant differences when compared to the fixed-pitch coaxial case will be listed and explained here.

It should also be noted that the main differences compared occur in two areas, namely, the rotor thrust generation and the rotor flapping dynamics. In this section, the formulations about thrust generation and rotor flapping dynamics of the variable-pitch coaxial helicopter will be derived from a more fundamental perspective, thus capturing more aerodynamic details. This also reflects the fact that model formulation of aerial vehicles can be done at different levels of complexity. If the model is to be used for accurate structural analysis and optimization or controller design for aggressive maneuvering, then a more precise and complicated modeling approach should be adopted. However, if the planned flight missions are relatively stable and peaceful, a simplified model formulation is usually more than enough, such as that of the fixed-pitch coaxial case.

Thorough aerodynamics study of coaxial rotor system is itself a big research topic. NASA researcher (Colin 1997) has provided a good survey covering the aerodynamics research worldwide, including America, Russia, Japan, and Germany. Of all the surveyed techniques, blade element momentum theory (BEMT) (Leishman 2006) is a standard method for preliminary rotor analysis before complex high-level analysis, such as free vortex models (FVM) and Lagrangian particle vortex methods (LPVM). In this chapter, the detailed derivation and reasoning of BEMT will not be repeated. Only the main computation procedures are listed for easier reference and understanding.

49.3.5.1 Thrust and Torque from Rotors

The formal analysis of helicopter rotor motion usually starts from its thrust, torque, and power generation. The rotor thrust, torque, and power can be expressed as

$$T = \rho C_T A \Omega^2 R^2, \quad (49.39)$$

$$Q = \rho C_Q A \Omega^2 R^3, \quad (49.40)$$

$$P = \rho C_P A \Omega^3 R^3. \quad (49.41)$$

Here, ρ is the air density, A is the rotor disk area, Ω is the rotor rotational speed, and R is the rotor radius. C_T , C_P , C_Q are the rotor thrust coefficient, power coefficient, and torque coefficient, respectively. It should be noted that the power is related to torque by $P = \Omega Q$. Hence, $C_P = C_Q$ numerically. All the parameters in Eqs. 49.39–49.41 are constant except for the three coefficients C_T , C_P , C_Q .

First of all, the incremental thrust coefficient is defined as

$$dC_T = \frac{1}{2} \sigma C_{l_\alpha} r^2 dr = \frac{\sigma C_{l_\alpha}}{2} (\theta_u r^2 - \lambda r) dr, \quad (49.42)$$

where C_{l_α} is the lift-curve slope of the airfoil section, which can be obtained by checking the official wind tunnel test results if the blade airfoil is standard. σ is the rotor solidity defined as the ratio of the blade area against the rotor disk area. θ_u is the blade pitch distribution on the upper rotor. r is the nondimensional radial distance along the blade. λ is the nondimensional induced velocity which can be expressed in terms of r and λ_∞ as

$$\lambda(r, \lambda_\infty) = \sqrt{\left(\frac{\sigma C_{l_\alpha}}{16F} - \frac{\lambda_\infty}{2}\right)^2 + \frac{\sigma C_{l_\alpha}}{8F} \theta_u r} - \left(\frac{\sigma C_{l_\alpha}}{16F} - \frac{\lambda_\infty}{2}\right), \quad (49.43)$$

where F is the factor to account for the Prandtl tip loss defined as

$$F = \left(\frac{2}{\pi}\right) \cos^{-1}(e^{-f}), \quad (49.44)$$

where f is given in terms of the number of blades N_b and the radial position of the blade element r :

$$f = \frac{N_b}{2} \left(\frac{1-r}{r\phi} \right), \quad (49.45)$$

and ϕ is the induced inflow angle, which equals to $\lambda(r)/r$.

For both the upper and lower rotors, the same principles can be applied. However, as the inner part of the lower rotor operates in the *vena contracta* of the upper rotor, the analysis can be more complicated. From the flow visualization results of Taylor (1950), the author stated that the wake of the upper rotor contracts fully within $0.25R$ below the upper rotor. The ideal wake contraction ratio, r_c , is 0.707, but in practice it is found closer to 0.8. The contracted wake is defined as $A_c = \pi r_c^2 R^2$. The inner area of the lower rotor encounters incoming stream-tubes with velocity $V_\infty + (A/A_c)v_u$ ($V_\infty + 2v_u$ in the ideal case). For beam sections lying inside the upper rotor contraction area, the inflow distribution is given as

$$\lambda(r, \lambda_\infty) = \sqrt{\left(\frac{\sigma C_{l_\alpha}}{16F} - \frac{\lambda_\infty + (A/A_c)\lambda_u}{2}\right)^2 + \frac{\sigma C_{l_\alpha}}{8F} \theta_l r} - \left(\frac{\sigma C_{l_\alpha}}{16F} - \frac{\lambda_\infty + (A/A_c)\lambda_u}{2}\right), \quad (49.46)$$

where θ_l is the blade pitch distribution on the lower rotor. For points outside the contraction area, the inflow distribution is given as

$$\lambda(r, \lambda_\infty) = \sqrt{\left(\frac{\sigma C_{l_\alpha}}{16F} - \frac{\lambda_\infty}{2}\right)^2 + \frac{\sigma C_{l_\alpha}}{8F} \theta_l r} - \left(\frac{\sigma C_{l_\alpha}}{16F} - \frac{\lambda_\infty}{2}\right). \quad (49.47)$$

When the inflow velocity distribution is obtained, the total thrust lift coefficient could be found by integrating Eq. 49.42 as

$$C_T = \int_{r=0}^{r=1} dC_T. \quad (49.48)$$

For the rotor torque coefficient and power coefficient, their incremental calculation is also provided by BEMT:

$$dC_Q = dC_P = \frac{\sigma}{2}(\phi C_l + C_d)r^3 dr, \quad (49.49)$$

where C_d is the rotor drag coefficient. By knowing the fact that $\lambda = \phi r$, the incremental power is defined as

$$\begin{aligned} dC_P &= \frac{\sigma}{2}\phi C_l r^3 dr + \frac{\sigma}{2}C_d r^3 dr \\ &= \frac{\sigma}{2}C_l \lambda r^2 dr + \frac{\sigma}{2}C_d r^3 dr \\ &= dC_{P_u} + dC_{P_o}. \end{aligned} \quad (49.50)$$

The induced power coefficient can be obtained by integrating dC_{P_u} as follows:

$$C_{P_u} = \int_{r=0}^{r=1} dC_{P_u} = \int_{r=0}^{r=1} \lambda dC_{T_u}, \quad (49.51)$$

and the profile part of the rotor power is given by

$$C_{P_o} = \frac{\sigma}{2} \int_0^1 C_d r^3 dr. \quad (49.52)$$

49.3.5.2 Bare Rotor Flapping Dynamics

Similar to that of the fixed-pitch coaxial case, the rotor flapping dynamics of the variable-pitch coaxial helicopter is also seen as a rigid disk which can tilt about its longitudinal and lateral axes. However, instead of directly linking the flapping angles' dynamics to the servo inputs, the flapping angles (a_i , b_i) are first related to the cyclic pitch angles ($\theta_{cyc,bi}$, $\theta_{cyc,ai}$) as intermediate variables. The presence of $\theta_{cyc,bi}$ and $\theta_{cyc,ai}$ is a joint consequence of servo inputs and the stabilizer bar flapping angles. The detailed description of the rotor equations is extremely complicated. Here, a simplified formulation is adopted, where the rotor forces and moments are expressed as a polynomial function of the rotor state variables (Mettler 2002).

By removing the higher-order terms of the TPP equation, the remaining first-order rotor dynamics could be expressed as

$$\tau \dot{b}_i = -b_i - \tau p + B_a a_i + \theta_{\text{cyc}, bi}, \quad (49.53)$$

$$\tau \dot{a}_i = -a_i - \tau q + A_b b_i + \theta_{\text{cyc}, ai}, \quad (49.54)$$

where

$$A_b = -B_a = \frac{8K_\beta}{\gamma \Omega^2 J_\beta}, \quad (49.55)$$

$$\tau = \frac{16}{\gamma_r \Omega} \left(1 - \frac{8e}{3R} \right)^{-1}, \quad (49.56)$$

$$\gamma = \frac{\rho c C_{l\alpha} R^4}{J_\beta}. \quad (49.57)$$

a_i and b_i are the first-order TPP flapping angles in the longitudinal and lateral directions for either the upper rotors or the lower rotors. τ and γ are the flapping time constant and the lock number of the rotor blades, respectively. J_β is the blade moment of inertia. $\theta_{\text{cyc}, ai}$ and $\theta_{\text{cyc}, bi}$ are the longitudinal and lateral cyclic pitch of rotor blade. The other terms are defined by the same symbols as that of the fixed-pitch coaxial helicopter analysis in Sect. 49.3.4. The approximate formulation in Eqs. 49.53 and 49.54 characterizes the crucial TPP responses with respect to rotor cyclic control inputs.

49.3.5.3 Stabilizer Bar Flapping Dynamics

The stabilizer bar, which is attached to the upper main rotor shaft via a free-teetering hinge, can be regarded as a third disk. It consists of two paddles and a steel rod. The stabilizer bar is not designed to produce thrust or moment on the main hub, whereas its main function is to adjust the helicopter dynamics via the Bell-Hiller mixer by augmenting the cyclic pitch command of the upper rotor. It serves as a feedback system which increases the helicopter robustness against wind gust and turbulence (Cai et al. 2011). The flapping dynamics of stabilizer bar can be expressed as two first-order differential equations:

$$\dot{\phi}_{\text{sb}} = -q - \frac{1}{\tau_{\text{sb}}} \phi_{\text{sb}} + \frac{C}{\tau_{\text{sb}}} \delta_{\text{ail}}, \quad (49.58)$$

$$\dot{\theta}_{\text{sb}} = -p - \frac{1}{\tau_{\text{sb}}} \theta_{\text{sb}} + \frac{D}{\tau_{\text{sb}}} \delta_{\text{ele}}, \quad (49.59)$$

where τ_{sb} is the stabilizer bar flapping time constant, and it can be calculated as

$$\tau_{\text{sb}} = \frac{16}{\gamma_{\text{sb}} \Omega}, \quad (49.60)$$

where γ_{sb} is the stabilizer bar lock number:

$$\gamma_{sb} = \frac{\rho c_{sb} C_{l\alpha, sb} (R_{sb}^4 - r_{sb}^4)}{I_{\beta, sb}}. \quad (49.61)$$

The free-teetering hinge does not constrain the flapping of the stabilizer bar; thus, there is no coupling between the longitudinal and lateral flapping motions. The augmented rotor cyclic pitch of upper rotor can be expressed as

$$\theta_{cyc, aup} = A_a \delta_{ele} + K_{sb} \phi_{sb}, \quad (49.62)$$

$$\theta_{cyc, bup} = B_b \delta_{ail} + K_{sb} \theta_{sb}, \quad (49.63)$$

where K_{sb} is the ratio of rotor blade cyclic pitch to stabilizer bar flapping.

49.3.5.4 Lumped Flapping Dynamics

In this variable-pitch coaxial configuration, the upper rotor and the lower rotor receive the same cyclic input (δ_{ail} , δ_{ele}) since the top swashplate and bottom swashplate are always parallel. To minimize the overall complexity of the model, the two counterrotating rotor disks are treated as one equivalent rotor disk with respect to flapping motions. This assumption is only valid when the helicopter does not perform rapid maneuvering. By making this assumption, the model can be simplified to a large extent yet maintaining reasonable fidelity. The imaginary rotor has equivalent longitudinal and lateral angles expressed as a_s and b_s . Combining Eqs. 49.53–49.63, the lumped flapping dynamics subsystem could be represented in the following state-space form:

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}, \quad (49.64)$$

$$\dot{\mathbf{y}} = \mathbf{C} \mathbf{x}, \quad (49.65)$$

where

$$\mathbf{x} = \begin{pmatrix} p \\ q \\ a_s \\ b_s \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \delta_{ail} \\ \delta_{ele} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} p \\ q \end{pmatrix}, \quad (49.66)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & L_b \\ 0 & 0 & M_a & 0 \\ 0 & -1 & -\frac{1}{\tau_s} & \frac{A_b}{\tau_s} \\ -1 & 0 & \frac{B_a}{\tau_s} & -\frac{1}{\tau_s} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & A'_{lon} \\ B'_{lat} & 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (49.67)$$

$$L_b = \frac{mg H_{mr} + K_\beta}{J_{xx}}, \quad M_a = \frac{mg H_{mr} + K_\beta}{J_{yy}}. \quad (49.68)$$

49.4 Model-Based Parameter Identification

When model formulation of a certain type of aerial platform has been derived, the next step is to identify unknown parameters of the derived model for a particular case. This is commonly known as the model-based parameter identification. In this section, several common parameter identification methods will be introduced based on the case study of ESky Big Lama, belonging to the fixed-pitch coaxial helicopter. It will be seen that some of the model parameters can be directly measured, while the remaining ones need special test bench experiments or flight tests to be carried out. The identification procedures for the case of variable-pitch coaxial helicopter will not be repeated since similar methods can be applied. It is also hoped that the readers will try the suggested experimental setups on other types of aerial platforms if they face similar parameter identification problems in their UAV-related projects.

49.4.1 Direct Measurement

For the ESky Big Lama fixed-pitch coaxial helicopter, some of its model parameters, especially those inherently defined by the platform dimension, geometry, weight loading, and the operational environment, can be directly measured or algebraically calculated, for example, the total mass of the platform (m), the distance from the rotor hubs to CG ($|\mathbf{l}_{up}|$ and $|\mathbf{l}_{dw}|$), the air density (ρ), the rotor diameter (R), the effective drag area of the fuselage (S_x , S_y , S_z), and the moment of inertia of the upper and lower rotor system (J_{up} , J_{dw}). Table 49.4 shows all the parameter values that can be identified through direct measurement and simple calculation.

49.4.2 Test Bench Experiment

In most aerial vehicle modeling cases, the direct measurement method is only able to determine a small portion of the parameters. The remaining majority parameters have to be identified by additional test bench experiments or actual flight tests. In this subsection, some common test bench methods are introduced, and they are again illustrated based on the ESky Big Lama fixed-pitch coaxial helicopter case.

First of all, the diagonal elements of the helicopter moment of inertia matrix J_{xx} , J_{yy} , J_{zz} can be measured by the so-called trifilar pendulum method proposed in Harris (1996). The experimental setup is shown in Fig. 49.7. In this experiment, the coaxial platform is suspended by three flexible strings with equal length l . The horizontal distances between the attached points and the CG are l_1 , l_2 , and l_3 , respectively. One can slightly twist and release the platform around the z -axis and record the oscillation period t_l . The moment of inertia is then given by

$$J_{zz} = \frac{mgl_1l_2l_3t_l^2}{4\pi^2l} \cdot \frac{l_1 \sin \alpha_1 + l_2 \sin \alpha_2 + l_3 \sin \alpha_3}{l_2l_3 \sin \alpha_1 + l_1l_3 \sin \alpha_2 + l_1l_2 \sin \alpha_3}, \quad (49.69)$$

Table 49.4 Parameters determined via direct measurement

| Parameter | Physical meaning |
|--|--|
| $\rho = 1.204 \text{ kg m}^{-3}$ | Air density |
| $m = 0.977 \text{ kg}$ | Total mass of the platform |
| $R = 0.250 \text{ m}$ | Rotor radius |
| $g = 9.781 \text{ m s}^{-2}$ | Earth gravity |
| $S_{fx} = 0.00835 \text{ m}^2$ | Effective longitudinal fuselage drag area |
| $S_{fy} = 0.01310 \text{ m}^2$ | Effective lateral fuselage drag area |
| $S_{fz} = 0.01700 \text{ m}^2$ | Effective vertical fuselage drag area |
| $ l_{up} = 0.195 \text{ m}$ | Distance from platform CG to the upper rotor hub |
| $ l_{dw} = 0.120 \text{ m}$ | Distance from platform CG to the lower rotor hub |
| $J_{up} = 6.8613 \cdot 10^{-4} \text{ kg m}^2$ | Moment of inertia of the upper rotor with stabilizer bar |
| $J_{dw} = 3.2906 \cdot 10^{-4} \text{ kg m}^2$ | Moment of inertia of the lower rotor |

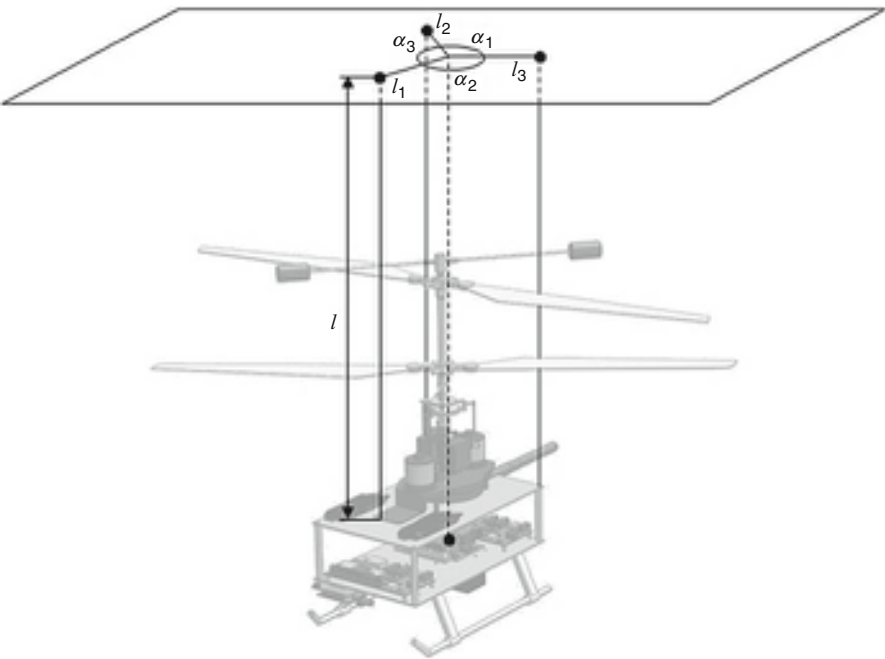


Fig. 49.7 The trifilar pendulum method

where α_1 , α_2 , and α_3 are the angles denoted in Fig. 49.7. Similar experiments can be done to obtain the moment of inertia around the y and z axes. Figure 49.8 shows the experimental setups to carry out this trifilar pendulum method to obtain the moment of inertia of the Esky Big Lama in all three axes.

Next, to identify the rotor thrust coefficient and torque coefficient ($k_{T,i}$ and $k_{Q,i}$), two self-designed test bench experiments were carried out

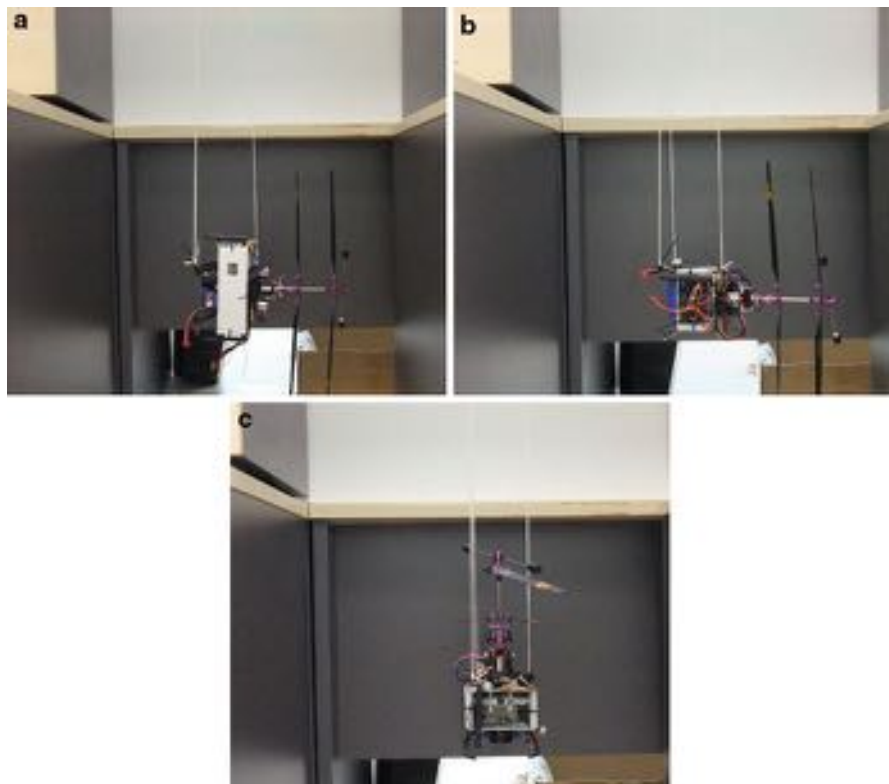


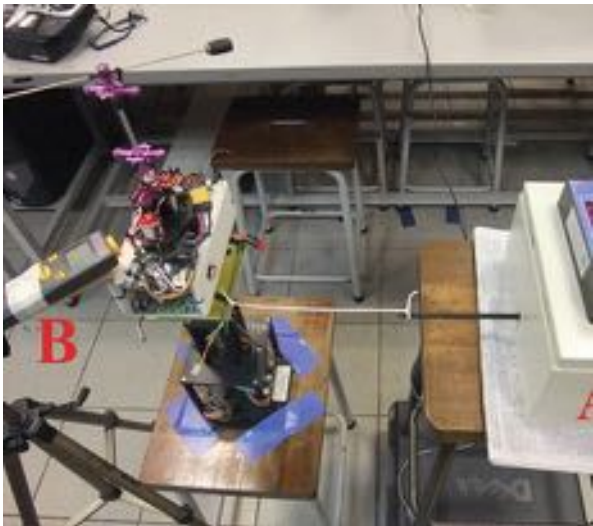
Fig. 49.8 The setups to test helicopter moment of inertia. (a) x -axis. (b) y -axis. (c) z -axis

(see Figs. 49.9 and 49.10). The main measurement sensors include a force meter and a tachometer. When different values of pulse width modulation (PWM) signals are given to the motors, the steady-state rotor rotational speed and the generated thrust/torque are recorded. For the thrust experiment, results are summarized in Fig. 49.11. There are four lines in the plot, in which two of them (solid lines) perfectly match. They represent the cases when only one rotor, either the upper rotor or the lower rotor, is rotating. The dashed line on the top is a numerical combination of the two solid lines, while the dash-dot line comes from the actual test with both rotors spinning at the same speed. The gap between the two lines shows a drop in thrust efficiency caused by aerodynamic interactions between the two rotors. According to Deng et al. (2003), for a coaxial helicopter operating in near-hover condition, the induced-velocity effect of the upper rotor to the lower rotor is significantly larger than that of the lower rotor to the upper rotor. Thus, the loss of thrust efficiency can be assumed to be fully absorbed by the lower rotor thrust coefficient. Hence, $k_{T,up}$ is the gradient of the solid line, and $k_{T,dw}$ is the gradient difference between the dash-dot line and the solid line. For the torque experiment, results are summarized in Fig. 49.12. The solid line represents the case when only

Fig. 49.9 Setup to investigate relation between rotor thrust and rotor rotational speed



Fig. 49.10 Setup to investigate relation between rotor torque and rotor rotational speed



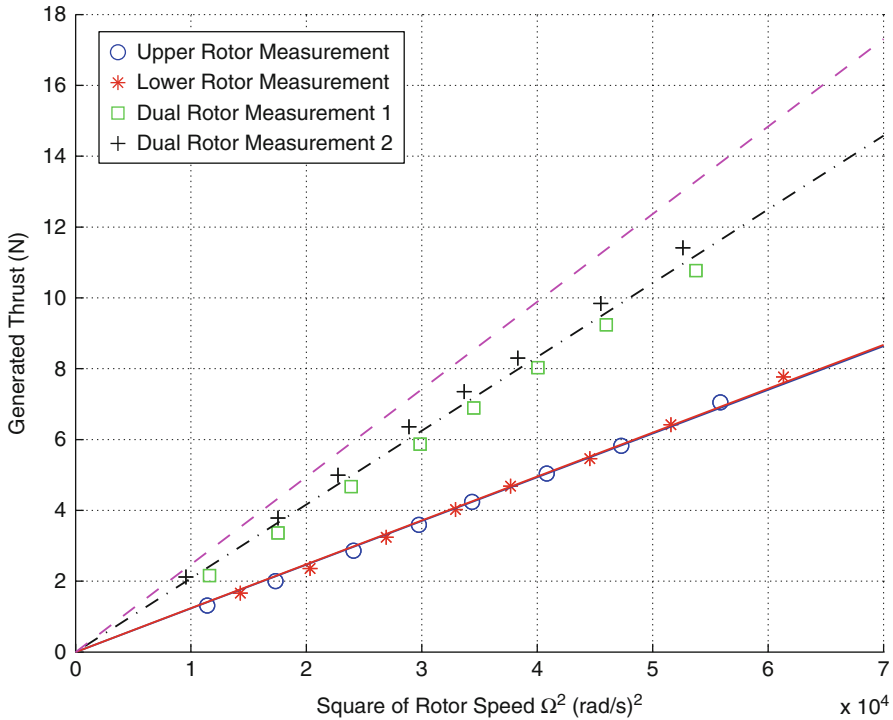


Fig. 49.11 Data plot of thrust against square of rotor speed

the stabilizer bar is rotating, while the dash-dot line is for a single rotating rotor. The dashed line is generated with the upper rotor and the stabilizer bar spinning together. Unsurprisingly, it matches the numerical combination of the lower two lines. Thus, the gradient of the dashed line is $k_{Q,up}$, and the gradient of the dash-dot line is $k_{Q,dw}$.

The identification of parameters involved in the motor dynamics can be done via test bench experiments also. The method to determine motor time constant (τ_{mt}) is a bit tricky, as the transient response of the rotor speed with a step motor input is very difficult to be recorded in real time. As such, instead of examining the transient response of the rotor speed with motor step input, the transient response of the input voltage subject to the changes of the motor Back-EMF (voltage generated by the spinning motor) is recorded using an oscilloscope (see Fig. 49.13). In theory, the time constant of the two transient responses should be the same. On the other hand, m_{up} and m_{dw} can be identified by plotting the steady-state relationship between the rotor speed and the normalized motor input (see Fig. 49.14). m_{up} and m_{dw} are the gradients of the two fitted lines in the figure.

The headlock gyro forms the most inner-loop control in the helicopter yaw channel. As mentioned in Sect. 49.3.4, it is a PI controller with three parameters (K_a , K_P , K_I) to be identified. The identification of K_a can be done via a hovering

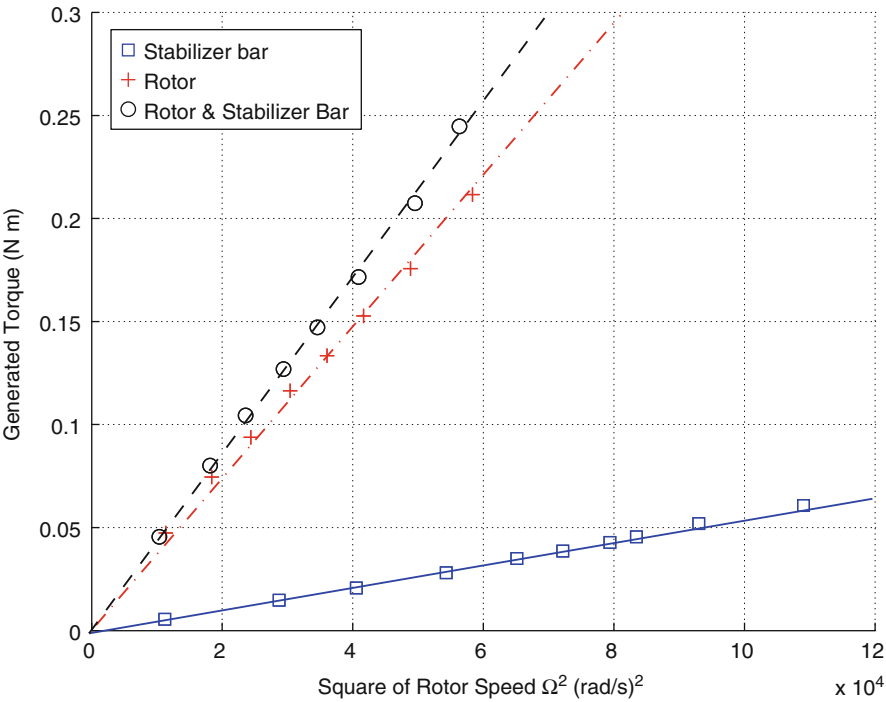


Fig. 49.12 Data plot of torque against square of rotor speed

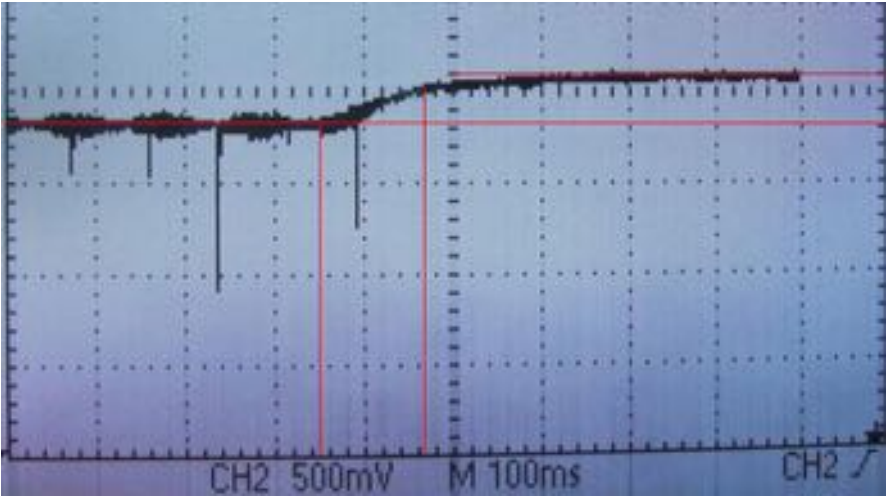


Fig. 49.13 Estimation of time constant of motor speed dynamics

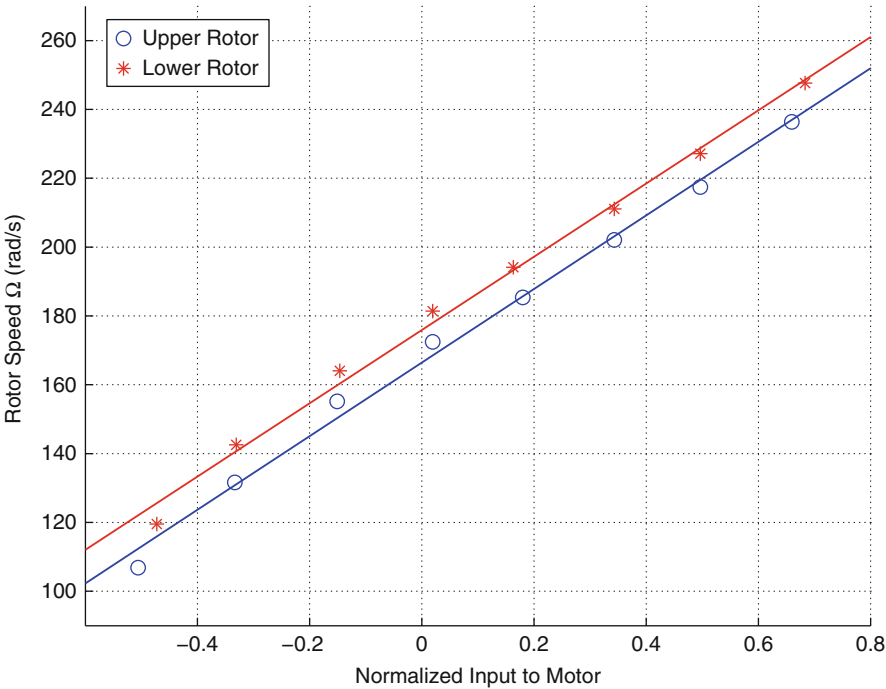


Fig. 49.14 Data plot of rotor speed against motor input

Table 49.5 Yaw rate against rudder input

| | | | | |
|------------------------|-------|-------|-------|-------|
| r (rad/s) | −1.50 | −2.50 | −2.60 | −3.50 |
| δ_{rud} (−1, 1) | 0.25 | 0.35 | 0.40 | 0.55 |

turn test of the coaxial helicopter with different steady-state yaw rates. This test belongs more to the test bench category instead of flight test because it can be done on a swivel table with minimal friction. Table 49.5 shows four sets of data recorded. By plotting the least-square-fit line (see Fig. 49.15) and calculating its gradient, K_a can be determined. For the identification of K_P and K_I , the helicopter is placed stationary on a table. K_P and K_I can be determined by observing the headlock gyro output signal (in pulse width modulation form) caused by a small known step inputs. The initial ratio between the output and the input is $K_P \cdot K_a$, while the climbing rate of the step response is $K_I \cdot K_a$.

Last but not least, test bench experiments are also capable of determining some of the model parameters involved in the rotor flapping dynamics. First of all, by tilting the helicopter suddenly with rotor rotating at hovering speed and observing the transient step response of the stabilizer bar TPP (see Fig. 49.16) by a high-speed camera, the time constant (τ_{sb}) can be found to be about 0.2 s. In addition, the on-axis parameters that appear in the rotor flapping equations ($A_{a,up}$, $B_{b,up}$, $A_{a,dw}$, and $B_{b,dw}$) can be roughly identified by measuring lengths and angles under

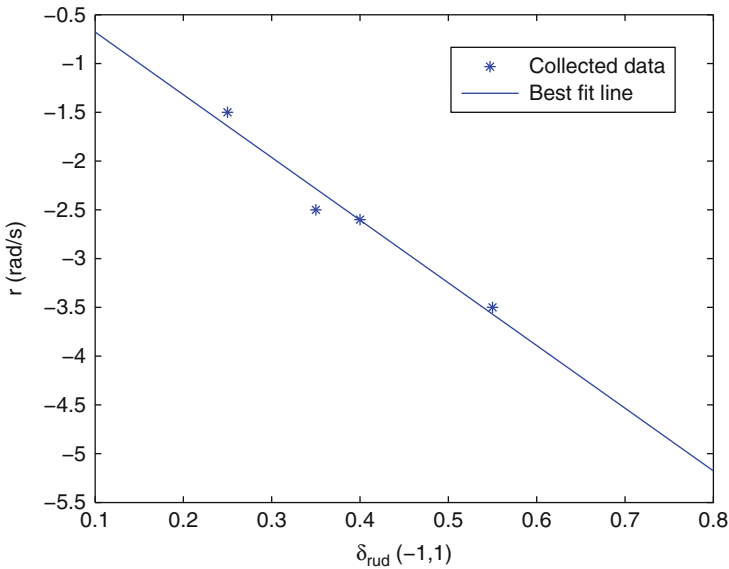


Fig. 49.15 Data plot of yaw rate against rudder input

extreme conditions (see Figs. 49.17 and 49.18) and assume a linear proportional relationship between each pair of them. It should be noted that the parameters related to rotor flapping dynamics are most critical to the whole coaxial helicopter model. The above rough measurements may not be good enough to finalized their values. However, they can be used as an initial guess and get fine-tuned later by a model-based numerical search method via in-flight test data. In the next subsection, this method will be explained in detail. Here, Table 49.6 only lists those parameters that are already finalized at this test bench experiment stage.

49.4.3 Flight Test

After the majority of parameters have been identified, the remaining ones and a few uncertain ones can be identified and refined by analyzing flight test data with input perturbations (frequency sweeping). The recommended software for this task is called “Comprehensive Identification from FrEquency Responses” (CIFER). It is a MATLAB-based software package developed by NASA Ames Research Center for military-based rotorcraft system identifications.

In this Esky Big Lama case, since the remaining unidentified parameters are all about rotor flapping dynamics, only aileron and elevator perturbations need to be done to collect the relevant data for CIFER analysis. However, CIFER and most other parameter identification tools can only handle linear models. Hence, linearization needs to be done first to relate the aileron and elevator inputs to

Fig. 49.16 Step response of stabilizer bar TPP motion (*top*: $t = 0$; *center*, $t \approx 0.2$ s; *bottom*, $t = \infty$)

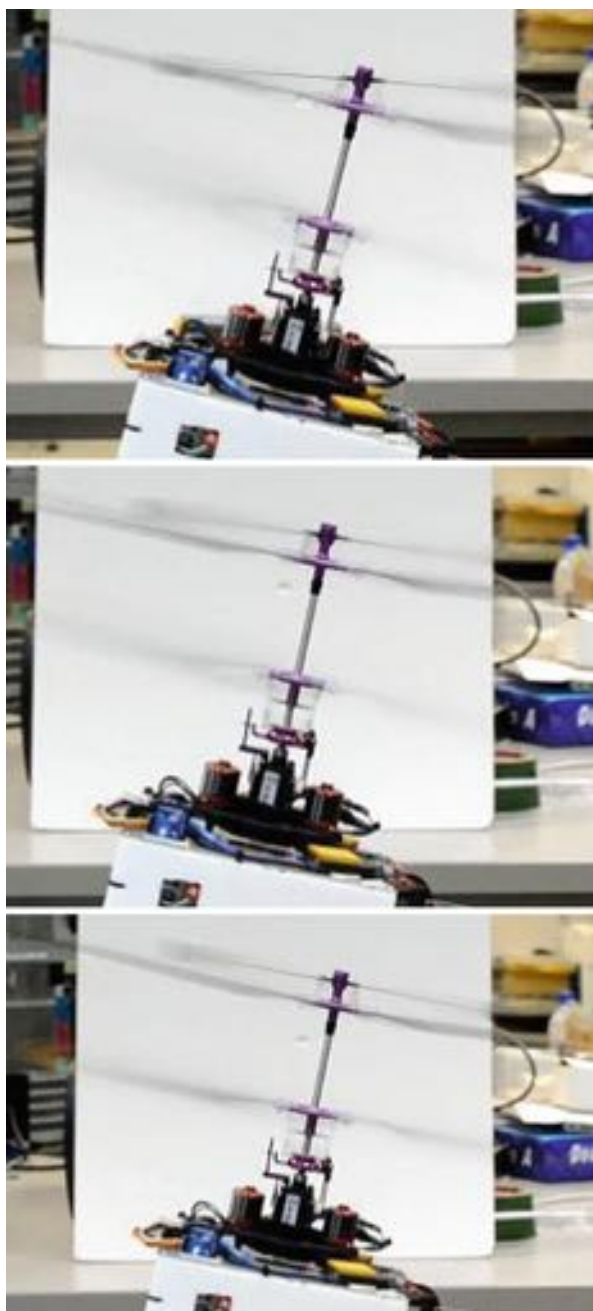




Fig. 49.17 *Left*: maximum teetering angle of the lower rotor hub; *right*: maximum flapping angle of the lower rotor

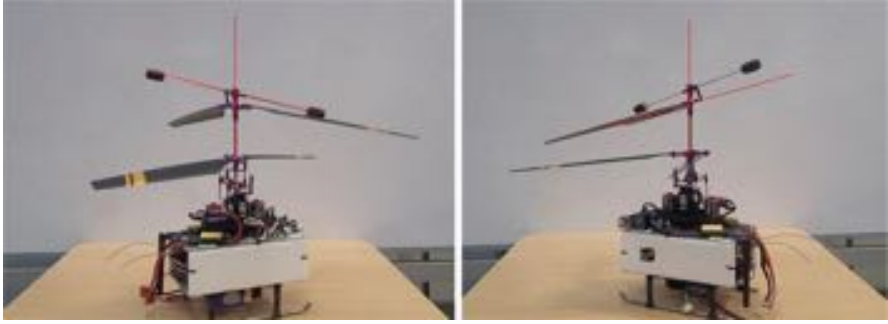


Fig. 49.18 *Left*: maximum teetering angle of the stabilizer bar; *right*: maximum teetering angle of the upper rotor hub

the helicopter roll, pitch angular rates. By relating and linearizing all the model formulation related to roll-pitch rate dynamics and upper rotor flapping dynamics at the hovering condition, one can obtain the following fourth-order linear state-space approximation:

$$\begin{pmatrix} \dot{p} \\ q \\ a_{up} \\ b_{up} \end{pmatrix} = \begin{bmatrix} \frac{-X_{dw}B_{p,dw}}{J_{xx}} & 0 & 0 & \frac{X_{up}}{J_{xx}} \\ 0 & \frac{-X_{dw}A_{q,dw}}{J_{yy}} & \frac{X_{up}}{J_{yy}} & 0 \\ -A_{b,up} & -A_{a,up} & -\frac{1}{\tau_{sb}} & 0 \\ -B_{b,up} & -B_{a,up} & 0 & -\frac{1}{\tau_{sb}} \end{bmatrix} \begin{pmatrix} p \\ q \\ a_{up} \\ b_{up} \end{pmatrix} + \begin{bmatrix} \frac{X_{dw}B_{b,dw}}{J_{xx}} & \frac{X_{dw}B_{a,dw}}{J_{xx}} \\ \frac{X_{dw}A_{b,dw}}{J_{yy}} & \frac{X_{dw}A_{a,dw}}{J_{yy}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \delta_{ail} \\ \delta_{ele} \end{pmatrix}, \quad (49.70)$$

where $X_{up} = T_{up}l_{up} + K_{\beta}$ and $X_{dw} = T_{dw}l_{dw} + K_{\beta}$. By treating δ_{ail} , δ_{ele} as the inputs and p , q as the outputs (all can be recorded during flight tests) and giving known

Table 49.6 Parameters determined via test bench experiments

| Parameter | Physical meaning |
|--|--|
| $J_{xx} = 0.0059 \text{ kg m}^2$ | Platform moment of inertia in the x -axis |
| $J_{yy} = 0.0187 \text{ kg m}^2$ | Platform moment of inertia in the y -axis |
| $J_{zz} = 0.0030 \text{ kg m}^2$ | Platform moment of inertia in the z -axis |
| $k_{T,\text{up}} = 1.23 \cdot 10^{-4} \text{ N s}^2 \text{ rad}^{-2}$ | Effective thrust coefficient of the upper rotor |
| $k_{T,\text{dw}} = 8.50 \cdot 10^{-5} \text{ N s}^2 \text{ rad}^{-2}$ | Effective thrust coefficient of the lower rotor |
| $k_{Q,\text{up}} = 4.23 \cdot 10^{-6} \text{ N ms}^2 \text{ rad}^{-2}$ | Effective torque coefficient of the upper rotor |
| $k_{Q,\text{dw}} = 3.68 \cdot 10^{-6} \text{ N ms}^2 \text{ rad}^{-2}$ | Effective torque coefficient of the lower rotor |
| $m_{\text{up}} = 106.90 \text{ rad s}^{-1}$ | Steady-state ratio between change of upper rotor speed and change of motor input |
| $m_{\text{dw}} = 106.45 \text{ rad s}^{-1}$ | Steady-state ratio between change of lower rotor speed and change of motor input |
| $\Omega_{\text{up}}^* = 203.38 \text{ rad s}^{-1}$ | Trimming rotational speed of the upper rotor |
| $\Omega_{\text{dw}}^* = 217.88 \text{ rad s}^{-1}$ | Trimming rotational speed of the lower rotor |
| $\tau_{\text{mt}} = 0.12 \text{ s}$ | Motor time constant |
| $\tau_{\text{sb}} = 0.2 \text{ s}$ | Stabilizer bar time constant |
| $K_a = 6.4267$ | Feed-forward gain of the headlock gyro system |
| $K_p = 0.667/K_a$ | Proportional feedback gain of the headlock gyro system |
| $K_I = 0.713/K_a$ | Integral feedback gain of the headlock gyro system |

constraints and reasonable initial values (on-axis values from Sect. 49.4.2 and off-axis values as zeros), CIPHER helps to search for optimal numerical solution based on frequency response matching. A stable result with good matching is obtained as follows:

$$\begin{pmatrix} \dot{p} \\ q \\ a_{\text{up}} \\ b_{\text{up}} \end{pmatrix} = \begin{bmatrix} -17.19 & 0 & 0 & 934.1 \\ 0 & -5.360 & 291.3 & 0 \\ 0.2745 & -0.49 & -5 & 0 \\ -0.49 & -0.2745 & 0 & -5 \end{bmatrix} \begin{pmatrix} p \\ q \\ a_{\text{up}} \\ b_{\text{up}} \end{pmatrix} + \begin{bmatrix} -102.48 & -38.08 \\ -11.73 & 31.95 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \delta_{\text{ail}} \\ \delta_{\text{ele}} \end{pmatrix}. \quad (49.71)$$

With this numerical result, Figs. 49.19–49.22 show the corresponding comparison of frequency response between the data collected via actual flight tests and the CIPHER derived model fit. For both the on-axis and off-axis responses, the matching is very good, indicating a high-quality identification result. Next, by comparing Eqs. 49.70 and 49.71, all the parameters involved in angular rate and rotor flapping dynamics can be finalized. Table 49.7 shows the identification results that have been obtained via flight test. Till now, all unknown parameters in the fixed-pitch coaxial model have been identified.

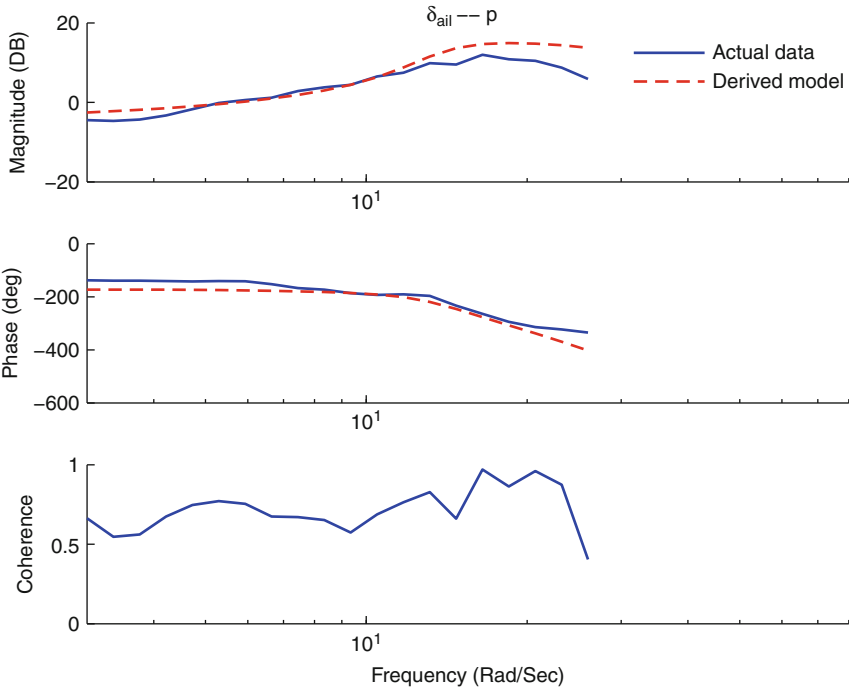


Fig. 49.19 Response comparison using frequency-sweep input ($\delta_{lat} - p$)

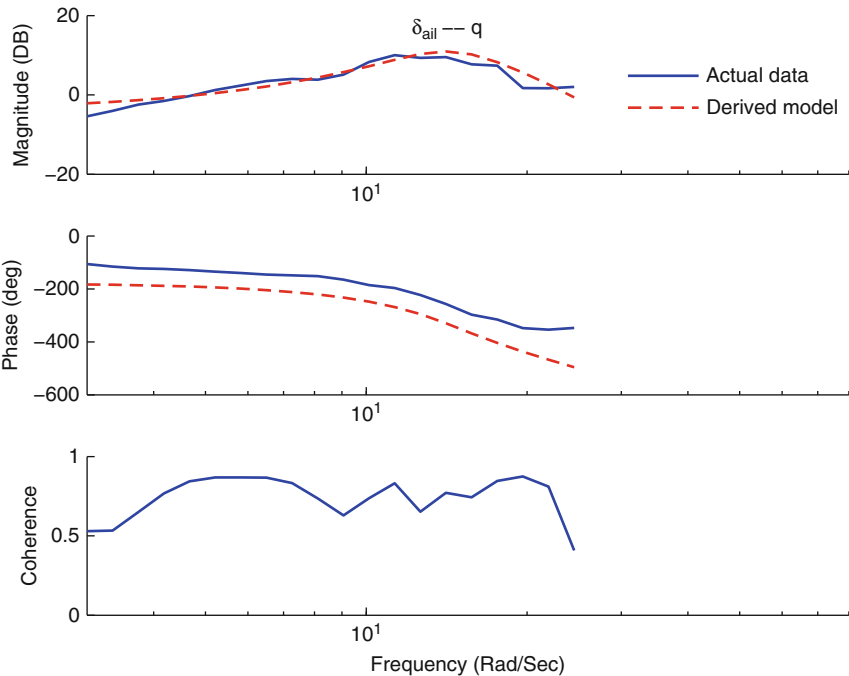


Fig. 49.20 Response comparison using frequency-sweep input ($\delta_{lat} - q$)

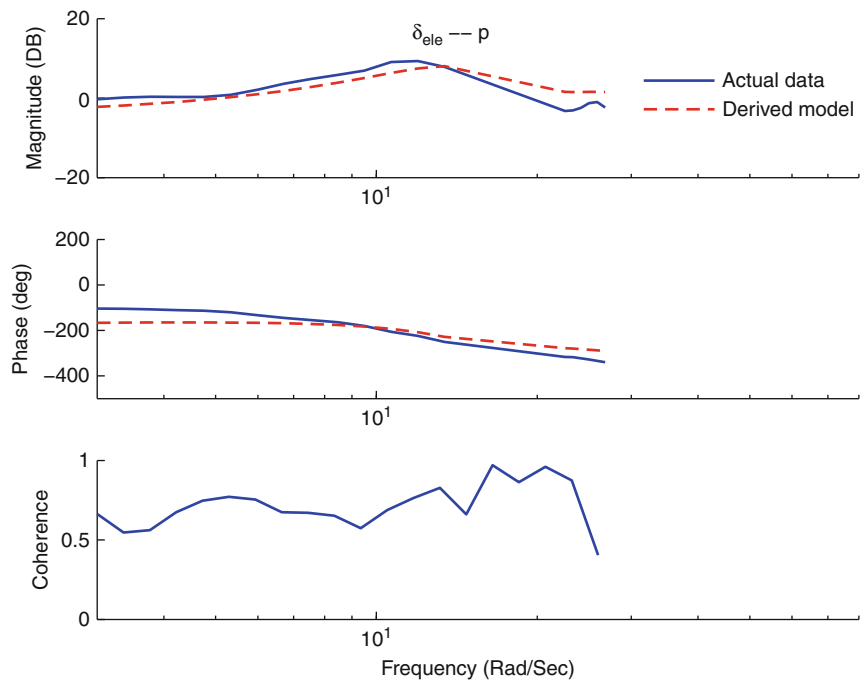


Fig. 49.21 Response comparison using frequency-sweep input ($\delta_{ele} - q$)

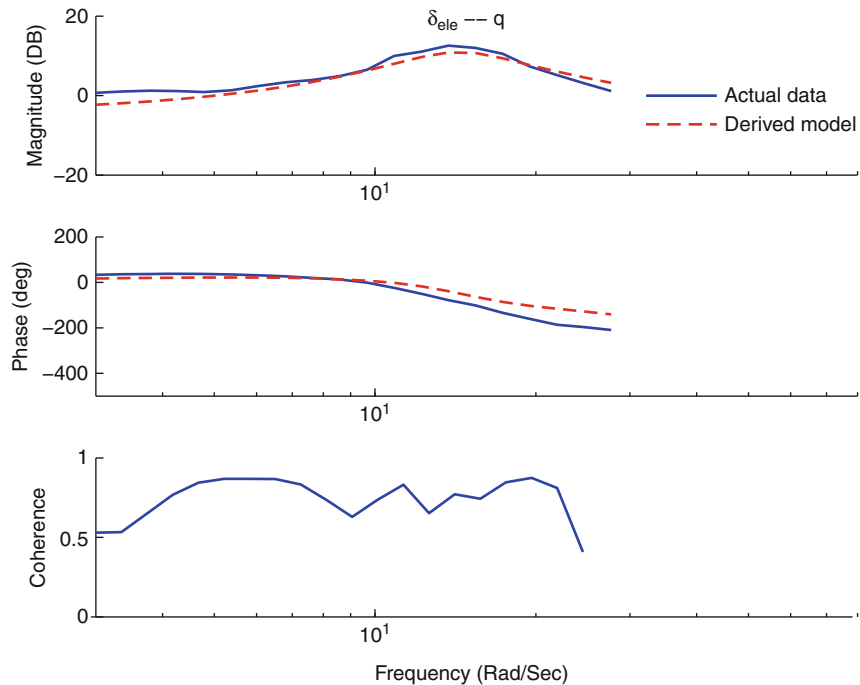


Fig. 49.22 Response comparison using frequency-sweep input ($\delta_{ele} - p$)

Table 49.7 Parameters determined via flight test and CIFER

| Parameter | Physical meaning |
|---|---|
| $A_{a,up} = 0.4900 \text{ rad s}^{-1}$ | DC gain from stabilizer bar pitching angle to upper rotor longitudinal flapping angle |
| $B_{b,up} = 0.4900 \text{ rad s}^{-1}$ | DC gain from stabilizer bar rolling angle to upper rotor lateral flapping angle |
| $A_{b,up} = -0.2745 \text{ rad s}^{-1}$ | DC gain from stabilizer bar rolling angle to upper rotor longitudinal flapping angle |
| $B_{a,up} = 0.2745 \text{ rad s}^{-1}$ | DC gain from stabilizer bar pitching angle to upper rotor lateral flapping angle |
| $A_{a,dw} = 0.1217 \text{ rad s}^{-1}$ | DC gain from elevator input to lower rotor longitudinal flapping angle |
| $B_{b,dw} = -0.1217 \text{ rad s}^{-1}$ | DC gain from aileron input to lower rotor lateral flapping angle |
| $A_{b,dw} = -0.0450 \text{ rad s}^{-1}$ | DC gain from aileron input to lower rotor longitudinal flapping angle |
| $B_{a,dw} = -0.0450 \text{ rad s}^{-1}$ | DC gain from elevator input to lower rotor lateral flapping angle |

49.5 Model Verification

It is always a good practice to verify a derived system model with actual input and output data. In this section, a comprehensive evaluation on the fidelity of the obtained nonlinear model of the ESky Big Lama (fixed-pitch coaxial helicopter) is shown. Four manual flight tests were carried out, which include:

1. Aileron channel perturbation with platform rolling left and right
2. Elevator channel perturbation with platform pitching forward and backward
3. Throttle channel perturbation with platform flying up and down
4. Rudder channel perturbation with platform yawing clockwise and anticlockwise

In each of these four flight tests, the pilot was asked to agitate only one of the four input channels. However, to make sure the helicopter position does not drift too much, minor off-axis inputs were also issued to lightly counter the cross-couplings between the channels. The time-domain results are shown from Figs. 49.23 to 49.26. Based on the recorded inputs, the transient response of the UAV attitudes, angular rates, and body-frame velocities are calculated by a MATLAB simulation program with the aforementioned nonlinear mathematical model (dashed lines in the figures). They are plotted together with the in-flight true data obtained by the onboard sensors (solid lines in the figures). The matching between the two is quite perfect. Note that for the roll and pitch angular rate dynamics, both the on-axis and the off-axis responses match very well, indicating a precise formulation of the coupling terms. Some minor mismatches are caused by the ignorance of high-frequency dynamics when formulating the model, especially for the motion of rotor flapping, which is theoretically highly complicated. Other discrepancies come from ground effect, wind disturbances, and measurement noises present in practical flight tests. In general, this is an accurate cross-coupled model for a fixed-pitch coaxial UAV with low maneuvering speed.

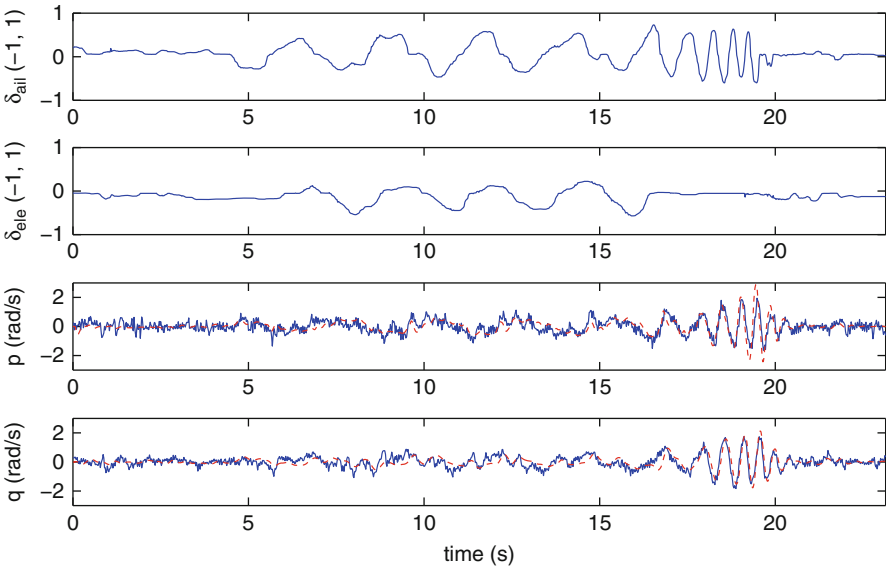


Fig. 49.23 Responses from aileron input perturbation

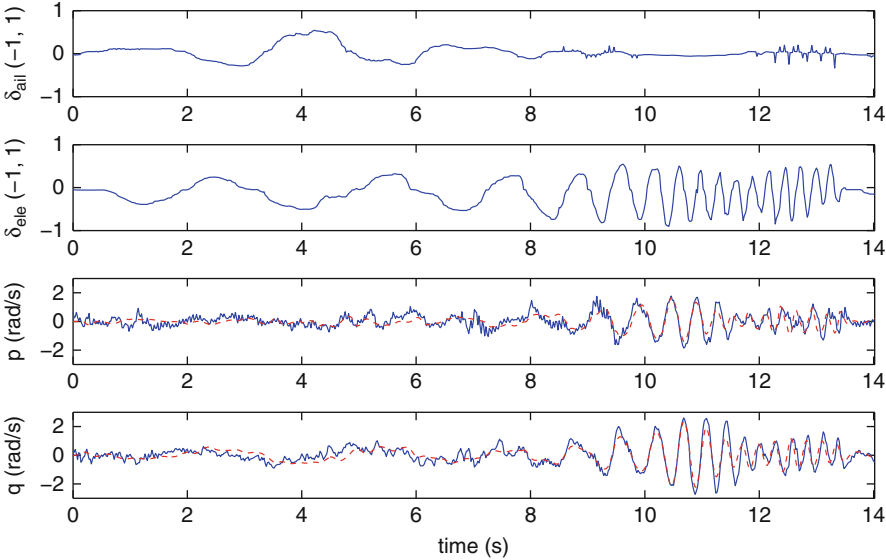


Fig. 49.24 Responses from elevator input perturbation

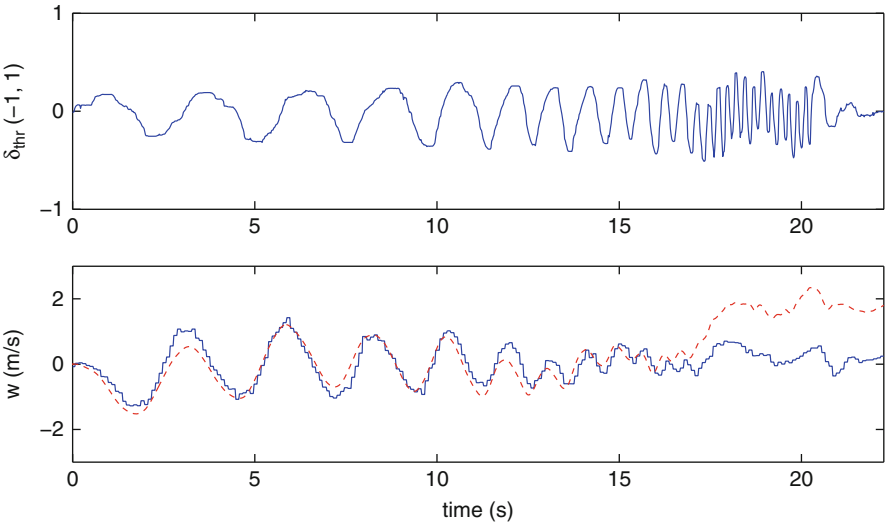


Fig. 49.25 Responses from throttle input perturbation

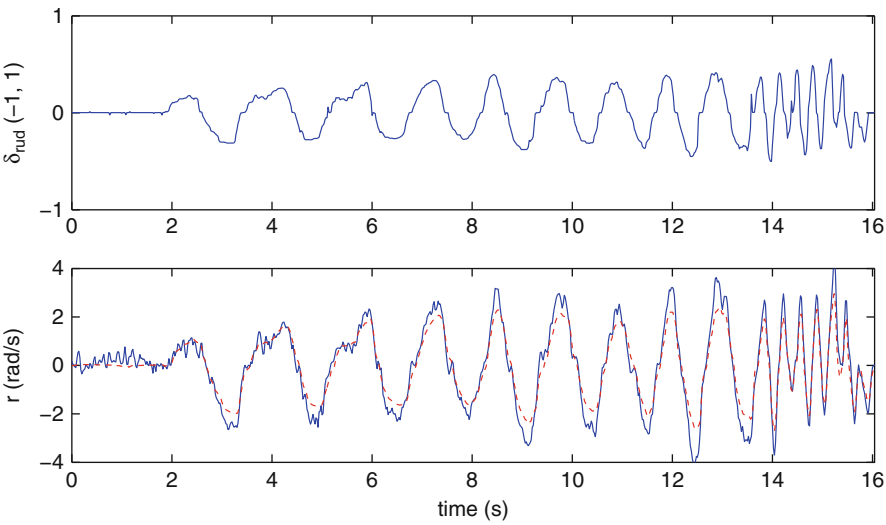


Fig. 49.26 Responses from rudder input perturbation

49.6 Conclusions

This book chapter has demonstrated the main procedures and challenges in modeling flight dynamics of aerial vehicles for the purpose of UAV development. By using the fixed-pitch and variable-pitch coaxial helicopters as examples, the model formulation, parameter identification, and model verification methods are explained in sequence with an adequate level of complexity for people with general engineering backgrounds.

For the model formulation, all analysis starts from the general working principle and model structure of the targeted systems. Formulas to describe the 6DoF kinematics and rigid-body dynamics are listed, and they are universal to all other types of UAVs. The general composition of forces and torques exerted on the UAV fuselage is more or less the same for the fixed-pitch and variable-pitch coaxial helicopters. However, the detailed formulation of individual force and torque generation are different and thus explained separately.

For the model-based parameter identification, the identification methods are categorized into three types, namely, direct measurement, test bench experiments, and flight tests. The ESKy Big Lama, belonging to the fixed-pitch coaxial helicopter, is chosen to be a case study to illustrate some useful test bench and flight test setups to determine the model parameters. After all, a nonlinear model of the ESKy Big Lama flight dynamics is fully derived with all parameters identified.

Finally, model verification is done to prove the feasibility of the whole methodology. By comparing simulation data with the actual in-flight data, the fidelity of the derived nonlinear model is guaranteed. It is also worth noting that the derived model has been actually used in control law design for an indoor coaxial UAV, and good hovering performance has been achieved (Wang et al. 2012).

It is believed that with the proposed systematic modeling methodology, the readers will get insightful knowledge about UAV modeling and parameter identification. Hopefully, these methods can be utilized and applied to other types of aerial platforms, too. It is also welcomed that more detailed aerodynamic formulations and other innovative parameter identification methods can be shared in future and thus complement this work.

References

- C. Bernes, Design and dynamic modeling of autonomous coaxial micro helicopters. Dissertation, Eidgenössische Technische Hochschule ETH Zurich, 2010, Nr. 18847
- A.R.S. Bramwell, G. Done, D. Balmford, *Bramwell's Helicopter Dynamics*, 2nd edn. (Butterworth-Heinemann, Oxford, 2001)
- G. Cai, B.M. Chen, T.H. Lee, *Unmanned Rotorcraft Systems*. Advances in Industrial Control Series (Springer, New York, 2011)
- G. Cai, B.M. Chen, T.H. Lee, K.Y. Lum, Comprehensive nonlinear modeling of a miniature unmanned helicopter. *J. Am. Helicopter Soc.* **57**, 1–13 (2012)
- C.P. Coleman, A survey of theoretical and experimental coaxial rotor aerodynamic research, NASA technical paper 3675 (1997)

- Y. Deng, R. Tao, J. Hu, Experimental investigation of the aerodynamic interaction between upper and lower rotors of a coaxial helicopter. *ACTA Aeronaut. ET Astronaut. Sin.* **24**(1), 10–14 (2003)
- C.M. Harris, *Shock and Vibration Handbook*, 4th edn. (McGraw-Hill, New York, 1996)
- R.K. Heffley, M.A. Mnich, Minimum-complexity helicopter simulation math model, NASA technical report 177476 (1988)
- W. Johnson, *Helicopter Theory* (Dover, Mineola, 1994)
- H.W. Kim, R.E. Brown, Coaxial rotor performance and wake dynamics in steady and manoeuvring flight, in *American Helicopter Society 62nd Annual Forum Proceedings*, Phoenix, 2006, vol. 1, pp. 20–40
- J.G. Leishman, *Principles of Helicopter Aerodynamics* (Cambridge University Press, Cambridge/New York, 2006)
- J.W. Lim, K.W. McAlister, W. Johnson, Hover performance correlation for full-scale and model-scale coaxial rotors. *J. Am. Helicopter Soc.* **54**(3):32005-1–32005-14 (2009)
- B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft* (Kluwer, Norwell, 2002)
- P. Mukherjee, S.L. Waslander, Modeling and multivariable control techniques for small coaxial helicopters, in *AIAA Guidance, Navigation, and Control Conference*, Portland, 2011. AIAA-2011-6545
- D. Neamtu, R. Deac, R.D. Keyser, C. Ionescu, I. Nascu, Identification and control of a miniature rotorcraft Unmanned Aerial Vehicle (UAV), in *AQTR'10: Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics*, Cluj-Napoca, 2010, pp. 1–6
- K. Nonami, F. Kendoul, S. Suzuki, W. Wang, D. Nakazawa, Fundamental modeling and control of small and miniature unmanned helicopters, in *Autonomous Flying Robots*, ed. by K. Nonami et al. (Springer, Tokyo, 2010), pp. 33–60
- O. Rand, V. Khromov, Aerodynamic optimization of coaxial rotor in hover and axial flight, in *27th International Congress of the Aeronautical Sciences*, Nice, 2010, pp. 1–13
- A. Rosen, D. Seter, Vertical autorotation of a single-winged samara, in *Proceedings of the ASME Joint Applied Mechanics/Bioengineering Conference*, Ohio State University, Columbus, 16–19 June 1991
- D. Schaefroth, C. Bermes, S. Bouabdallah, R. Siegwart, Modeling and system identification of the muFly micro helicopter. *J. Intell. Robot. Syst.* **57**(1–4), 27–47 (2010)
- M.K. Taylor, A balsa-dust technique for air-flow visualization and its application to flow through model helicopter rotors in static thrust, NACA technical note 2220 (1950)
- S. Tsach, A. Tatievsky, L. London, Unmanned Aerial Vehicles (UAVs), in *Encyclopedia of Aerospace Engineering*, ed. by R. Blockley, W. Shyy (Wiley, Hoboken, 2010)
- B. Wang, F. Wang, B.M. Chen, T.H. Lee, Robust flight control system design for an indoor miniature coaxial helicopter, in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, Beijing, 2012, pp. 2918–2924

Ying Feng, Camille Alain Rabbath, and Chun-Yi Su

Contents

50.1 Introduction 1258

50.2 Modeling for Quadrotor UAV with Slung Payload 1259

 50.2.1 Quadrotor UAV Dynamic Model 1259

 50.2.2 Modeling of External Slung Payload 1261

50.3 Simulation 1266

 50.3.1 Stabilization Analysis of UAV with Payload 1267

 50.3.2 Estimation of Payload Position 1268

50.4 Conclusion 1270

50.5 Symbols 1271

References 1271

Abstract

In this chapter, the mathematical modeling and simulation of the micro UAV with a payload is derived. When the load is slung underneath the UAVs by cable, the flight dynamics of the UAVs will be altered, which makes the stability of the UAVs disturbed. The unstable oscillation may occur to degrade the performance of the UAVs, and the accurate placement of the load will be affected. Unlike the external disturbance, the negative effects are related to the characteristics of the UAV and the payload. In order to make the UAVs have the ability to adopt the change of the system dynamics and reduce the effects caused by the swing of the load, one modeling method of micro UAVs with single slung payload is addressed. The slung payload is treated as a pendulum-like mass point, and the coupling factors between the UAVs and the payload are considered in the Lagrangian formulation. The conducted model can be used to estimate the

Y. Feng (✉) • C.A. Rabbath • C.-Y. Su
Department of Mechanical and Industrial Engineering, Concordia University, Montreal,
QC, Canada
e-mail: zhdfengying@gmail.com; Rabbath@bell.net; chun-yi.su@concordia.ca

negative effects acting on the UAVs, and it also can be used to assist estimating the trajectory of the load, which provides the possibility to improve the accuracy of placement of the loads.

50.1 Introduction

In recent years, unmanned aerial vehicles (UAVs) have become useful mobile platforms in industrial and academic fields, and their potential applications in numerous areas and their scientific significance in academic research have attracted more attentions (MacKunis et al. 2010; Mahony et al. 2012). Among various UAVs, the micro UAVs can provide an added measure of safety, security, and convenience for different applications, specifically, the micro UAV helicopters are suitable for a limited area or application by using their specific characteristics such as hovering and vertical landing (Marconi et al. 2011; Ryll et al. 2012; Tayebi and McGilvray 2006). In this chapter, one typical micro UAV, quadrotor-type aircraft, is chosen as the platform for the research work.

As a versatile aerial vehicle, one of the potential tasks of the UAV is to carry loads hanging in cable underneath the vehicles. However, carrying a slung load will take a new challenge since the flight characteristics of the UAVs is altered. Besides increasing the mass of UAVs, the slung load will induce oscillations that degrade the UAV's performance. For example, unstable oscillations may occur at high speeds due to the different aerodynamics of the slung loads that results in the crash. Also, the stability of UAV will be disturbed by the slung load, which slows or prevents an accurate placement of the loads. For example, placing a suspended payload accurately on a desired location, the swing load will degrade the placement precision, also the swing load can couple with the vehicle motion leading to poor flight handling performance with a potential for instability. There is, therefore, an interest in the scientific community to develop technologies that can address the challenge in operating UAVs with slung loads (Cruz and Fierro 2012; Min et al. 2011).

The modeling and control for vehicles carrying external suspended loads have been paid attention since the 1950s (Fusato et al. 2001; Hoh and Heffley 2006). Previous work addressed for single-helicopter or multi-helicopter with slung loads (Cicolani and Kanning 1992; Ronen 1985; Theron et al. 2005). Due to the coupling between the helicopter and the loads, it may not be suitable to deal with the oscillation or the instability caused by the slung loads as the external disturbance or the parameter uncertainty directly (Yang et al. 2011), and the related works have explored the specific or general dynamic models of the helicopter system coupled with the swing of the loads, and the control methods are designed for the stabilization of the helicopter systems (Cicolani and Kanning 1992; Fusato et al. 2001).

With the development of the unmanned aerial vehicles over the last few years, the flight stabilization of the UAVs with slung loads poses novel problems for the research of UAVs (Bernard et al. 2008; Bisgaard et al. 2010). The characteristics of the UAVs are different with the conventional helicopters, which makes the available methods may not be shifted to the UAV systems directly (Thanapalan and Wong 2010; Zameroski et al. 2011). For the quadrotor used in this research platform, it is

a four-rotor helicopter, which can be treated as an underactuated, dynamic vehicle with four inputs (four rotors) and six output coordinates (Castillo et al. 2005a; Peng et al. 2009). Different with the conventional helicopters changing the pitch and roll control torques by swashplate, in a quadrotor, rotors are paired to spin in opposite directions so that the quadrotor-type UAV has high payload and relatively easily balanced venter of mass. The quadrotor can be controlled by varying the angular speed of each rotor and moving forward by pitching. These characteristics of quadrotor make it necessary to consider the modeling method of the micro UAVs with payload, which can be used to estimate the performance of the micro UAV systems and to improve the precision of the payload location by choosing the proper cable length, UAV/load mass ratios, etc.

In this chapter, the modeling of quadrotor flying with single slung load is addressed. The approach taken in this chapter is to develop a systematic analytical formulation for general micro UAV systems with slung payload. The micro UAV systems with payload are treated as a class of multibody dynamic systems consisting of two rigid bodies connected by massless cable, and the cable is considered as massless and inelastic link. Therein, the mathematical model is derived from the Newton-Euler equations for rigid bodies and the Lagrange's equations for general dynamic systems.

Besides the stabilization analysis of the micro UAV system slung payload, the position of the payload can be obtained by calculating the suspension angles of the payload, which is useful to estimate the position of the payload.

50.2 Modeling for Quadrotor UAV with Slung Payload

In this section, the derivation of a mathematical model of the micro UAV with slung load is presented. As an illustration, one quadrotor-type UAV with slung load is addressed. The quadrotor system with payload is treated as the rigid body motion of slung-load systems, and the effects caused by the pendulum-like behavior of the slung load are considered in the general simulation model. The modeling method discussed in this chapter is to develop a systematic analytical formulation for micro UAVs with slung payload, which can be utilized to improve the control performance of the UAVs. Firstly, the dynamic model for quadrotor is introduced briefly.

50.2.1 Quadrotor UAV Dynamic Model

Taking inspiration from the work addressing on the classical helicopter, a quadrotor UAV system is studied in this chapter (Castillo et al. 2005a; Min et al. 2011). As shown in Fig. 50.1, the fuselage dynamics and the coordinate system can be divided into an earth frame $\{E\}$ and a body frame $\{B\}$, which are used to describe the relative motions between the two coordinate frames. In order to simplify the highly nonlinear factors in quadrotor UAV system (Min et al. 2011), the following assumptions can be considered in developing the mathematical model of the quadrotor UAV, which are made based on the slower speed and lower altitude:

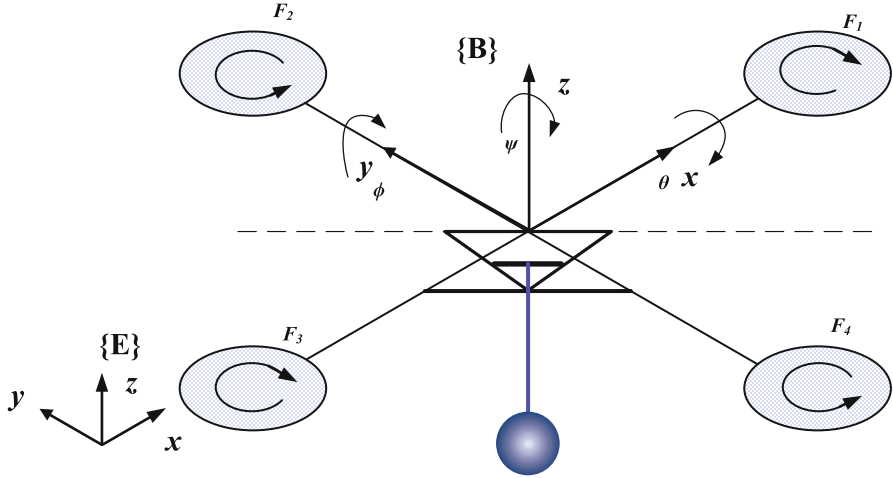


Fig. 50.1 Schematic of quadrotor UAV dynamics

Assumption 1. The center of mass and the body frame origin are assumed to coincide.

Assumption 2. Interaction with ground or other surfaces is neglected.

Assumption 3. The body is rigid and symmetrical.

Utilizing the above assumptions and considering the hovering and flying with low-speed situation, the quadrotor UAV dynamic model (50.1) can be derived via a Lagrange approach to describe a six-degree-of-freedom rigid body model (Alexis et al. 2011), driven by forces and moments as follows:

$$\begin{aligned}
 \ddot{x} &= \frac{U_1(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)}{M} \\
 \ddot{y} &= \frac{U_1(\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi)}{M} \\
 \ddot{z} &= \frac{U_1(\cos \phi \cos \theta)}{M} - g \\
 \ddot{\phi} &= \dot{\theta} \dot{\psi} \left(\frac{J_y - J_z}{J_x} \right) + \frac{l}{J_x} U_2 \\
 \ddot{\theta} &= \dot{\phi} \dot{\psi} \left(\frac{J_z - J_x}{J_y} \right) + \frac{l}{J_y} U_3 \\
 \ddot{\psi} &= \dot{\phi} \dot{\theta} \left(\frac{J_x - J_y}{J_z} \right) + \frac{1}{J_z} U_4
 \end{aligned} \tag{50.1}$$

where (x, y, z) are three positions; (θ, ϕ, ψ) three Euler angles; representing pitch, roll and yaw, respectively; g is the acceleration of gravity; M is the mass of the UAV; J_x, J_y, J_z are the inertia moments applied to the center of mass of UAV; and l is the distance between the propeller and the center of mass of UAV. The system inputs are posed U_1, U_2, U_3 , and U_4 , which can be defined as follows (Castillo et al. 2005a):

$$U_1 = F_1 + F_2 + F_3 + F_4$$

$$U_2 = F_3 - F_1$$

$$U_3 = F_4 - F_2$$

$$U_4 = F_1 + F_3 - F_2 - F_4$$

where F_1, F_2, F_3 , and F_4 represent control forces generated by four rotors, which are proportional to the square of the angle speed. U_1 is the sum of the individual thrusts of each rotor. U_2 and U_3 can change the roll torque and pitch torque, respectively. The yaw motion can be changed by U_4 while keeping U_1 as constant. The characteristics of the quadrotor make the quadrotor-type UAVs with good performance, such as the high payload and relatively easily balanced center of mass.

50.2.2 Modeling of External Slung Payload

In order to obtain the general system equations of motion, the underslung payload is considered as a point mass that behaves like a spherical pendulum from a single point in this chapter. As shown in Fig. 50.2, the payload is approximated as a point mass with an aerodynamic drag force acting on it (Chen 2009). The equations that describe the load dynamics are obtained by considering motion with reference to the longitudinal suspension angles α_x in the x - z plane and α_y in the y - z plane. The investigation of the dynamics of the payload is changed to discuss the motion of the mass point, and the position of the payload is determined by the length of the cable and the suspension angles α_x and α_y .

50.2.2.1 Payload Dynamics

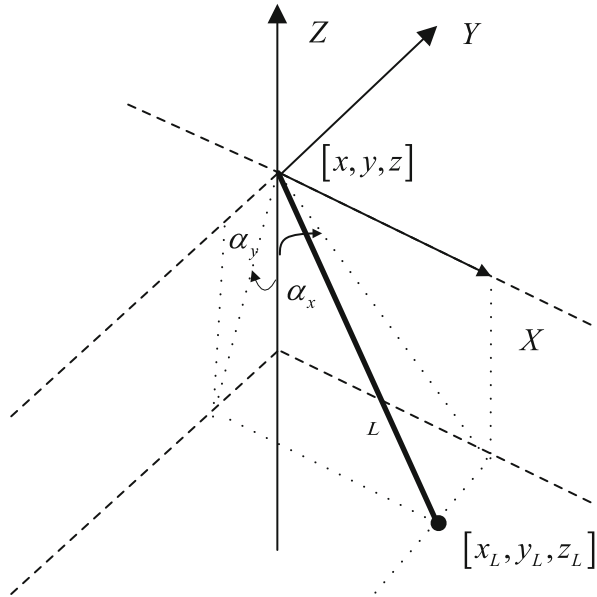
Since the payload is modeled as a three-dimensional point mass pendulum, the position of the slung payload in the frame $\{\mathbf{B}\}$ can be defined as

$$r^B = \begin{bmatrix} 0 \\ 0 \\ L \end{bmatrix} \quad (50.2)$$

where L is the distance between the mass point of the UAV and the payload.

Remark 1. Based on Assumptions 1 and 3, the hook point is assumed under the mass point of the UAV, and the length of the hook with respect to the UAV L_d is considered as the distance between the mass point of the UAV and the UAV. Then, it

Fig. 50.2 The point mass slung payload model



has $L = L_t + L_d$, where L_t is the length of the cable. When $L_t \gg L_d$, the length of the cable L_t can be treated as L for convenience.

To make the payload independent of the UAV attitude change, the payload position will be derived using the earth-fixed coordinates, then the position of the payload with respect to the center of mass of UAV is

$$r = R \begin{bmatrix} 0 \\ 0 \\ L \end{bmatrix} \quad (50.3)$$

where

$$\begin{aligned} R = \text{Rot}(\alpha_y)\text{Rot}(\alpha_x) &= \begin{bmatrix} \cos \alpha_y & 0 & \sin \alpha_y \\ 0 & 1 & 0 \\ -\sin \alpha_y & 0 & \cos \alpha_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_x & -\sin \alpha_x \\ 0 & \sin \alpha_x & \cos \alpha_x \end{bmatrix} \\ &= \begin{bmatrix} \cos \alpha_y & \sin \alpha_x \sin \alpha_y & -\cos \alpha_x \sin \alpha_y \\ 0 & \cos \alpha_x & -\sin \alpha_x \\ -\sin \alpha_y & \sin \alpha_x \cos \alpha_y & -\cos \alpha_x \cos \alpha_y \end{bmatrix} \end{aligned} \quad (50.4)$$

and $\text{Rot}(\alpha_x)$ and $\text{Rot}(\alpha_y)$ are rotational matrices.

The position of the payload using earth-fixed coordinates is redefined as

$$r_L = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} -L \cos \alpha_x \sin \alpha_y \\ -L \sin \alpha_x \\ -L \cos \alpha_x \cos \alpha_y \end{bmatrix} = \begin{bmatrix} x - L \cos \alpha_x \sin \alpha_y \\ y - L \sin \alpha_x \\ z - L \cos \alpha_x \cos \alpha_y \end{bmatrix} \quad (50.5)$$

then the absolute velocity \dot{r}_L of the payload is given by

$$\dot{r}_L = \begin{bmatrix} \dot{x} + L(\dot{\alpha}_y \cos \alpha_x \cos \alpha_y - \dot{\alpha}_x \sin \alpha_x \cos \alpha_y) \\ \dot{y} - L\dot{\alpha}_x \cos \alpha_x \\ \dot{z} + L(-\dot{\alpha}_x \sin \alpha_x \cos \alpha_y - \dot{\alpha}_x \cos \alpha_x \sin \alpha_y) \end{bmatrix} \quad (50.6)$$

To derive the dynamic model of the slung payload, the energy conservation property by means of Lagrange's equations is applied. The effects due to the motion of the quadrotor and the characteristics of the payload are involved in the investigation of the suspension angles.

Define the Lagrangian function Λ as

$$\Lambda = \Gamma - V \quad (50.7)$$

where $\Lambda = \Gamma - V$, Γ is the kinetic energy, and V is the potential energy.

The kinetic energy of the system Γ is the sum of the translational kinetic energy of the payload Γ_t and rotational kinetic energy of the payload Γ_r . Therein, the translational kinetic energy is defined as

$$\Gamma_t = \frac{1}{2} M_L \dot{r}_L^T \dot{r}_L \quad (50.8)$$

and the rotational kinetic energy is defined as

$$\Gamma_r = \frac{1}{2} M_L L^2 \dot{\alpha}_x^2 + \frac{1}{2} M_L L^2 \dot{\alpha}_y^2 \quad (50.9)$$

and the potential energy of the payload is $V = -M_L L \cos \alpha_x \cos \alpha_y$.

The equations of motion for the payload (no direct force acting on the payload) can be derived from the general form of Lagrange's equations as

$$\frac{d}{dt} \left(\frac{\partial \Lambda}{\partial \dot{\alpha}_x} \right) - \frac{\partial \Lambda}{\partial \alpha_x} = 0 \quad (50.10)$$

$$\frac{d}{dt} \left(\frac{\partial \Lambda}{\partial \dot{\alpha}_y} \right) - \frac{\partial \Lambda}{\partial \alpha_y} = 0 \quad (50.11)$$

where $\frac{\partial \Lambda}{\partial \dot{\alpha}_x}$ and $\frac{\partial \Lambda}{\partial \dot{\alpha}_y}$ are the momentum conjugate to α_x and α_y , respectively.

Based on the above definition of Lagrangian function Λ , each item $\frac{d}{dt} \left(\frac{\partial \Lambda}{\partial \dot{\alpha}_x} \right)$, $\frac{\partial \Lambda}{\partial \alpha_x}$, $\frac{d}{dt} \left(\frac{\partial \Lambda}{\partial \dot{\alpha}_y} \right)$, and $\frac{\partial \Lambda}{\partial \alpha_y}$ can be derived by using mathematical softwares, such as Maple.

Remark 2. Solving the equations defined in (50.10) and (50.11), the symbolic expressions for $\ddot{\alpha}_x$ and $\ddot{\alpha}_y$ can be calculated by software, which are the functions of α_x , α_y , $\dot{\alpha}_x$, $\dot{\alpha}_y$, \dot{x} , \dot{y} , \dot{z} , \ddot{x} , \ddot{y} , \ddot{z} , and L_c .

$$\begin{bmatrix} \ddot{\alpha}_x \\ \ddot{\alpha}_y \end{bmatrix} = \mathbf{F}(L, \alpha_x, \alpha_y, \dot{\alpha}_x, \dot{\alpha}_y, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}) \quad (50.12)$$

However, the expressions of $\ddot{\alpha}_x$ and $\ddot{\alpha}_y$ are too complicated to be implemented for the real-time application. Usually, the small-angle assumption $\sin \alpha_x \rightarrow \alpha_x$ and $\sin \alpha_y \rightarrow \alpha_y$ are used, then calculation of the angle acceleration of payload $\ddot{\alpha}_x$ and $\ddot{\alpha}_y$ can be simplified.

Remark 3. In some special cases, the high speed or high acceleration of the UAV may cause the high oscillation of the payload, the small-angle assumption is not applicable. For these cases, $\sin \alpha_x$ and $\sin \alpha_y$ can be expanded as Taylor series $\sin \alpha_x = \alpha_x - \frac{\alpha_x^3}{3!}$ and $\sin \alpha_y = \alpha_y - \frac{\alpha_y^3}{3!}$. Usually, the small-angle assumption is suitable for the regular operation of quadrotor flying with low speeds.

50.2.2.2 Dynamics of Micro UAV with Payload

As a motivational example to show the effects acting on the micro UAV, the dynamics of the quadrotor carrying single payload by cable is discussed in this section. The quadrotor and the payload are considered two separate rigid bodies and the force acting on the axes of quadrotor, respectively.

As shown in Fig. 50.3, the curvilinear motion of the payload will result in the force acting on the quadrotor, which can be separated in the tangential, normal directions. The formulation of the tangential acceleration a_τ and the normal acceleration a_n can be obtained, respectively, using the theorem of motion of center of mass law of conservation of momentum, which are calculated through the suspension angles α_x and α_y . Then, the forces caused by the payload in respective axes are derived as follows:

$$\begin{aligned} F_{ox} &= M_L a_{ox} = M_L a_{\tau x} \cos \alpha_x - M_L a_{nx} \sin \alpha_x \\ &= M_L \ddot{\alpha}_x L \cos \alpha_y \cos \alpha_x - M_L \dot{\alpha}_x^2 L \cos \alpha_y \sin \alpha_x \\ F_{oy} &= M_L a_{oy} = M_L a_{\tau y} \cos \alpha_y - M_L a_{ny} \sin \alpha_y \\ &= M_L \ddot{\alpha}_y L \cos \alpha_x \cos \alpha_y - M_L \dot{\alpha}_y^2 L \cos \alpha_x \sin \alpha_y \\ F_{oz} &= M_L a_{oz} - M_L g \end{aligned}$$

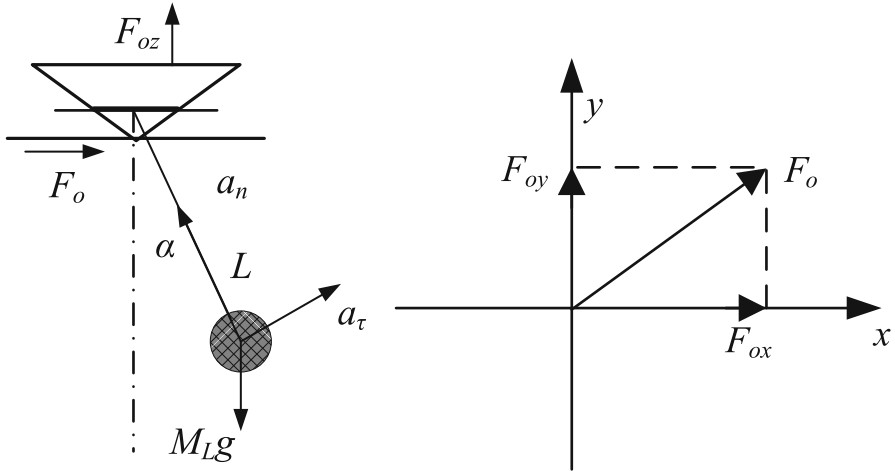


Fig. 50.3 System dynamics of quadrotor with pendulum-like payload

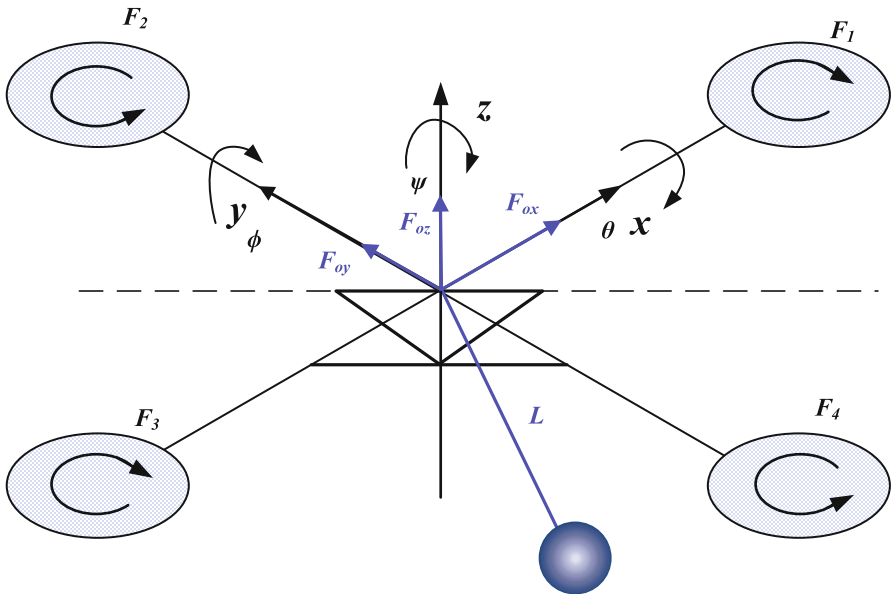


Fig. 50.4 Mechanical model of quadrotor with single payload

$$\begin{aligned}
 &= M_L \ddot{\alpha}_x L \cos \alpha_y \sin \alpha_x + M_L \dot{\alpha}_x^2 L \cos \alpha_y \cos \alpha_x \\
 &\quad + M_L \ddot{\alpha}_y L \cos \alpha_x \sin \alpha_y + M_L \dot{\alpha}_y^2 L \cos \alpha_x \cos \alpha_y - M_L g
 \end{aligned}$$

Using the Newton-Euler function, the mechanical model reflecting the force acting on the quadrotor is shown in Fig. 50.4. Therefore, the model of the quadrotor with payload can be expressed as

$$\begin{aligned}
(M + M_L)\ddot{x} &= U_1(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - F_{ox} \\
(M + M_L)\ddot{y} &= U_1(\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) - F_{oy} \\
(M + M_L)\ddot{z} &= U_1(\cos \phi \cos \psi) - Mg + F_{oz} \\
\ddot{\phi} &= \dot{\theta} \dot{\psi} \left(\frac{J_x - J_z}{J_x} \right) + \frac{l}{J_x} U_2 \\
\ddot{\theta} &= \dot{\phi} \dot{\psi} \left(\frac{J_z - J_x}{J_y} \right) + \frac{l}{J_y} U_3 \\
\ddot{\psi} &= \dot{\phi} \dot{\theta} \left(\frac{J_x - J_y}{J_z} \right) + \frac{1}{J_z} U_4
\end{aligned} \tag{50.13}$$

50.3 Simulation

In order to gain insight into the dynamics of the quadrotor with slung payload, the numerical simulation for the slung payload with quadrotor is presented in this section. The simulation parameters for the quadrotor are chosen as $M = 1.52$ kg, and its moments of inertia were estimated as 0.03, 0.03, and 0.04 kg · m² for the x , y , and z body axes, respectively. The distance L_d between the center of the quadrotor and the anchor point is 20 cm. In order to show the effects caused by the slung payload, the linear quadratic regulator (LQR) control laws (Castillo et al. 2005b) are used in the position controller and orientation controller, and the control parameters are kept as the same for different mass of payload and length of cable.

Remark 4. In this chapter, the main task is to discuss the modeling method of the micro UAV with payload, and the controller used in simulation has been discussed in the previous work for controller design of quadrotor- type UAV. As an illustration to show the system performance with slung payload, the controller used in this simulation can be replaced by other controllers for quadrotor UAV (Fig. 50.5).

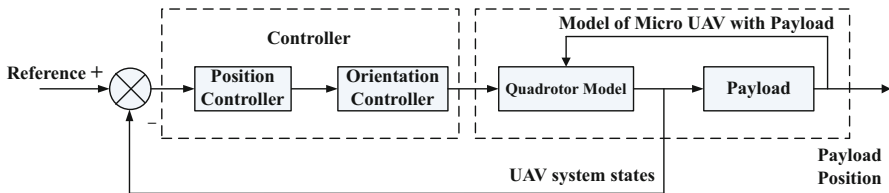


Fig. 50.5 The close-loop scheme of the quadrotor with payload

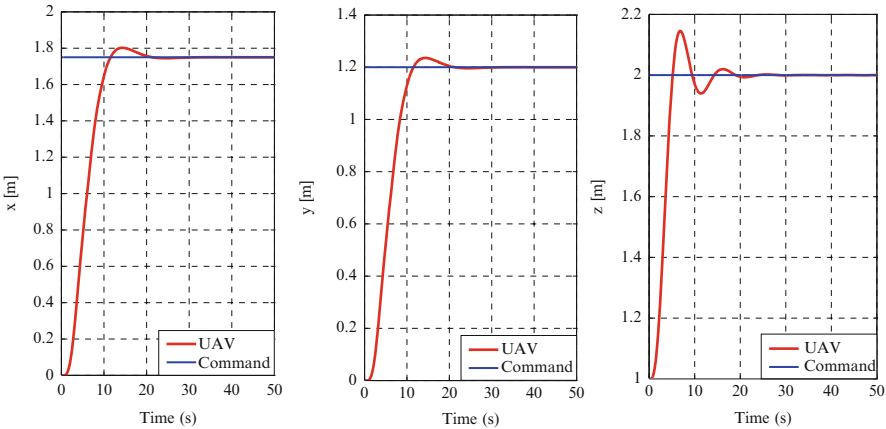


Fig. 50.6 The reference and UAV outputs without payload

Table 50.1 List of payload parameters

| Payload mass (g) | Cable length (cm) | Payload mass (g) | Cable length (cm) |
|------------------|-------------------|------------------|-------------------|
| 68 | 10 | 108 | 10 |
| 68 | 70 | 108 | 70 |

50.3.1 Stabilization Analysis of UAV with Payload

In this section, the modeling method for the quadrotor with payload derived in the previous section is utilized to estimate the unknown effects caused by the payload. The simulation performs the motion of the quadrotor from position (0 m, 0 m, 1 m) to (1.75 m, 1.2 m, 2 m). The quadrotor system simulation response without payload is shown in Fig. 50.6.

In order to illustrate the coupling between the quadrotor and payload which is related to the length of the cable and the mass of the payload, the simulation is conducted with the same controller parameters working on the quadrotor without payload, then compare the simulation results by choosing different payload parameters listed in Table 50.1.

Figures 50.7 and 50.8 show the trajectory oscillation caused by the slung payload $M_L = 68\text{ g}$ with different cables $M_L = 10\text{ cm}$ and $M_L = 70\text{ cm}$. Based on the simulation results, the frequency of the oscillation caused by the slung load is related to the cable length, and the existence of the payload will cause the delay and the increase of the overshoot in z axis, when the controller keeps the same parameters.

Figures 50.9 and 50.10 show the trajectory oscillation caused by the slung payload $M_L = 108\text{ g}$ with the cable length as $L_t = 10\text{ cm}$ and $L_t = 70\text{ cm}$. Compared with the results in Figs. 50.7 and 50.8, the negative effects caused by the payload are increasing with the mass of the payload.

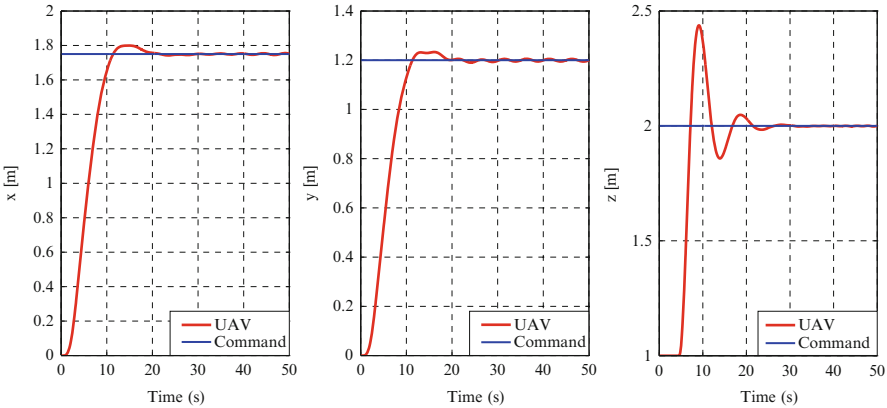


Fig. 50.7 The reference and UAV outputs with payload $M_L = 0.068$ kg and length of cable $L_t = 10$ cm

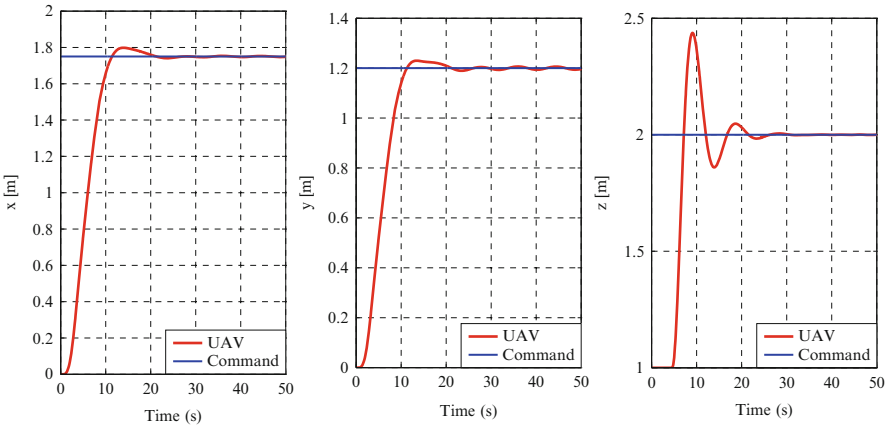


Fig. 50.8 The reference and UAV outputs with payload $M_L = 0.068$ kg and length of cable $L_t = 70$ cm

By constructing the dynamic model for the UAV with payload, the coupling effects related to the length of cable and mass of the payload can be considered in the system dynamics, and these characteristics can be addressed in the control approaches, improving the effectiveness of the control performance.

50.3.2 Estimation of Payload Position

Another purpose for modeling of UAV carrying payload is to estimate the position of the payload, improving the accuracy of placement of payload. Addressing this

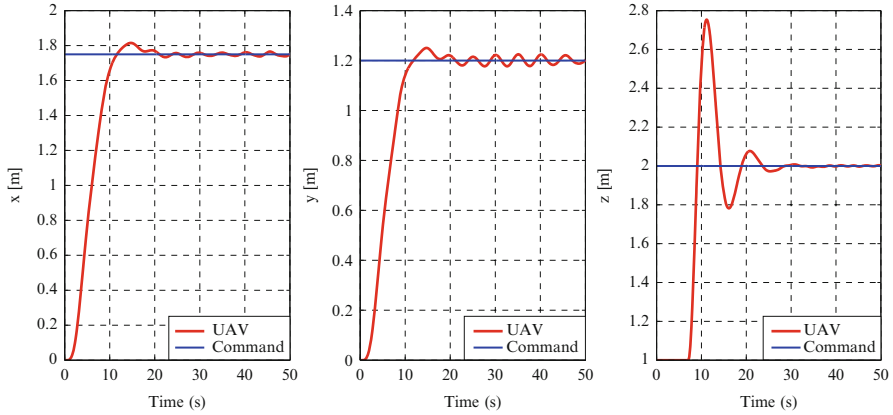


Fig. 50.9 The reference and UAV outputs with payload $M_L = 0.108$ kg and length of cable $L_t = 10$ cm

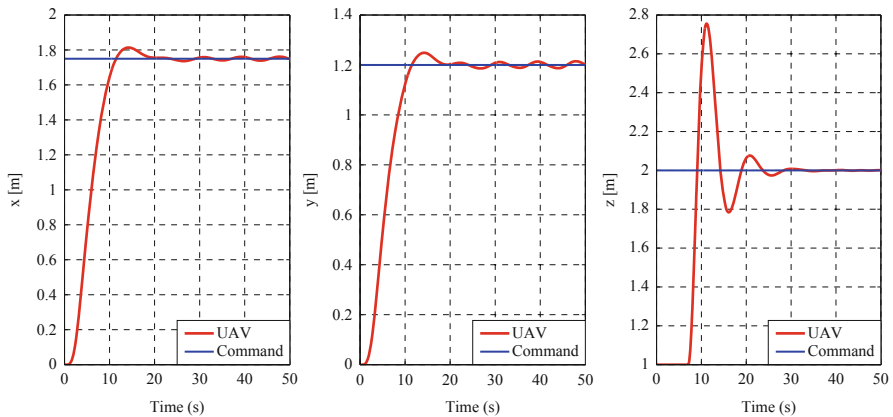


Fig. 50.10 The reference and UAV outputs with payload $M_L = 0.108$ kg and length of cable $L_t = 70$ cm

task, the dynamics modeling for the quadrotor can be used to describe the motion of the payload. Following the definition in the earth-fixed frame $\{E\}$, the position of the payload can be obtained when the suspension angles of the payload α_x and α_y are determined. As an illustration, the trajectories of the payload are shown in Figs. 50.11 and 50.12, when the desired trajectories are set as $x_d = 1.2 \sin(0.1t)$ and $y_d = \sin(0.1t)$.

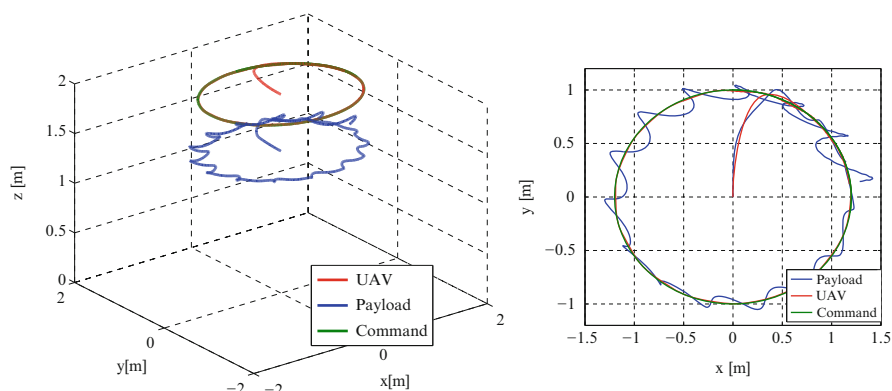


Fig. 50.11 The positions of the reference, quadrotor, and payload with $M_L = 0.068$ kg and $L_t = 37$ cm

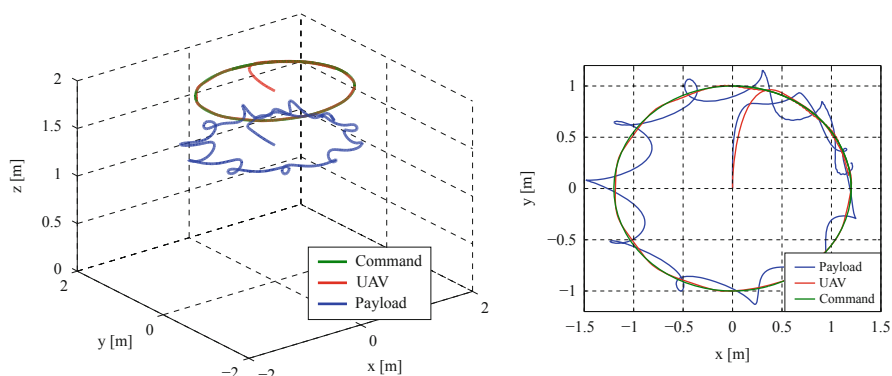


Fig. 50.12 The positions of the reference, quadrotor, and payload with $M_L = 0.108$ kg and $L_t = 37$ cm

50.4 Conclusion

One of the UAVs' tasks is to carry the payload by cable, and the negative effects caused by the slung load are coupled with the UAV systems, degrading the system performance and altering the flight dynamics of UAVs, which poses a new problem for the operating UAVs with slung payload. In this chapter, the model of single UAV with single payload is addressed. The slung payload is treated as a pendulum-like mass point, and the dynamics of the UAV with payload is obtained by formulating Lagrangian function. The derivation of the suspended angle of the payload makes it possible to estimate the position of the payload. The results discussed in this chapter can be utilized for further development of stabilizing and high-performance control laws.

50.5 Symbols

| | | |
|-----------------|--|------------------------|
| E | Inertial world frame | |
| B | UAV body frame | |
| x, y, z | Positron along $\{E\}$ frame x -, y -, and z -axes | [m] |
| θ | Pitch angle of UAV | [rad] |
| ϕ | Roll angle of UAV | [rad] |
| ψ | Yaw angle of UAV | [rad] |
| g | Gravitational acceleration constant | [m/s ²] |
| M | The mass of UAV | [kg] |
| l | The distance between the propeller and the center of mass of UAV | [m] |
| J_x, J_y, J_z | The inertia moments applied to the center of mass of UAV | [kg · m ²] |
| M_L | The mass of payload | [kg] |
| L_t | The length of the cable | [m] |
| L_d | The length of the hook with respect to the UAV | [m] |
| L | The distance between the mass point of the UAV and the payload | [m] |
| α_x | The suspension angle of the payload x - z plane | [rad] |
| α_y | The suspension angle of the payload y - z plane | [rad] |

References

- K. Alexis, G. Nikolakopoulos, A. Tzes, Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Eng. Pract.* **19**, 1195–1207 (2011)
- M. Bernard, K. Kondak, G. Hommel, A Slung load transportation system based on small size helicopters, in *Autonomous Systems – Self-organization, Management, and Control* (2008), pp. 49–61. doi:10.1007/978-1-4020-8889-6-6
- M. Bisgaard, A. La Cour-Harbo, J.D. Bendtsen, Adaptive control system for autonomous helicopter slung load operations. *Control Eng. Pract.* **18**, 800–811 (2010)
- L.S. Cicolani, G. Kanning, Equations of motion of slung-load systems, including multilift systems. NASA report paper, 1992
- P. Castillo, R. Lozano, A. Dzul, Stabilization of a mini rotorcraft with four rotors. *IEEE Control Syst. Mag.* **25**(6), 45–55 (2005a)
- P. Castillo, R. Lozano, A. Dzul, *Modelling and Control of Mini-Flying Machines*. Springer-Verlag Series in Advances in Industrial Control (Springer, New York, 2005b)
- C.Y. Chen, Multiple degree of freedom inverted pendulum dynamics: modeling, computation and experimentation. Ph.D Dissertation of University of Southern California, 2009
- P. Cruz, R. Fierro, Agile load transportation: safe and efficient load manipulation with aerial robots. *IEEE Robot. Autom. Mag.* **19**(3), 69–79 (2012)
- D. Fusato, G. Guglieri, R. Celi, Flight dynamics of an articulated rotor helicopter with an external slung load. *J. Am. Helicopter Soc.* **45**(1), 2–13 (2001)
- R.H. Hoh, R.K. Heffley, Development of handling qualities criteria for rotorcraft with externally slung loads. NASA report, 2006
- W. MacKunis, Z.D. Wilcox, M.K. Kaiser, W.E. Dixon, Global adaptive output feedback tracking control of an unmanned aerial vehicle. *IEEE Trans. Control Syst. Technol.* **18**(6), 1390–1397 (2010)
- R. Mahony, V. Kumar, P. Corke, Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. *IEEE Robot. Autom. Mag.* **19**(3), 20–32 (2012)
- L. Marconi, R. Naldi, L. Gentili, Modelling and control of a flying robot interacting with the environment. *Automatica* **47**, 2571–2583 (2011)

- B.C. Min, J.H. Hong, E.T. Matson, Adaptive robust control (ARC) for an altitude control of a quadrotor type UAV carrying an unknown payloads. Paper presented at the 2011 international conference on control, automation and systems, Gyeonggi-do, Korea, 26–29 Oct 2011
- K. Peng, G. Cai, B.M. Chen, M. Dong, K.Y. Lum, T.H. Lee, Design and implementation of an autonomous flight control law for a UAV helicopter. *Automatica* **45**(10), 2333–2338 (2009)
- T. Ronen, A Helicopter with a sling load. Ph.D Dissertation of Stanford University, 1985
- M. Ryll, H.H. Bulthoff, P.R. Giordano, Modeling and control of a quadrotor UAV with tilting propellers. Paper presented at 2012 IEEE international conference on robotics and automation, Minnesota, 14–18 May 2012
- A. Tayebi, S. McGilvray, Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Trans. Control Syst. Technol.* **14**(3), 562–571 (2006)
- K. Thanapalan, T.M. Wong, Modeling of helicopter with an under-slung load system. Paper presented at the 29th Chinese control conference, Beijing, 29–31 July 2010
- J.N. Theron, E.P.N. Duque, L. Cicolani, Three-dimensional computational fluid dynamics investigation of a spinning helicopter slung load. NASA report, Document ID: 20070017931, 2005
- Y. Yang, J. Wu, W. Zheng, Variable structure attitude control for an UAV with parameter uncertainty and external disturbance. *Procedia Eng.* **15**, 408–415 (2011)
- D. Zamoski, G. Starr, J. Wood, R. LumiaRapid, Swing-free transport of nonlinear payloads using dynamic programming. *J. Dyn. Syst. Meas. Control* **130**, 041001 (2011)

David H. Scheidt

Contents

51.1 Introduction 1274

51.2 Autonomous Unmanned Air Vehicles 1276

 51.2.1 The Case for Autonomous Systems 1278

 51.2.2 C2 Fundamentals 1278

 51.2.3 The Organic Persistent Intelligence Surveillance and
 Reconnaissance System 1286

51.3 Conclusion 1297

References 1297

Abstract

Motivated by Generals Rommel and Guderian’s innovative command and control techniques used in Europe in 1940, this chapter begins by using information theory to examine unmanned air vehicle (UAV) command and control (C2). The information-theoretic analysis provides a justification and uses cases for *autonomous* UAVs. An autonomous unmanned vehicle system “Organic Persistent Intelligence Surveillance and Reconnaissance” (OPISR) that is designed to duplicate Guderian’s innovations is introduced. OPISR is an autonomous unmanned vehicle system that combines the immediate response to tactical ISR needs provided by organic assets with the time-on-station, minimal logistics provided by persistent unmanned systems. OPISR autonomous vehicles collectively interpret real-time tactical intelligence surveillance and reconnaissance (ISR) objectives submitted by any number of disadvantaged users, gather the required ISR data, and return the needed intelligence directly to the affected user. OPISR is an ad hoc, decentralized system that requires no central base or authority and is capable of functioning in communications-denied environment. The chapter

D.H. Scheidt
Johns Hopkins University Applied Physics Laboratory, Laurel, MD, USA
e-mail: David.Scheidt@jhuapl.edu

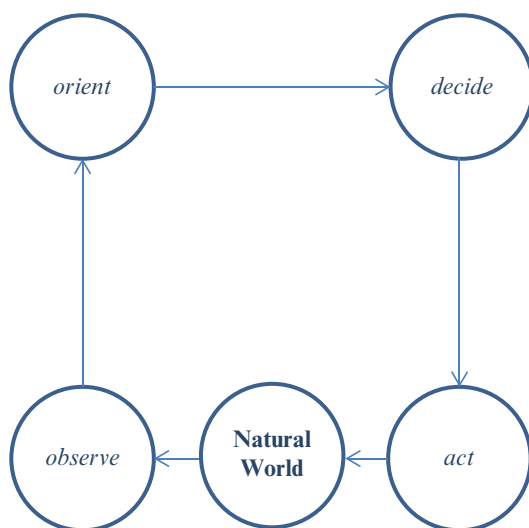
describes a series of experiments including 2011 experiments in which 16 fully autonomous unmanned vehicles, including 9 unmanned air vehicles, were used to simultaneously support mounted, dismounted and maritime users. During these experiments users provided abstract mission-level ISR needs to the “vehicle cloud.” These needs were interpreted by the vehicles, which self-organized and efficiently achieved the user’s objectives.

51.1 Introduction

In the spring of 1940, the combined French, British, Dutch, and Belgian forces outnumbered their German counterparts in troops, mechanized equipment, tanks, fighter planes, and bombers. The ME109E German fighter aircraft was roughly equivalent to the British Spitfire, and the French CharB1 tank was superior to the German Panzer III. In addition, the allies were fighting on their home soil which greatly simplified their logistics. Yet in less than 6 weeks, the Belgians, Dutch, and French surrendered to the Germans, and the British retreated across the English Channel. Even though the allies had superior equipment and larger forces, they were defeated by the Germans who employed *Auftragstaktik*, a command and control technique that enabled “edge” war fighters to directly coordinate on tactical decisions using modern communications equipment (in WWII this was radio). Allied forces were forbidden to use radio because it “was not secure,” and allied maneuver decisions were made by generals at headquarters and based upon hand-couriered reports. German decisions were made on the fly by Panzer III commanders and JU-87 (Stuka) pilots conversing over the radio. By the time the French commanders met to decide what to do about the German advance, General Erwin Rommel and General Heinz Guderian’s Panzers had travelled over 200 miles and reached the English Channel.

As demonstrated repeatedly in military history, including the German advance in 1940, the speed at which battlefield decisions are made can be a deciding factor in the battle. A process model that describes military command and control is the Observe, Orient, Decide, Act (OODA) loop described by Boyd (Fig. 51.1). Boyd shows that, in military engagements, the side that can “get inside the opponent’s OODA loop” by more rapidly completing the OODA cycle has a distinct advantage. In their influential book *Power to the Edge*, Alberts and Hayes use the term agility to describe an organization’s ability to rapidly respond to changing battlefield conditions. Modern warfare case studies, such as the Chechens against the Russians, and not-so-modern warfare, such as Napoleon at Ulm, indicate that agile organizations enjoy a decisive military advantage. Alberts points out that a common feature of agile organizations is an empowerment of frontline forces, referred to as “edge” war fighters. Commanders facilitate organizational agility by exercising “command by intent,” in which commanders provide abstract goals and objectives to edge war fighters who then make independent decisions based upon these goals and their own battlefield awareness. This empowerment of edge war fighters reduces the OODA loop at the point of attack, providing the desired agility.

Fig. 51.1 Boyd's Observe, Orient, Decide, Act (OODA) cycle models the military decision-making process. Military organizations that perform their OODA cycle more rapidly than opponents gain a substantial competitive advantage



A distinguishing characteristic of the conflicts in Afghanistan and Iraq is the explosive growth in the use of unmanned air vehicles. Between the first and second Gulf Wars, unmanned vehicles transitioned from a novelty item to an indispensable component of the U.S. military. Field deployable organic unmanned air vehicles such as the AeroVironment Raven are essential equipment for the modern war fighter.

The agility provided by field deployable vehicles comes at a cost, as the use of field deployable units increases logistics and workload demands on frontline forces. When compared to larger unmanned vehicles, field deployable units such as the Raven (Fig. 51.2) offer limited sensing and time-on-target capabilities. Medium-sized vehicles, such as the Boeing-Insitu ScanEagle and AAI Shadow, offer longer time on station and more capable payloads. Medium-sized vehicles also do not make logistics or workload demands on the edge war fighter. Large unmanned vehicles such as the General Atomics Reaper and Northrup Grumman Triton offer still more capable payloads and increased time on station and also do not increase edge war fighter logistics or workload. However, providing timely edge war fighter access to intelligence products produced by medium and large vehicles is a challenge because medium- and large-sized unmanned air vehicles produce massive amounts of data that is difficult to process and disseminate from centralized command posts. In fact, as reported by Ariel Bleicher, "In 2009 alone, the U.S. Air Force shot 24 years' worth of video over Iraq and Afghanistan using spy drones." The trouble is there aren't enough human eyes to watch it all. The deluge of video data from these unmanned aerial vehicles, or UAVs, is likely to get worse. A single Reaper drone can record 10 video feeds at once, and the Air Force plans to eventually upgrade that number to 65. John Rush, chief of the Intelligence, Surveillance and Reconnaissance Division of the U.S. National



Fig. 51.2 An AeroVironment Raven being launched

Geospatial-Intelligence Agency, projects that it would take an untenable 16,000 analysts to study the video footage from UAVs and other airborne surveillance systems. The intelligence, surveillance, and reconnaissance (ISR) capability represented by medium- and large-scale unmanned vehicles represents a tremendous potential for the edge war fighter if only the information could be processed and distributed in time. For the edge war fighter to take advantage of the ISR capability represented by these assets, information relevant to that specific war fighter must be gleaned from the mass of information available and presented to the war fighter in a timely manner. This presents a challenge as crews analyzing UAV payload data (far fewer than Rush's 16,000 analysts) are not apprised of the changing tactical needs of all war fighters, nor do the war fighters have access or the time required to select and access data from UAV sources. Currently, operation centers are used to gather and disseminate information from persistent ISR assets. This centralized information management process introduces a delay between the observation and transmission to the war fighter which reduces force agility and operational effectiveness. While U.S. soldiers are empowered to operate on "command by intent," their ISR systems are all too frequently centralized systems reminiscent of the French command structure. For U.S. forces to become a fully agile force, the ISR systems supporting the U.S. soldier must be as agile as the soldier it supports. Agile unmanned vehicle systems require that some decisions are made at the edge nodes; therefore to become agile, unmanned vehicles must become autonomous.

51.2 Autonomous Unmanned Air Vehicles

UAVs currently in use are described as unmanned strictly because no human rides inside the air vehicle; however, the manual labor required to operate an air vehicle

remains largely unchanged as ground-based UAV pilots perform similar tasking to airborne pilots. Because flight procedures have not evolved to match the removal of the human from the air vehicle, the manpower required to operate an UAV equals or exceeds the manpower required to operate a manned aircraft. Because UAV pilots, by definition, fly the vehicle remotely, there is no longer a requirement that a pilot be co-located with the UAV area of operations, and it is not uncommon particularly for large, expensive UAVs for the UAV pilot to operate the vehicle from an office-like environment thousands of miles from the operating area. As the size, range, and capability of a UAV diminish, its use becomes more tactical, and the UAV pilot is located closer to the operating area with medium-sized UAVs such as the Boeing ScanEagle and AAI Shadow being controlled from forward operating bases and the AeroVironment Raven being controlled by a tactical unit in the field. Clearly remote vehicle operations significantly reduce pilot risk when compared to manned aircraft, yet this risk reduction comes at the cost of tactical awareness and involvement. Remote UAV pilots, when emplaced in a hierarchical command structure, have a reduced ability to assist in agile operations; without being immersed in the environment and able to, when required, communicate with actors outside of the current command structure, UAV pilots cannot support tactical operations as well as a JU-87 pilot from 1940.

Automated aircraft control techniques are emerging that allow UAV designers to design and field UAVs with varying degrees of autonomy. For the purpose of discussion, three general levels of autonomy are defined: tele-operation, which is essentially no automation at all; automatic, in which UAVs perform simple actions that can be fully enumerated and tested during design; and autonomy, in which the vehicle independently devises a course of action in response to complex operating conditions. The distinction between automatic and autonomy is subtle but important. Most UAVs being used today are automatic. Engineers of automatic UAVs have designed in action-based commands for pilot use (e.g., follow this path, loiter here, land there), the response matrix has been exhaustively enumerated and tested by development engineers, and the task of managing uncertainty, and complexity, lies with the pilot. Autonomous UAVs, which are not currently in service, are capable of devising a course of action in response to a complex, uncertain situation that was not, in detail, examined by the engineer during design or by the pilot. In other words, an automatic UAV is capable of following a path; an autonomous UAV is capable of finding a path.

The technology required to build autonomous UAVs is available today. Prototype autonomous UAVs that exhibit a variety of complex autonomous behaviors have been developed at academic institutions and industry research centers. Demonstrated capabilities involve relatively simple tasks including search, interdiction, tracking, obstacle avoidance, path planning, logistics, communications, launch, and recovery. While vehicle tasking remains simple, the complex unpredictable nature of the operating environment represents a complex problem requiring an autonomous, not automatic, solution. The bulk of autonomous UAV efforts are conducted at a relatively low technology readiness level (TRL) using low-cost rotorcraft in a laboratory (Kumar and Michael 2012; Bethke et al. 2008). More mature demonstrations

of autonomous tier 1 and tier 2 UAVs have been demonstrated outdoors at U.S. government ranges (Scheidt et al. 2004; Kwon and Pack 2011; Tisdale et al. 2008). Arguably the most mature autonomous UAV effort is the DARPA Heterogeneous Airborne Reconnaissance Team (HART) program, providing automatic unmanned air vehicles with an autonomous tactical decision aide (TDA). In these systems simple actions such as waypoint following are performed independently by each UAV, and complex decisions are performed by the TDA algorithm located on the pilot's computer. Prior to execution, the pilot reviews and approves (or disapproves and modifies) the plan. Proponents of HART correctly argue that the pilot is allowed to "auto-approve" plans, effectively making HART an autonomous system; however, the requirement to continually enable the pilot to review and approve all flight plans profoundly impacts the UAV system architecture and performance by delaying the exchange of information between the sensor and the UAV controller. The U.S. Army planned on partial fielding of HART in 2012 (Defense Systems staff 2012).

51.2.1 The Case for Autonomous Systems

Technological availability does not mean that autonomous UAVs will, or should, be used in the field. Autonomous UAV use requires that, when compared to manned aircraft or tele-operated/automatic UAVs, autonomous UAVs provide some tangible benefit to the organization deploying the autonomous UAV. Three general benefits are commonly offered that could justify the use of autonomous UAVs: first, by reducing the manpower required to fly/operate the UAV, autonomous vehicles are less expensive to fly; second, because autonomous UAVs do not require constant communications with a base, they are less vulnerable to electronic warfare attacks and capable of electromagnetic stealth; and third, by making better, more timely decisions, autonomous UAVs can, in certain circumstances, provide more effective performance. This third argument that autonomous unmanned vehicles can improve mission performance and that this performance can be understood and predicted by viewing autonomy as a command and control technique is the central theme of this chapter.

51.2.2 C2 Fundamentals

In *Power to the Edge*, command and control (C2) is defined as the "common military term for management of personnel and resources" but also gives the formal definition of command as found in the Joint Chiefs of Staff Publication, which subsumes some portions of control in that definition. Viewed as a black box the purpose of the C2 system use observations to produce decisions. C2, including UAV C2, involves the production and execution of decisions that, when executed, change the world in which the UAV is operating in ways that benefit the operator. That world (X) is described as a set of states, $X = \{x_1, x_2, \dots, x_n\}$, each one of which represents a unique configuration of actors and attributes within the natural world in which

the UAV operates. The command and control system can be viewed as a transfer function ($f(x_i) \rightarrow x_j$) that produces a state change in the world. The “quality” of each state can be determined by applying a mission-based fitness criteria to elements within the state. For example, if a UAV mission is to track a target, then states in which the target is within the field of view of the UAV’s sensor are evaluated as of higher quality than those states in which the target is not seen by the UAV, and an effective C2 system would cause state transitions that are of high quality when compared to alternative transitions. C2 is a constant battle between chaos and order, with order being state transitions designed to achieve mission goals that are instigated by the C2 system and chaos being unanticipated state transitions produced by adversaries, poorly coordinated teammates or random acts.

C2 can be best understood as an information-theoretic problem. This is apropos as both the situational awareness upon which decisions are based as well as the decision products can be viewed as information-theoretic messages and both the C2 process and the natural world can be viewed as information transfer functions. For an interesting, albeit somewhat off-topic, discussion on the information-theoretic nature of physics, see Wheeler (1990).

The information content (S) of a message (m) is defined by Shannon (1948) as:

$$S(m) = \log_2(1/P(m)) \quad (51.1)$$

The more improbable the message being received, the larger the information content. For example, barring a highly improbable change to celestial mechanics, a message stating “tonight it will be dark” has zero information content because the a priori probability that it would be dark is one. By comparison, “tonight it will rain” has positive information content because the a priori probability that it would rain during a given evening is less than one. Information content is measured in bits, which are real numbers. Note that the term “bit” is overloaded, and information theory bits are not the same as computer science bits, which are integers.

The *state space* (p) of the world in which our UAVs operate is the amount of bits required to uniquely identify all possible states in the world, which is referred to as the *state space* of the world.

$$p = \log_2 |X| \quad (51.2)$$

Completely describing the world requires a message of length p . A communication that completely describes the natural world would require a message whose length would be effectively infinite as the natural world includes each blade of grass, molecule of air, quantum states of ions within each atom, and so forth. Fortunately, effective C2 of UAVs does not require such detailed knowledge, and effective UAV control can be provided using artifacts that are abstract, limited, and while potentially quite large in number, expressible in messages that are small enough to be used within a modern computer network. In fact, military C2 systems routinely express tactical “worlds” in finite languages such as the protocols used by the *Global Command and Control System* and the *Link16* network. In practice, the size of a UAV’s world as represented by the C2 system can be dynamic, with the state

space of the tactical world changing as artifacts enter, or leave, the operational area. That the complexity of a UAV's world, as represented by the state space of the current situation, is subject to change is important to understanding how to control UAVs.

Knowledge of the UAV's world is rarely complete, and UAVs are expected to operate in the presence of varying degrees of uncertainty. Uncertainty in tactical information can be produced by errors in sensor systems, gaps in sensor coverage, or approximation error, which is the difference between the actual ground truth and the coding scheme selected. For example, if the unit representation of a coding scheme used to represent linear position is 1 m, then an approximation error of 0.5 m is unavoidable. Consider the message (m) that encapsulates a C2 systems' current situational awareness (SA). When $S(m) \neq p$, uncertainty exists and additional information is required to produce complete SA. Now consider a second message (m') that contains all of the missing information required to reduce uncertainty to zero. The information content of m' is information entropy (H) of m , shown as

$$S(m') = H(m) = \sum P(x) \log \frac{1}{P(x)}$$

where x is the assemblage of information contained in m (51.3)

The information content of m is "negentropy" (N) or "order," Command and control is a battle between the forces of order and chaos, in which the command and control system seeks to generate order by forcing the world into the most beneficial state for the commander and the uncontrollable forces within the world, which may include adversaries, continuously generates entropy that moves the world away from the commander's ideal state.

Together, the information content of m and m' represents the total amount of information potential within the system s.t:

$$S(m') = p - S(m) \quad (51.4)$$

For each bit of order produced by the C2 system, entropy is reduced by a bit. Likewise, each bit of entropy produced by unanticipated change reduces order by 1 bit.

As demonstrated by Guderian and Rommel, C2 systems are temporally sensitive. As time elapses the information content of a message that describes a dynamic scene decreases in proportion to the unpredictable change in the scene. An example of this are unexpected maneuvers made by a target after a sensor observation was made but before the execution of an response to the observation. The loss of information over time is defined as *entropic drag* (Γ) which is expressed mathematically by Scheidt and Schultz (2011) as:

$$\Gamma(x, t, t_0) = \frac{H(x(t) | x(t_0))}{t - t_0} \quad (51.5)$$

Note that entropic drag is specific value for each state. Control of UAVs operating in an uncertain world in which multiple states are feasible requires a consideration of all admissible states. The measurement of entropic change that incorporates all feasible states is the normalized form of entropic drag (Γ_{norm}) that is the average entropic drag for all states within the system:

$$\Gamma_{norm}(t) = \sum_{\forall x_i} \frac{H(x_i(t) | x_i(t_0))}{(t - t_0) \cdot |X|} \quad (51.6)$$

UAVs acquire, process, and share information over a control infrastructure that includes onboard network and processing, radio downlinks and uplinks, and off-board processing. When processing information, two key characteristics of the control infrastructure are *latency* (δ_0), which is the unavoidable delay in processing an information packet regardless of packet size, and *bandwidth* (β), which is the rate in bits per second at which bits of information can be processed irrespective of the latency. The delay time (δ) required to process a message m of information can be viewed as

$$\delta_m = \delta_0 + \left(\frac{\text{size}_m}{\beta} \right) \quad (51.7)$$

When communicating information from a sensor to a user over a communications link, large amounts of information take more time to transmit and process than small amounts of information. If the information observed by the sensor is a dynamic scene, which is often the case for UAVs, increases in information gathering cause an increase in processing delays that, in turn, cause an increase in entropy. This presents a paradox with respect to attempts to increase information content by increasing the quantity of data. This paradox is defined by the relationships between Eqs. (51.5) and (51.7) that was described by Scheidt and Pekala (2007) and is shown in Fig. 51.3. The figure shows uncertainty as a function of the unit resolution used to describe a dynamic scene using a constant cognitive bandwidth and also assuming that all sensor data is correct. In the plot the highest unit resolution (e.g., least precision) is shown on the right, while the smallest unit resolution (most precision) is shown on the left. Interpreting the plot from right to left shows, as one might expect, that initially increasing the information content of a message substantially reduces uncertainty. As precision increases, the time requires to communicate and process the information increases, which increases information loss due to the entropic drag. Eventually an uncertainty minima is reached at which the information gain from additional information content within the initial message equals the information loss due to entropic drag. Any attempt to use additional precision beyond the minima produces net loss in information content.

If the function of the UAV is intelligence, surveillance, and reconnaissance (ISR), then mission goals are to provide information on targets of interest with minimal uncertainty. Counterintuitively, transmitting all possible information on a target may not be the best approach for an ISR system. The plot in Fig. 51.3 provides a guide as to the optimal amount of data that should be gathered, processed, and

Fig. 51.3 When information describing a dynamic (changing) scene is sent over, a network uncertainty can be minimized by using the optimal resolution (shown by the local minima). The local minima shown is the balance point where the rate of information gain equals the rate of entropy (entropic drag)

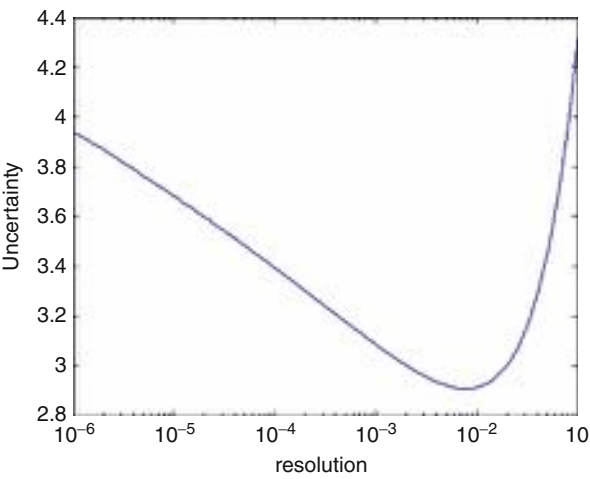


Table 51.1 The complexity and rate of unpredictable change vary by mission class

| Use case | Description | Complexity, $P(x)$ | Entropic drag, $\Gamma(t)$ |
|--------------------|---|--------------------|----------------------------|
| Strategic | Strategic missions are dedicated to acquiring information on standing infrastructure | – | None |
| Operational | Operational missions are dedicated to acquiring information on the general movement and condition of large units (e.g., company size or larger) | Low | Low |
| Focused tactical | Small units (individuals, squads, or platoons) performing tightly defined, focused missions in isolation | Low | High |
| Multiunit tactical | Small units operating as part of a large engagement that involves multiple units | High | High |

provided from a UAV to the UAV operator. The local minima in the plot represents the optimal amount of information that should be transmitted. UAV system that transmit too much information, represented by high-resolution data on the left of the graph, reduces the net information provided due to the large loss of information across all data due to entropic drag. To further understand this relationship, let us examine bandwidth from Eq. (51.7) and its relationship to Eqs. (51.5) and (51.6) in more detail. This is accomplished by viewing canonical ISR missions and operating conditions through information theory and applying these views to differing forms of UAV control. Table 51.1 describes four general classes of ISR missions for UAVs and defines, in general terms, the complexity and entropic drag associated with those missions.

Recall that UAVs may be controlled using three general methods: tele-operation, automatic control, and autonomous control. Descriptions of these methods are:

- Tele-operation – The most common form of UAV control is tele-operation. Tele-operated UAVs use a ground-based pilot to fly the UAV using techniques that are identical those of human-piloted aircraft. Tele-operated UAVs utilize low-level control features found in an airplane cockpit as well and onboard sensor data on the pilot's ground station that are duplicates of those found in the cockpit of a piloted aircraft. All decisions used to control tele-operated planes are made by the ground-based human pilot.
- Automatic flight – UAVs that contain autopilots are capable of *automatic* flight. Automatic UAVs use autopilots to assure stable-controlled flight. The pilots of automatic UAVs provide waypoint locations that direct the path of the UAV. In addition to flying to pre-defined waypoints, automatic UAVs may be preprogrammed to handle simple changes in operating conditions; however, management of complex or unanticipated changes are handled by a ground-based pilot.
- Autonomous flight – UAVs that contain an autopilot and an onboard, intelligent controller are capable of *autonomous* flight. Similar to automatic UAVs, autonomous UAVs provide stable flight between waypoints; however, unlike automatic UAVs, the intelligent controllers onboard an autonomous UAVs define new waypoints in response to unanticipated changes in operational conditions.

Autonomous UAVs fundamentally change the relationship between the human and the UAV because autonomous UAVs, unlike tele-operated UAVs, automatic UAVs, and manned aircraft, do not require pilots. Autonomous UAVs do use human supervision; however, that supervision is performed at a higher level than the typical plane-pilot relationship. The relationship between an autonomous UAV and the supervising human resembles the relationship between a human pilot and an air traffic controller or, for Navy pilots, the Air Boss. Autonomous UAV operators supervise their UAVs by providing abstract, mission-level objectives as well as rules of engagements that are equivalent to the instructions provided to human pilots prior to a mission. During the mission autonomous UAVs devise a course of action aligned with these instructions in response to the current situation. As conditions change during the course of a mission, an autonomous UAV constantly modifies the course of action in accordance with the mission-level objectives. Unlike automatic UAVs, autonomous UAVs respond to complex situations that were not explicitly considered during UAV design.

The different forms of UAV control demand different levels of communications. Direct control of UAV control surfaces requires that tele-operated UAV pilots use low-latency, high quality of service communications to command UAVs. For tele-operated UAVs even small perturbations in service run the risk of loss of control and catastrophic failure. Automatic UAVs are more forgiving, as the pilot is only required to provide guidance at the waypoint level. The communications requirement for automatic UAVs is determined by the rate of change of those operational elements that dictate the mission pace. For example, if the UAV is engaged in an ISR task to track a specific target, the UAV communications

infrastructure used to control the UAV must be capable of providing target track data to the pilot prior to the target exiting the UAV's field of view. Depending upon the nature of the target, the response time required for automatic UAVs can range from sub-second intervals to minutes. Autonomous vehicles are the most forgiving of communications outages and delays. In fact, autonomous vehicles are capable of performing without communication to human operators during an entire mission. The ability to function without continual human supervision changes operator and designer perspectives on UAV communications from being a requirement to being an opportunity. When autonomous UAVs can communicate, either to humans or other vehicles, mission performance is improved by the sharing of information and collaborating on decisions; however, when communications are not available, autonomous UAVs are still capable of fulfilling the mission. A synopsis on the communications availability, in terms of latency and bandwidth, for four different classes of conditions is provided in Table 51.2.

When considering whether a UAV control decision should be made by a human operator or by a control processor onboard a UAV, three criteria should be considered: (1) what is the quality of decisions made by the human/machine, (2) what are the ethical and legal requirements for making decisions, and (3) what accessible information is available to the human and the machine? There are ethical and legal advantages and disadvantages for both human and machine decision-making. Arkin provides an excellent overview of the legal and ethical issues, concluding that no consensus exists that would favor human control over machine control or vice versa (Arkin 2009). Regarding the ability to make higher-quality decisions, anecdotal evidence suggests that there exist problem sets for which humans provide better quality decisions and problem sets for which intelligent control algorithms provide better decisions. For example, few would argue that the path planning algorithms provided by Mapquest and Google find superior paths over complex road networks in times unmatched by humans. On the other hand, even the most sophisticated pattern recognition algorithms are incapable of matching small children in rapidly identifying and manipulating common household items in a cluttered environment. The neuroscience and psychology communities have long-studied human cognitive abilities, and computer science, particularly the subfield of complexity theory (Kolmogorov 1998), has been used to study the performance of cognitive algorithms as a function of the problem space being addressed; however, a comparative understanding between human and machine cognition (or even the tools necessary to achieve this understanding) does not exist at this time.

In order to move the discussion of UAV C2 into a manageable space, two simplifying assumptions are asserted: first, decisions should be made using the maximum amount of information, and (51.2) given equivalent information, it is preferred that decisions be made by a human. These simplifying assumptions allow us to focus on the availability of information as the primary driver for command and control. While it is somewhat disconcerting to ignore the quality of the decision-maker and ethical issues, our focus on information as the driving factor in C2 is consistent with the lessons learned from Guderian and Rommel earlier.

Table 51.2 Communication availability experienced by UAVs during a mission can vary greatly. Depending upon the operational conditions, latency and bandwidth can vary greatly

| Communications availability | Description | Latency (δ_0) | Bandwidth (β) |
|-----------------------------|--|------------------------|-----------------------|
| Dedicated communications | Dedicated infrastructure whose access is tightly controlled and not contested by environmental or adversarial activities | Very low | Extremely high |
| Uncontested broadband | Standing infrastructure that is broadly used that is not contested by environmental or adversarial activities | Very low | High |
| Contested communications | Standing infrastructure that is broadly used and is not contested by environmental or adversarial activities that produce periodic outages and/or reduction in service | Low | Low |
| Over the horizon | Operations that involve periodic movement in areas that are beyond communications range or involve periodic, planned communications blackouts | High | Moderate |

Having narrowed our focus into the information used to make a decision, three information-theoretic distinctions using between an operator and an onboard computer to determine a course of action are identified. These distinctions are as follows: (a) the complexity of the scene that must be described to make a decision, which dictates the size of the packets that must be communicated and the run time of decision processes; (b) the entropic drag of the system being represented by the data, which dictates the time for which the information is valid; and (c) the communications delay time provided in Eq. (51.7) which defines the earliest time at which the decision could be made. These values may be combined to form an information value $g(x)$ that defines the UAV control problem. The information value is defined as the product of the complexity of the UAV’s world and the entropic drag of the world’s unpredictable change divided by the delay associated with communicating the world state to the decision-maker s.t:

$$g(x) = \frac{P(x)\Gamma(t_m)}{\delta_m(x)}$$

(51.8)

The information value correlates to the utility of the autonomous, automatic, and tele-operated controls approaches. When the world the UAV operates in provides an information value that is high autonomous control dominates, when the world the UAV operates in provides an information value that is low tele-operated control dominates and automatic control is preferred in the midrange. Mapping this relationship to the UAV use cases and communications conditions

Table 51.3 The appropriate conditions for using tele-operated, automatic, or autonomous UAV control are defined by the operational criteria and available communications

| Dominant control technique | Dedicated communications | Uncontested broadband | Contested broadband | Over the horizon |
|----------------------------|--------------------------|-----------------------|---------------------|------------------|
| Strategic | Tele-operated | Tele-operated | Automatic | Automatic |
| Operational | Tele-operated | Tele-operated | Automatic | Autonomous |
| Focused tactical | Tele-operated | Tele-operated | Automatic | Autonomous |
| Multiunit tactical | Autonomous | Autonomous | Autonomous | Autonomous |

defined earlier, operational scenarios that are appropriate for autonomous, auto-
matic, and tele-operated UAV control are defined and enumerated in Table 51.3.
As Table 51.3 indicates, there exist real-world circumstances in which UAV C2
should be tele-operated, automatic, or autonomous. Not surprisingly, those times
which autonomous C2 is dominant are complex, dynamic situations. Exactly the
sort of situation faced by Guderian and Rommel. So, having identified that complex,
constantly changing scenarios is best supported by autonomous UAVs, how might
we develop such as system.

**51.2.3 The Organic Persistent Intelligence Surveillance and
Reconnaissance System**

OPISR autonomous UAVs utilize a software and communications subsystem that
is designed to support the rapid, autonomous movement of information across
a tactical force. Commander and/or operators interact with OPISR as a system.
When using OPISR, war fighters connect into the OPISR “cloud,” task OPISR with
mission-level ISR needs and are subsequently provided with the intelligence they
need (Fig. 51.4). This capability provides intelligence directly to the war fighter
without requiring the war fighter to personally direct or, even know about, the
OPISR assets gathering the information. OPISR is autonomous. As a system, OPISR
seeks out relevant information, pushing key tactical information directly to impacted
soldiers in real time. OPISR is capable of rapidly managing large complex, dynamic
situations because it utilizes a decentralized, ad hoc organizational structure.
Systems that use decentralized structures such as OPISR are known to be more
effective at the timely coordination of complex systems (Scheidt and Schultz 2011).
OPISR tracks the location and ISR needs of all blue forces, maintaining a contextual
awareness of the war fighter’s current tactical needs.

As relevant tactical information becomes available, OPISR presents it directly to
the war fighter through an intuitive handheld device. The information requirements
that are used to determine information relevance are defined by the war fighter
through the same handheld interface. This interface supports abstract queries such as
(1) patrol these roads, (2) search this area, (3) provide imagery of a specific location,
(4) track all targets of a specific class on a specific route of location, or (5) alert me
whenever a threat is identified within a certain distance of my location. Information

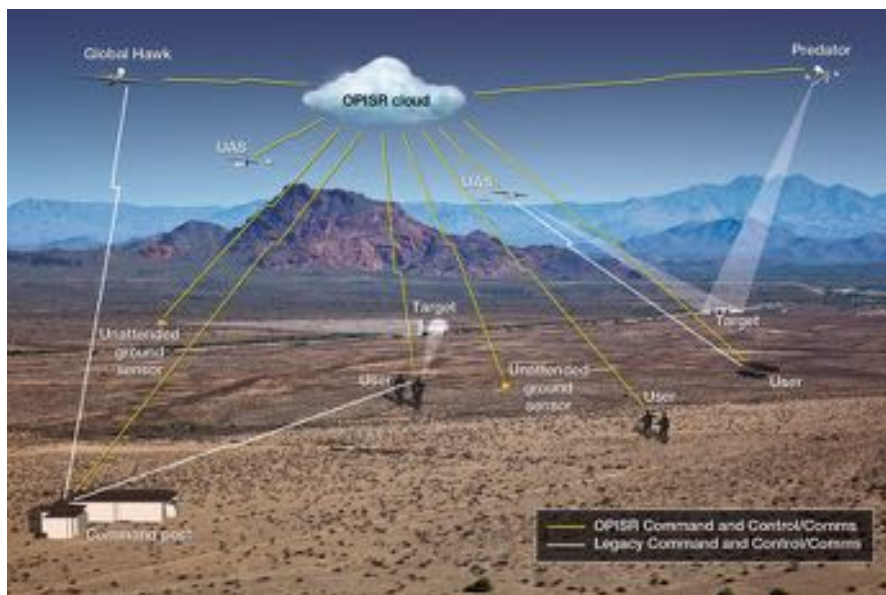


Fig. 51.4 OPISR's concept of operation allows UAVs of various sizes to communicate with each other, with users, and with commanders through an ad hoc, asynchronous cloud. The cloud communicates goals from users to vehicles and sensor observations from users to vehicles

that matches these queries is sent by the system to the handheld device. The handheld interface provides a map of the surrounding area that displays real-time tracks and detections and imagery metadata. The imagery metadata describes, at a glance, the imagery available from the surrounding area. OPISR-enabled vehicles are autonomous; if the information required by the war fighter is not available at the time the query is made, OPISR unmanned vehicles autonomously relocate so that their sensors can obtain the required information. OPISR-enabled unmanned vehicles support multiple war fighters simultaneously, with vehicles self-organizing to define joint courses of action that satisfy the information requirements of all war fighters.

Because war fighters are required to operate in harsh, failure-prone conditions, OPISR was designed to be extremely robust and fault tolerant. OPISR's designers viewed communications opportunistically, designing the system to take advantage of communications channels when available but making sure to avoid any/all dependencies on continual high quality of service communications. Accordingly, all OPISR devices are capable of operating independently as standalone systems or as ad hoc coalitions of devices. When an OPISR device is capable of communicating with other devices, it will exchange information through networked communications and thereby improve the effectiveness of the system as a whole. However, if communications are unavailable each device will continue to perform previously

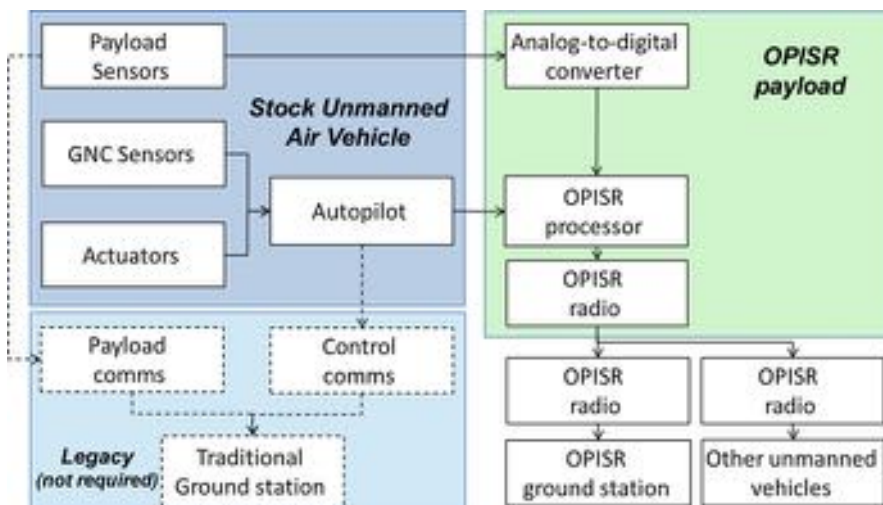


Fig. 51.5 OPISR's hardware architecture is based on a modular payload that can be fitted onto different types of unmanned vehicles

identified tasks. When multiple devices are operating in the same area, they will self-organize to efficiently perform whatever tasks have war fighters have requested.

51.2.3.1 OPISR Hardware

Off-the-shelf unmanned vehicles and unattended sensors can be incorporated into the OPISR system by adding the OPISR payload. As shown in Fig. 51.5, the OPISR payload consists of three hardware components: an OPISR processor that executes the OPISR software, an OPISR radio that provides communications to other OPISR nodes including OPISR's handheld interface devices, and an analog to digital converter that is used to convert payload sensor signals into digital form. Unmanned vehicles that have an onboard autopilot capable of providing stable flight can be modified to become *autonomous* vehicles by connecting the autopilot to the OPISR processor. When the vehicle is operating autonomously, the autopilot sends guidance and control (GNC) telemetry to the OPISR processor. The processor using the GNC data to devise a continual stream of waypoints are sent to the autopilot to follow. The OPISR processor also uses the GNC telemetry to produce metadata that is associated with the sensor data. The combined sensor data and metadata is then used by the OPISR system as a whole.

In service unmanned vehicles frequently use separate communications channels for control and imagery. Since OPISR devices perform both image processing and control onboard the device, these communications channels, and the traditional pilot-ground station, are no longer required. Effectively OPISR devices are capable of operating fully independent of direct human supervision. Note that OPISR devices are still responding to war fighter requests; however, these devices accomplish this without requiring continual communications with the war fighter

being serviced. While OPISR does not require traditional control and payload communications, OPISR devices do support these legacy capabilities. Because the OPISR nodes communicate over a separate channel, OPISR functionality may be provided in tandem with traditional control. This is in keeping with the OPISR dictum that OPISR is an entirely additive capability; unmanned vehicle owners lose no functionality by adding OPISR. However, OPISR vehicles are responsive to commands from human operators and will, at any time, allow an authorized human operator to override OPISR processor decisions. Likewise, legacy consumers of information will still receive their analog data streams. Note that even when the OPISR processor is denied control by the UAV pilot, the OPISR system will continue to share information directly with edge war fighters as appropriate.

51.2.3.2 OPISR Software

OPISR is based upon a distributed multi-agent software architecture. Each software agent serves as a proxy for the device on which it is located, and all devices within OPISR have their own agents including unmanned vehicles, unattended sensors, and user interfaces. Each agent is composed of four major software components (Fig. 51.6): a distributed blackboard, which serves as a repository for the shared situational awareness within the agent system; an agent communications manager, which manages the flow of information between agents; a cSwarm controller, which determines a course of action for those devices that are capable of autonomous movement; and a payload manager, which manages the sensor information from the device's organic sensors. All devices within the system, including the war fighter's handheld device, are peers within OPISR.

51.2.3.3 Distributed Blackboard

In the 1980s, Nii (1986) described a method for multi-agent systems to communicate between each other in an asynchronous manner called a blackboard system. Like its namesake in the physical world, blackboard systems allow agents to post messages for peer agent consumption at an indeterminate time. Each OPISR agent contains a personal blackboard system that maintains a model of the agent's environment. Three types of information are stored on each agent's blackboard: beliefs, metadata, and raw data. Raw data is unprocessed sensor data from a sensor within the OPISR system. Metadata is information that provides context to a set of raw data including sensor position, pose, and time of collection. Beliefs are abstract "facts" about the current situation. Beliefs include geo-spatial artifacts such as targets, blue force locations, or search areas. Beliefs can be developed autonomously from onboard pattern recognition software and data fusion algorithms or asserted by humans. Mission-level objectives, the goals that drive OPISR, are a special class of belief that must be produced by a human. The storing and retrieval of information to and from agent blackboards is performed by the blackboard manager. The blackboard manager accepts stores and retrievals from sensors onboard the agent's device, other agents, or pattern recognition/data fusion software contained within the agent. The integrity of the data stored on the blackboard is maintained by a truth maintenance system (TMS). The TMS performs two functions. First, the TMS resolves conflicts between beliefs. The simplest form of conflict resolution is

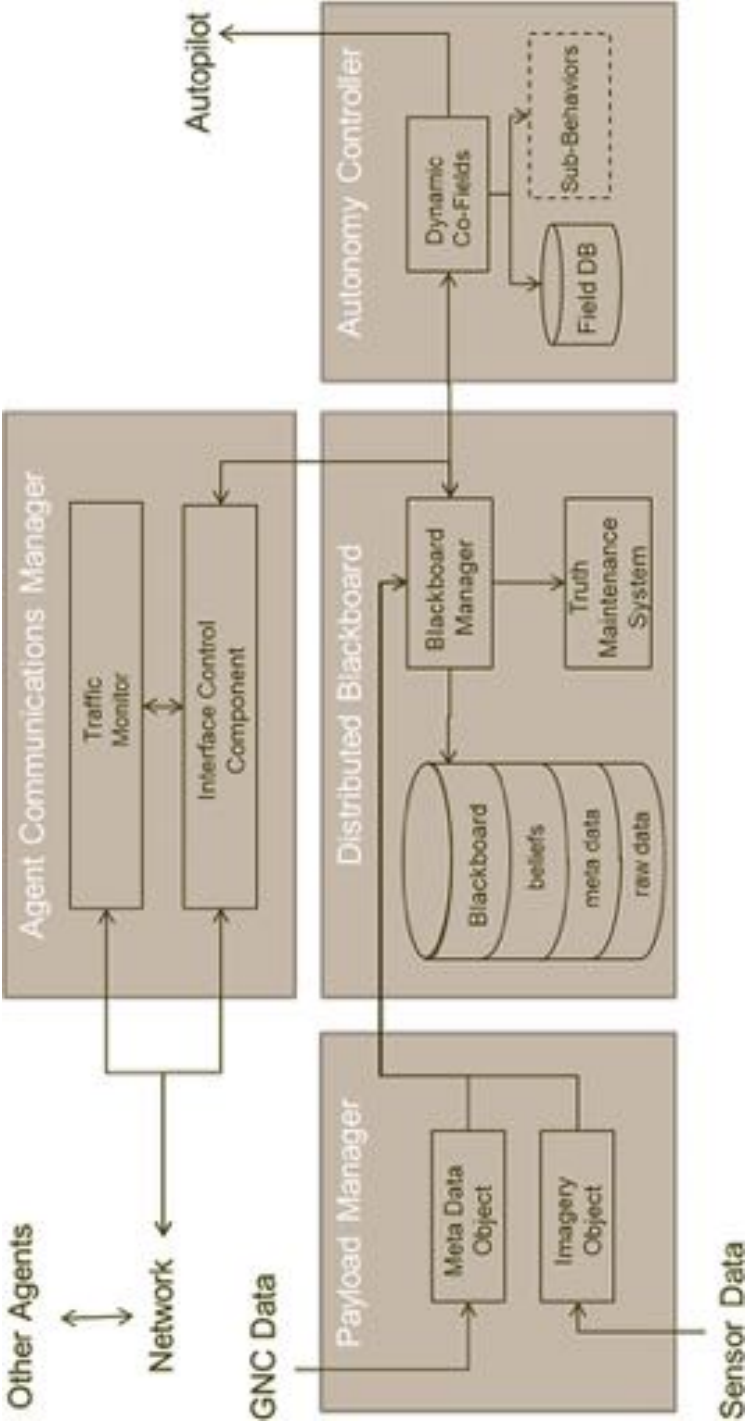


Fig. 51.6 Each OPISR node contains a four major software components that manages information flow and decision-making

accomplished by storing the belief with the more recent time stamp. For example, one belief might posit that there is a target at grid $[x, y]$ at time t_0 , and a second belief might posit that there is no target at grid $[x, y]$ at time t_1 . More sophisticated conflict resolution algorithms are scheduled to be integrated into OPISR in 2012. The second TMS function is the efficient storage of information within the blackboard. When performing this task, the TMS caches the most relevant timely information for rapid access, and when long-lived systems generate more data than can be managed within the system, the TMS removes less important information from the blackboard. For caching and removal, the importance of information is defined by the age, proximity, uniqueness, and operational relevance.

Coordination between agents is asynchronous, unscheduled, and completely decentralized, as it has to be, for any centralized arbiter, or scheduled communications introduce dependencies that reduce the robustness and fault tolerance that is paramount in the OPISR design. Because agent communication is asynchronous and unscheduled, there is no guarantee that any two agents will have matching beliefs at an instance of time. Fortunately, the control algorithms used by OPISR are robust to belief inconsistencies. Cross agent truth maintenance is designed to the same criteria as agent-agent communications: *Information exchanges between agents seek to maximize the consistency of the most important information but does not require absolute consistency between agent belief systems.* Information exchange between agents is performed by the agent communications manager. When communications are established between agents, the respective agent communications managers (ACM) facilitate an exchange of information between their respective blackboards. When limited bandwidth and/or brief exchanges limit the amount of information exchanged between agents, each ACM uses an interface control component to prioritize the information to be transmitted. Information is transmitted in priority order with priority being determined by information class (beliefs being the most important, followed by metadata), goal association (e.g., if a war fighter has requested specific information that information is given priority), timeliness, and uniqueness.

51.2.3.4 Autonomous Control

OPISR's autonomous unmanned vehicles use dynamic co-fields (DCF), also known as stigmergic potential fields, to generate movement and control actions. DCF is a form of potential field control. Potential field control techniques generate movement or trigger actions by associating an artificial field function with geo-spatial objects. In OPISR, the objects that are used to derive fields are beliefs. Fields represent some combination of attraction and/or repulsion. By evaluating the fields for all known beliefs at a vehicle's current location, a gradient vector is produced. This gradient vector is then used to dictate a movement decision. Developed in 2003 (Scheidt et al. 2005), DCF extends an earlier potential field approach called co-fields (Mamei et al. 2002) by making the potential fields used dynamic with respect to time and also making vehicle fields self-referential. Self-referential fields are fields that induce vehicle decisions that are generated by the vehicle's own presence. Adding these dynamic qualities is key to managing two well-known problems with potential fields

approaches: namely, the tendency of vehicles to become stuck in local minima and the propensity to exhibit undesired oscillatory behavior. As implemented in OPISR, DCF is used to effect specific behaviors such as search, transit, or track, as well as behavioral selection. The DCF algorithm is encoded in the cSwarm software module. All unmanned vehicles in OPISR execute cSwarm. DCF behaviors specific to unique classes of vehicle are produced by tailoring the field formula which is stored in a database within cSwarm. OPISR autonomous unmanned vehicles is a variety of behaviors including:

- Searching contiguous areas defined by war fighters.
- Searching linear networks such as roads.
- Transiting to a waypoint.
- Blue-force over-watch.
- Target tracking.
- Perimeter patrol.
- Information exchange infrastructure, in which unmanned vehicles maneuver to form a network connection between an information source, such as an unattended sensor, and war fighters that require information on the source. Note that the war fighter is not required to specify this behavior; the war fighter need only specify the information need, and the vehicle(s) utilizes this behavior as a means to satisfy the need.
- Active diagnosis, in which vehicles reduce uncertain or incomplete observations through their organic sensing capabilities. For example, a UAV with a sensing capability capable of classifying targets will automatically move to and classify unclassified targets being tracked by a cooperating radar.

In addition to the mission-level behaviors enumerated above, OPISR vehicles exhibit certain attributes within all behaviors. These universal attributes are:

- Avoiding obstacles or user-defined out-of-bounds areas.
- Responding to direct human commands. OPISR unmanned vehicles are designed to function autonomously in response to mission-level objectives; however, when operators provide explicit flight instructions, OPISR vehicles *always* respond to the human commands in preference to the autonomous commands.

51.2.3.5 Experimentation

The current OPISR system is the culmination of a decade-long exploration in autonomous unmanned vehicles. Experimentation with DCF began in 2002 as part of an effort to investigate agent-based control of unmanned vehicles to support the U.S. Army's Future Combat System. These early efforts focused predominantly on cooperative search, the results of which are described by Chalmers (Scheidt et al. 2004). Since 2002 thousands of simulated engagements have been conducted with DCF. These simulations have shown that DCF's computational load is independent of the number of vehicles cooperating to solve a mission. Two-hundred vehicle real-time simulations have been run on a single-Pentium class processor. Simulations from a variety of ISR missions have repeatedly shown that vehicle behavior is robust to perturbations in the number of vehicles or the lay-down of those vehicles.

Hardware in-the-loop experimentation with DCF began in 2003 under a joint effort between the Johns Hopkins University, the Army Research Laboratory and Altarum, Inc. In the summer of 2004, this effort conducted a series unmanned air and unmanned ground vehicles experiments at the Aberdeen Proving Grounds (APG). Low-level control was provided by MicroPilot autopilots (air vehicles) and iRobot Mobility (ground vehicles). High-level control of these vehicles was provided by DCF and Altarum's pheromone-based swarming algorithm (Parunak 1997). The distributed blackboard system was used to facilitate sharing between the ground vehicles, although centralized data sharing was used to support UAV control. High-level ground vehicle control software was located onboard the vehicles while high-level air vehicle control software was located onboard a ground station that communicated with the onboard autopilot over a 900 MHz communications link. The air vehicles used in these experiments were Army Mig-27 target drones (Fig. 51.7). These drones have a 6-foot wing span and are capable of air speeds of 60 knots. The ground vehicles used were iRobot ATRV, ATRV-JR, and mini robots. This effort concluded in an October 2004 demonstration in which two air vehicles and four ground vehicles conducted a multi-objective mission at APG. The air vehicles were equipped with GPS for localization and notional EO sensors for target detection and tracking. Mission objectives were provided by three independent human users. Objectives included are as follows: (1) patrol the base, (2) protect the moving convoy, (3) search operator-defined areas, (4) track unclassified targets, (5) classify unclassified targets, and (6) interdict targets classified as threats. AUVs supported objectives 2, 3, and 4. The demonstration started with UGVs patrolling the base and UAVs searching the engagement area. A convoy entered the engagement area with the intention of transiting to the base. The officer leading the convoy requested protection, causing the UAVs to change mode to cover the convoy en route. While patrolling the area in front of the convoy, the UAVs detected a number of dismounts loitering at a road intersection in the convoy's path. This information was relayed to the convoy, causing the human driver to stop prior to the intersection. This same information was relayed to the UGVs, causing the UGV with the acoustic sensor payload to approach the intersection. Once in range of the intersection, the acoustic UGV classified a subset of the dismounts as hostile targets. This information caused the UGVs to pursue the hostile targets, which then fled the area. Once the intersection was cleared, the convoy completed its transit to the base. This demonstration, and variants of it, was successfully performed a number of times. During one demonstration a UGV suffered a hardware fault and went off-line. The other vehicles recognized the sudden absence of their peer, adapting their actions to maintain overall operational effectiveness.

A similar set of experiments were conducted at the Department of Energy's Nevada Test Site in the summer of 2005. These experiments used three Procerus Unicorn UAVs and infrared unmanned ground sensors (UGS). In these experiments all of the vehicles used DCF as their high-level control policy. The UAVs performed area search, road search, target tracking, and, for the first time, airborne information exchange. The engagement consisted of two mounted blue-force patrols,

Fig. 51.7 The first UAVs to fly using OPISR's DCF were modified U.S. Army Mig-27 drones



a single-dismounted aggressor, and two-mounted aggressors. The UGS, using onboard automated target recognition algorithms, detected and identified the dismounted and mounted adversaries. These detections caused the UAVs to track the adversaries and relay the contact information to the blue-force patrols. These experiments were the first in-flight uses of the previously described active-metadata framework.

Also in 2005, DCF was successfully used in experiments to control UAVs detecting, taking samples from and tracking atmospheric plumes. These experiments used three AeroVironment Dragon-eye UAVs. The plume detection experiments were motivated by a desire to provide first-responders with an ability to rapidly identify aerosol contaminants emanating from an industrial accident and to understand and predict the location of those contaminants. These experiments were conducted at a Department of Homeland Security facility in Michigan.

From 2006 to February 2008, DCF has been used regularly in the Naval Postgraduate School's TNT experiments held at Camp Roberts, CA. These experiments have deployed as many as six Procerus Unicorns a fully mission-based swarm. The UAVs used for TNT are fully independent, as the high-level control, a variety of sensors, and the automated target recognition algorithms have been moved onboard the UAVs. Additional behaviors have also been demonstrated at TNT including automated obstacle avoidance (fixed and airborne) and communications. In a 2007 experiment six vehicles cooperated to provide streaming video of arbitrarily defined objects from over the horizon to human users.

DCF has been used as a control metaphor in sea-based experiments conducted predominantly in littoral environments. Starting in 2005 DCF has been used to control several types of unmanned sea surface vehicles conducting search and track n'trail missions at speeds of up to 40 knots with no man in the loop. In 2007 and 2008 OPISR software was used to control unmanned undersea vehicles on search and track missions.

Between 2002 and 2010 twenty hardware experiments were conducted using elements of the OPISR system, including DCF (Scheidt et al. 2005), the distributed blackboard (Nii 1986; Hawthorne et al. 2004), delay-tolerant communications



Fig. 51.8 OPISR vehicles from the 2011 Webster field demonstration including one (of four) ScanEagles, two custom surface vehicles, one (of six) Procerus Unicorns, and an OceanServer Iver2 undersea vehicle

(Bamberger et al. 2004), and simultaneous support for multiple end users (Stipes et al. 2007). As successful as these experiments have been prior to 2011, the full suite of OPISR capabilities described in this chapter had not been demonstrated on a large disparate set of vehicles. In September 2011 a multi-vehicle system consisting of 16 OPISR-enabled nodes, including 9 UAVs in support of 3 users, was conducted at Webster Air Field in St. Inigoes, MD, and the surrounding Chesapeake Bay. The 2011 demonstration mixed air, ground, and sea ISR needs with surveillance being conducted under the water, on the water, and on and over land. The autonomous unmanned vehicles included four Boeing ScanEagles, six Procerus Unicorns, a Segway RMP ground vehicles, custom surface vehicles, and an OceanServer Iver2 undersea vehicle. The air, surface, and undersea vehicles are shown in Fig. 51.8. These vehicles used a wide range of payload sensors to detect, classify, and track waterborne vehicles, land vehicles, dismounts, and mine-like objects, including EO, IR, radar, AIS, passive acoustic, side-scan sonar, and LIDAR. ISR tasking was generated by three proxy operators, two of which were on land (one mounted and one dismounted) and one of which was on the water. ISR tasks requested required the use of all of the vehicle behaviors previously described.

The OPISR capabilities demonstrated by OPISR UAVs at St. Inigoes included a set of six enabling autonomous submission capabilities, which are:

1. Area search – When user(s) requests that one or more contiguous areas should be searched, the autonomous vehicles respond by searching those areas.
2. Road network search – When user(s) request that one or more roads should be searched, the autonomous vehicles respond by searching those roads. This is functionally identical to performing an area search over an area that is confined to (or focused on) roads of interest.
3. Overwatch – A convoy protection mode that when a user requests “overwatch” protection, one or more vehicles circle the protected user. This is functionally identical to track a moving target, except that the $U \times V$ sensors will be directed at the region surrounding the convoy, rather than directly at a target.
4. Communication relay – When an area to be searched (see behavior #1) is farther from the user interested in the area than the vehicle-user communications range, one or more vehicles autonomously form a communications chain to relay data from the searched area to the user.
5. Obstacle avoidance – Obstacles that are known by a vehicle are avoided. Note that obstacles may include physical obstacles detected by vehicle sensors (e.g., trees) and/or abstract obstacles provided by users such as no-fly zones.
6. Behavior switching – Vehicles are capable of exhibiting multiple behaviors depending upon current user goals and circumstances.

One highlight of the experiment was the indirect access to time-sensitive data from a remote camera sensor – located well outside the direct communication ranges of the C2 ground stations and their radios – to OPISR users by autonomous UAV communications chains. This communications link was formed fully autonomously, even to the extent that there was no specific command given to the ScanEagle UAV that it forms a communications chain. The ScanEagle was tasked to patrol the region, and as it became aware of sensor data, it relayed that data to the user, who immediately had the sensor data available on his display.

Another key OPISR feature that was demonstrated in St. Inigoes was the OPISR UAV ability to coordinate on tasks that it (the UAV) cannot satisfy without recruiting other vehicle types. Very importantly in this experiment, command was shown not only from one ground station to multiple heterogeneous vehicle platforms but also commands from multiple users at multiple ground stations could set the goals of any and all OPISR components.

During experimentation sensor and mission information was delivered to both C2 user stations both directly and via UAV communications chaining. This delivery occurred both automatically and in response to specific user requests. All relevant mission data was made available and reported to the C2 user, including the status of assigned search missions, represented as “fog of war” over the map geo display; detections from the various UGS, the two USVs, and the UUV; and camera imagery from the various platforms. Within the flight time windows available, over 65,000 images were collected by the various camera sensors, including onboard UAV payload sensors and relayed to the C2 stations, available to any of the notional war fighter users as their ground node was connected to cognizant portions of the mesh network.

51.3 Conclusion

The OPISR system is a framework that provides a capability through which numerous unmanned platforms simultaneously provide real-time actionable intelligence to tactical units, provide abstract manageable situation awareness to theater commanders, and provide high-quality forensic data to analysts. OPISR is a demonstrated system that includes a distributed self-localizing camera payload that provides imagery and positional metadata necessary to stitch information from multiple sources, a distributed collaboration system that is based upon robust ad hoc wireless communications and agent-based data management, and a user interface that allows users to receive real-time stitched imagery from unmanned vehicles that does not require users to directly control (or even expressly be aware of) the unmanned vehicles producing the imagery. OPISR is a bold vision that presents an innovative approach to ISR, an important enabler emphasized in the Quadrennial Defense Review and other key policy documents, and gives the laboratory an enhanced ability to help sponsors address future capability gaps in this critical area.

References

- R.C. Arkin, *Governing Lethal Behavior in Autonomous Robots* (CRC, Boca Raton, 2009)
- R. Bamberger, R.C. Hawthorne, O. Farrag, A communications architecture for a swarm of small unmanned, autonomous air vehicles, in *AUVSI's Unmanned Systems North America Symposium*, Anaheim, 3 Aug 2004
- B. Bethke, M. Valenti, J. How, Experimental demonstration of UAV task assignment with integrated health monitoring. *IEEE Robot. Autom. Mag.*, Mar 2008
- Defense Systems Staff (2012) Army readies on-demand imagery tool for battlefield use. *Defense Systems*, 1 June
- R.C. Hawthorne, T. Neighoff, D. Patrone, D. Scheidt, Dynamic world modeling in a swarm of heterogeneous autonomous vehicle, in *AUVSI Unmanned System North America*, Aug 2004
- A.N. Kolmogorov, On tables of random numbers. *Theor. Comput. Sci.* **207**(2), 387–395 (1998)
- V. Kumar, N. Michael, Opportunities and challenges with autonomous micro aerial vehicles. *Int. J. Robot. Res.* **31**(11), 1279–1291 (2012)
- H. Kwon, D. Pack, Cooperative target localization by multiple unmanned aircraft systems using sensor fusion quality. *Optim. Lett. Spl. Issue Dyn. Inf. Syst.* (Springer-Verlag) (2011)
- M. Mamei, F. Zambonelli, L. Leonardi, Co-fields: a unifying approach to swarm intelligence, in *3rd International Workshop on Engineering Societies in the Agents' World*, Madrid (E), LNAI, Sept 2002
- H. Nii, Blackboard systems. *AI Mag* **7**(2), 38–53 (1986); **3**, 82–106
- V.D. Parunak, 'Go to the Ant': engineering principles from natural multi-agent systems. *Ann Oper Res* **76**, 69–101 (1997)
- D. Scheidt, M. Pekala, The impact of entropic drag on command and control, in *Proceedings of 12th International Command and Control Research and Technology Symposium (ICCRTS)*, Newport, 19–21 June 2007
- D. Scheidt, K. Schultz, On optimizing command and control, in *International Command and Control Research Technology Symposium*, Quebec City, June 2011
- D. Scheidt, T. Neighoff, R. Bamberger, R. Chalmers, Cooperating unmanned vehicles, in *AIAA 3rd "Unmanned Unlimited" Technical Conference*, Chicago, 20 Sept 2004

- D. Scheidt, T. Neighoff, J. Stipes, Cooperating unmanned vehicles, in *IEEE International Conference on Networking, Sensing and Control*, Tuscon, 19–22 Mar 2005
- C.E. Shannon, A mathematical theory of communications. *Bell Syst. Tech. J.* **27**, 379–423, 623–656 (1948)
- J. Stipes, D. Scheidt, R.C. Hawthorne, Cooperating unmanned vehicles, in *International Conference on Robotics and Automation*, Rome, 10 Apr 2007
- J. Tisdale, Z. Kim, K. Hedrick, An autonomous system for cooperative search and localization using unmanned vehicles, in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Honolulu, Aug 2008
- J.A. Wheeler, Information, physics, quantum: the search for links, complexity, entropy and the physics of information, in *A Proceedings Volume in the Sante Fe Institute Studies in the Sciences of Complexity*, ed. by W.H. Zurek (Westview Press, 1990)

Section XI

MAVs and Bio-Inspired UAVs

Robert C. Michelson

Kimon P. Valavanis and George J. Vachtsevanos

MAVs and Bio-inspired UAVs addresses the emerging UAV area of Micro Aerial Vehicles (MAVs) and bio-inspired MAVs and UAVs. Research and development activities in MAVs have accelerated significantly over the past years, driven by a need for small autonomous vehicles that can execute a variety of tasks, such as Intelligence, Surveillance, and Reconnaissance (ISR) in complex urban environments, search and rescue operations, and security and border patrol, among other applications. A substantial component of the MAV research is inspired by the exceptional flying behaviors of biological species, i.e., birds and insects. MAVs can be operated by a single person offering low weight and cost, extreme maneuvering capabilities, and rapid response times to requests for visual observations. Innovative MAV concepts are motivating advanced research and development sponsored by the government and industry.

► **Micro Air Vehicles** by R.C. Michelson addresses challenges related to the design of MAVs. Such challenges span across aerospace, electrical, mechanical, and computer engineering because of flight regime in which these tiny aircrafts operate. Aerospace designers must contend with issues surrounding low Reynolds number flight, while electrical and mechanical designers are concerned with issues of energy storage, behavior of materials at small scales, and non-scaling items. The missions at which MAVs excel demand increased levels of autonomy, forcing computer engineers to create innate onboard intelligence exhibiting high bandwidth and

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

superior abilities to interpret obstacle-rich environments not usually encountered by larger flying machines.

► **Survey of the Human-Centered Approach to Micro Air Vehicles** by S. Michelson presents a detailed overview of some of the Human Systems Integration (HSI) and Human Factors Engineering (HFE) issues involved with MAVs. The importance of a total systems engineering approach to MAV design, how MAVs fit into commonly accepted Human Systems Integration domains, and an exposure of some emerging issues with MAVs that require further research are discussed. The unique attributes of MAVs in terms of their size and control methods, combined with the challenges of the dynamic operational environments where they are deployed, represent HFE issues exclusive to the MAV platform that require special consideration. The importance of designing for the human operator is paramount for successful outcomes with MAV platforms. Specifically highlighted are some areas where currently researched HFE issues are particularly applicable to MAVs as opposed to large-scale systems.

► **Development of Insect-Sized MAVs** by Sunada, Liu, Tokutake, and Kubo describes a prototype bio-inspired flapping MAV with flexible wings, with a specific focus on the flexible-wing aerodynamics. The flapping-wing MAV has a weight of 2.4–3.0 g and a wingspan of 10–12 cm, which is comparable to hawk moths and hummingbirds. The MAV's flexible-wing aerodynamics is analyzed by combining an *in-house* computational fluid dynamics (CFD) method and wind tunnel experiments (EXP). In addition, fixed-wing and rotary-wing MAVs with elements that enable the miniaturization of an aerial vehicle are introduced.

► **Flapping-Wing Propelled Micro Air Vehicles** by Jones and Platzer presents a brief history of the major discoveries in the scientific exploration of flapping-wing flight. This is followed by a short review of the basic concepts of lift generation on wings in low-speed, steady flight, which leads into a discussion of the generation of thrust due to the flapping of wings. The aerodynamics of single flapping wings in forward and hovering flight, of flapping tandem and biplane wings, and of dual wings using the clap-and-fling effects are discussed. The chapter concludes with an overview of the major characteristics of five representative flapping-wing propelled MAVs developed to date, including models developed at the AeroVironment Company, Naval Postgraduate School, Wright State University, and Delft University.

► **Inventing a Biologically Inspired, Energy Efficient Micro Aerial Vehicle** by Ratti and Vachtsevanos introduces a novel framework for the design and control of a MAV, where the conceptual design is based on biologically inspired principles and emulates a dragonfly. The chapter addresses the design and control features of the proposed design and gives an overview on the developmental efforts towards the prototyping of the flyer. The potential applications for such a high endurance vehicle are numerous, including air deployable mass surveillance in cluster and swarm formations. The disposability of the vehicle helps in battlefield deployment as well, where such an MAV is made available to soldiers for proximity sensing and threat-level assessment. Other applications include search and rescue operations and civilian law enforcement.

► **Issues Surrounding Communications with Micro Aerial Vehicles** by C. Michelson seeks to answer many of the communications-linked questions that MAV designers have and gives a high-level overview of the factors that affect MAV data and control links. Challenges related to communications with MAVs are because of their size and very limited payload capabilities. Limited payload capacity leads to considerable constraints on power sources, sensors, and communication systems. Power sources are by far the most weight-inefficient components on an MAV. MAV designers are forced to look elsewhere to optimize their designs. The best way to do so in lieu of focusing on improving battery technology is to optimize the systems that draw power, thereby increasing endurance. Motors, onboard processing, and communications transceivers are the largest three power consumers on MAVs today. While motors and embedded processing are important to optimize, the sheer number of available communications options may leave MAV designers unsure how to proceed. By building an MAV around its onboard communications system, designers increase reliability, endurance, and capability with little or no added cost. Care must be taken to ensure that the end result meets the power, aerodynamic, and electromagnetic requirements for the particular MAV and its particular mission.

Collectively, the chapters in this section refer to all aspects of MAV design, control, communications, operation, and applications.

Robert C. Michelson

Contents

53.1 Introduction 1306

53.2 Historical Perspective 1306

53.3 The Energy Barrier 1307

53.4 Biological Inspiration 1308

53.5 Mission Effectiveness and Payload 1309

53.6 The Man and the Machine 1309

53.7 Conclusion 1309

References 1310

Abstract

The design of micro air vehicles (MAV) presents one of the most formidable engineering challenges, not only to aerospace, but electrical, mechanical, and computer engineers because of flight regime in which these tiny aircraft operate. Aerospace designers must contend with issues surrounding low Reynolds number flight, while electrical and mechanical designers will be concerned with issues of energy storage, behavior of materials at small scales, and non-scaling items. The missions at which MAVs will excel demand increased levels of autonomy, forcing computer engineers to create innate onboard intelligence exhibiting high bandwidth and superior abilities to interpret obstacle-rich environments not usually encountered by larger flying machines. This section deals with MAVs that conform to the original Defense Advanced Research Projects Agency (DARPA) definition (15 cm and smaller), rather than the larger UAVs which many are prone to label as “micro air vehicles,” but in reality are just small-scale UAVs. Topics covered in this section are somewhat unique relative to typical MAV discussions

R.C. Michelson
Aerospace, Transportation, and Advanced Systems Laboratory, Georgia Tech Research Institute,
Smyrna, GA, USA
e-mail: robert.michelson@gtri.gatech.edu

in other texts dealing with MAVs that focus primarily on the air vehicle rather than the system. Not only is the air vehicle covered here but also issues of MAV deployment which are all too often neglected in discussions of MAV operations (e.g., communications and operational issues).

53.1 Introduction

“No other air vehicle design space has presented the mix of challenges as that of miniature flight platforms. By definition these tiny platforms are unmanned and endeavor to invade the flight regime of birds and insects. In order to do so, the creators of these aerial robots must address the same physical design constraints which have already been mastered by the world of airborne biology, including low Reynolds number aerodynamics, high energy density, and extreme miniaturization” (Michelson 2004).

Critical issues arise when working at this scale, energy storage being one of the most significant. Nontraditional aerodynamics in low Reynolds number regimes complicate flight efficiency, driving designs toward the biologically inspired and away from the conventional. Due to the inability to scale all systems as a result of basic physics-based constraints, normally simple telemetry systems are constrained to operate at short ranges and high frequencies due to limitations on antenna structures. Further, teleoperated operation becomes impractical as the tiny MAV fades from view over fairly short distances. This leads to designs requiring greater or full autonomy – but an entirely new set of issues arise with fully autonomous MAVs and their potential for emergent behaviors that are difficult to predict and test.

The following sections of this introductory MAV chapter will put these issues into perspective and show their interrelation.

53.2 Historical Perspective

Interest in tiny flying machines had its origins with the notion that a small insect-like flying platform could be devised for covert operations. The U.S. Central Intelligence Agency (CIA) experimented with the creation of a remotely-controlled pneumatic “dragon fly.” In 1993 a RAND Corporation study discussed the potential use of sensor-carrying insects and the concept of truly “micro” air vehicles (Hundley and Gritton 1994). Japanese researchers attempted to stimulate motor neurons in cockroaches to control the insect trajectory by a radio link (BBCi Sci/Tech 1998). Tiny airplanes powered by biological motors (large flies glued to a fixed-wing fuselage) have been demonstrated recently but was actually a subject of experimentation at the turn of the twentieth century by Nikola Tesla (Cheney 1981).

“Micro Air Vehicle (MAV) is a most unfortunate name given to this class of air vehicles because none are truly ‘micro’ and the original (ca. 1995) official DARPA vehicle definition requiring a maximum 15 cm (6 in.) dimension confirmed the name to be a total misnomer” (Michelson 2006). DARPA defined MAVs as “being less

than 15 cm” because this represented the juncture at which low Reynolds number effects begin to dominate and beyond which, integration of energy, propulsion, aerodynamic structures, and intelligence is a necessity (Michelson 2004).

Initially (ca. 1997), DARPA’s vision for MAVs was that the individual soldiers at the platoon, company, or brigade level would use such vehicles for reconnaissance and surveillance, battle damage assessment, targeting, emplacing sensors, communications relays, or for sensing chemical, nuclear, or biological substances. The 15-cm vehicles would be able to conduct real-time imaging, have ranges of up to 10 km, and speeds of up to 30 miles per hour for missions that are 20 min–2 h long. By 2012, this vision has been realized, but not without many operational impediments.

53.3 The Energy Barrier

In order to realize the endurance envisioned by DARPA, energy sources of significant energy density had to be developed. Various MAV concepts using high energy density battery technology, fuels (both combustion and catalyst-based), as well as energy harvesting techniques have been explored. Battery technologies exist that exhibit amazing energy densities; however, most of these are primary cells (nonrechargeable) and because of the greater danger of explosion from rapid overheating due to internal or external shorting of the cells, these battery types have not been fielded in MAVs. Presently, the most common high energy density battery chemistry in common use is the lithium polymer (LiPo) battery. LiPo technology allows for the creation of relatively high energy density secondary (rechargeable) cells and is the most widely used energy source for electric flight propulsion at present.

Electric propulsion is almost exclusively by means of electric motors driving propellers/rotors or flapping wing mechanisms. Synthetic muscle technologies such as piezoelectric elements, shape-memory alloys, rheological fluids, and electro-polymers offer an array of electrically controlled actuators which have been touted as a potential source of power for biologically inspired aerial robots, but all have yet to demonstrate sufficient efficiency to lift even the best high energy density battery (Michelson 2004).

Researchers have also considered the use of chemical power sources due to the higher energy density contained in fuels of various types. Stored energy becomes a significant impediment as MAV mission duration increases. The present state of the art in battery technology does not allow for long endurance MAV missions, though it is hoped that someday improved electrical storage media (carbon-air, fuel cells, etc.) will result in the energy densities required for useful long endurance missions in MAV-sized vehicles. Near-term solutions to onboard energy storage will come from chemical or fossil fuels because of their superior energy density. As a point of comparison, consider the amount of releasable energy stored in a drop of gasoline compared to that which can be stored in a battery the size of a drop of gasoline (Michelson 1998). Further, as the mission progresses, a fueled propulsion system

gets lighter, whereas a battery-operated propulsion system remains just as heavy at the end of the mission as the beginning. One example of a chemically fueled MAV concept is the Entomopter, a flapping wing MAV that was under development by DARPA and NASA. The “Entomopter” (*entomo* as in entomology + *pteron* meaning wing or a “winged insect machine”) uses a liquid monopropellant such as high-concentration hydrogen peroxide or hydrazine in the presence of a catalyst to create gas for reciprocating actuation of flapping wings.

A third approach to powering MAVs for longer endurance missions is energy harvesting. Small birds and insects are consumed with the task of energy harvesting: the search for food. Hummingbirds, the smallest of all avians, feeding on dilute nectar can ingest nearly three times their body mass in nectar per day to sustain life and mobility (Powers and Nagy 1988; McWhorther and Martinez del Rio 1999). Their small bodies cannot carry large amounts of food, so to improve efficiency they choose high energy foods that provide immediate energy access (sugars) as do many insects. Tiny aerial robots suffer from the same need for readily available energy. The energy density of the best battery technologies currently available still cannot match that which is locked chemically in various compounds such as sugars. Some have advocated energy harvesting through the use of solar panels on MAVs. Unfortunately, the efficiency of current solar cells (roughly 5 % for common cells, ranging up to 28 % for some of the best triple-junction gallium arsenide space-qualified cells) in sizes that could be carried by a MAV is insufficient for sustained flight (Dornheim 2003). The extra weight of such cells negates their use as an endurance extender, and their low voltage output is incompatible with many of the electronic actuators proposed (e.g., piezoelectric, electro polymers). Finally, night operation or flight through shadows is precluded.

Research has been conducted in leaching power from power lines by UAVs, and a ground vehicle known as “slugbot” was developed to consume organic matter and digest it to produce energy for locomotion. In both cases, the amount of energy harvested and the time taken to process the energy made the application impractical.

53.4 Biological Inspiration

MAVs have been demonstrated as fixed, rotary, and flapping wing vehicles. As the size of the MAV decreases and the vehicle operates in lower and lower Reynolds number regimes, the efficiency for flight moves away from classical fixed-wing aerodynamics toward flapping wing solutions. It is no accident that all insects use flapping wings to fly. A number of MAVs such as the Entomopter mentioned above and various other unconventional flappers have been demonstrated. Some of these designs are “biomimetic” (copying nature), while others are “biologically inspired” (taking principals from nature but implementing them in a way that goes beyond what is found in nature). Examples of both approaches are discussed in detail in the subsequent chapters ► [Development of Insect-sized MAVs](#) and ► [Flapping-Wing Propelled Micro Air Vehicles](#).

53.5 Mission Effectiveness and Payload

One area that is usually glossed over in most texts dealing with MAVs involves issues surrounding communications with micro air vehicles. Because of their small size, MAVs are restricted not only in the power that they can carry to support telemetry, but they are also hampered by the physical constraints placed on radiating elements. Operation at greater ranges is enhanced by the use of lower frequencies. The same is true when attempting to transmit through obstacles such as buildings or foliage. Unfortunately, lower frequency operation becomes inefficient as the antenna apertures shrink in size. Increasing transmitted power is not an effective solution due to the limited energy carried onboard the MAV as well as the mismatch between a low-frequency transmitter and a miniature antenna. As a result, higher-frequency telemetry is desirable from an emissions standpoint, but not from an attenuation standpoint. Specific consideration must be given to transmitter power, data and command link design, amplifier types, modulation techniques, simultaneous communications, and antenna aerodynamics. Various techniques to deal with these conflicting MAV telemetry design parameters are discussed in chapter [►Issues Surrounding Communications with Micro Aerial Vehicles](#).

53.6 The Man and the Machine

Certainly one of the least discussed topics relating to MAVs is the human interaction with the MAV system. A prime example of an MAV-specific human interaction problem is the fact that as the MAV flies away from the operator, even at only short ranges, useful visual feedback is lost due to its tiny size. Further, if the MAV mission involves the penetration of a building, visual feedback is lost. As briefly discussed in the prior section, teleoperation via a data link is problematic for any number of reasons, none the least of which is the cost in onboard stored energy.

Another significant issue arises when autonomy is used to relieve the operator from having to control the MAV remotely. A fully autonomous MAV, while very desirable for many reasons, is highly difficult to test because of the unpredictable potential for emergent behaviors (Michelson 2008).

Chapter [►Survey of the Human-Centered Approach to Micro Air Vehicles](#) discusses these issues as well as how to effectively incorporate a human-systems integration approach to the design of MAVs in light of a valid MAV concept of operation. Automation, situation awareness, and workload issues are discussed, as are the test, evaluation, and training considerations particular to MAV operators.

53.7 Conclusion

The genesis of the micro air vehicle concept and the initial funding by DARPA to push the technology forward have resulted in many MAV configurations using

fixed, rotary, and flapping wing propulsion. The critical element preventing greater performance over the past 20 years has been high energy density propulsive power, and this remains true to this day. Battery and chemical fuel power sources, with few exceptions, have not progressed at a rate which allows MAVs to have the endurance of birds of similar size. Biomimetic control systems have been developed in laboratory settings, but the ability to control a MAV with the agility of a small bird, much less than that of a dragonfly, still eludes designers.

When considering the best communications architecture for MAVs, designers must remember that everything is a trade-off when balancing miniaturization with high bandwidth remote non-line-of-sight telemetry. RF characteristics are largely bounded by the universal constraints of physics and are therefore not controllable by the designer. Remaining mindful of the distinction between human systems integration and human factors engineering with regard to MAV development is pivotal to correctly accounting for the human in the design process. The uniqueness of the MAV and its deployment is creating the need for a new understanding of man-machine systems in the unique operational environment of the MAV.

MAVs are still in their infancy 20 years after the DARPA vision for such a vehicle, largely due to technical challenges. It is not that one does not understand the design issues; rather, one lacks the component efficiency and level of mechanical miniaturization necessary to compete with biological systems. Since fully 50 % of a MAV's gross takeoff weight is consumed in stored energy, one of the areas in which future MAV designers will reap the greatest gains will come from advances in propulsive energy density.

References

- BBCi Sci/Tech, Cyberbugs are go. Wednesday, March 4, 1998 – Published at 00:21 GMT, at, <http://news.bbc.co.uk/2/hi/science/nature/61822.stm>. Accessed 18 Aug 2012
- M. Cheney, *Tesla: Man Out of Time* (Dorset, New York, 1981), p. 7. ISBN 0-88029-419-1
- M. Dornheim, Get me through the night, in *Aviation Week & Space Technology* (McGraw-Hill, New York, 2003), pp. 66–68
- R. Hundley, E. Gritton, *Future Technology-Driven Revolutions in Military Operations*. Document No. DB-110-ARPA (RAND Corporation, Santa Monica, 1994)
- T. McWhorther, C. Martinez del Rio, Food ingestion and water turnover in hummingbirds: how much dietary water is absorbed? *J. Exp. Biol.* **202**, 2851–2858 (1999)
- R. Michelson, Update on flapping wing micro air vehicle research- ongoing work to develop a flapping wing, crawling entomopter, in *13th Bristol International RPV/UAV Systems Conference Proceedings*, Bristol, 1998, pp. 30.1–30.12
- R. Michelson, Novel approaches to miniature flight platforms. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **218**(Special Issue Paper), 363–373 (2004)
- R. Michelson, Very small flying machines, in *2006 Yearbook of Science & Technology* (McGraw-Hill, New York, 2006), pp. 341–344. ISBN 0-07-146205-8
- R. Michelson, Test and evaluation of fully autonomous micro air vehicles. *ITEA J.* **29**(4), 367–374 (2008)
- D. Powers, K. Nagy, Field metabolic rate and food consumption by free-living Anna's hummingbirds (*Calypte anna*). *Physiol. Zool.* **61**, 500–506 (1988)

Stuart Michelson

Contents

54.1 Introduction 1312

54.2 Terms 1313

54.3 Incorporating an HSI-Centric Design Approach 1314

54.4 MAV Concept of Operations 1315

54.5 MAVs and the Man-Machine System 1315

54.6 Automation, Situation Awareness, and Workload 1318

54.7 Test, Evaluation, and Training 1321

54.8 MAV Ground Control Station Considerations 1323

54.9 Ruggedization 1325

54.10 Conclusion 1326

References 1327

Abstract

A detailed overview of some of the Human Systems Integration (HSI) and Human Factors Engineering (HFE) issues involved with the newest and perhaps fastest growing research area in unmanned systems, micro air vehicles (MAVs), will be presented. This work will be useful to those studying MAV system concepts and designs, managers of HSI programs, users of MAV systems, and those who design MAVs and the resources to support them. The importance of a total systems engineering approach to MAV design, how MAVs fit into commonly accepted Human Systems Integration domains, and an exposure of some emerging issues with MAVs that require further research are discussed.

S. Michelson
Human Systems Integration Division, Georgia Tech Research Institute, Atlanta, GA, USA
e-mail: stuart.michelson@gtri.gatech.edu

The unique attributes of MAVs in terms of their size and control methods, combined with the challenges of the dynamic operational environments where they are deployed (such as the battlefield), represent HFE issues exclusive to the MAV platform that require special consideration. The importance of designing for the human operator is paramount for successful outcomes with MAV platforms.

Literature currently addressing HFE issues with unmanned platforms generally lump all flying systems together, making no distinction between the large high-altitude platforms and smaller ones, despite there being a unique set of challenges that are specific to smaller platforms. Specifically highlighted are some areas where currently researched HFE issues are particularly applicable to MAVs as opposed to large-scale systems.

54.1 Introduction

Successfully implementing micro air vehicle (MAV) programs requires one to maintain a total systems approach, and Human Systems Integration (HSI) must be an integral part of that approach. These are presuppositions to the remainder of the content in this work. Acquiring a human-centered approach to analyzing the observable problems with unmanned systems is essential, but one might feel inclined to question what the human has to do with such systems at all, especially when the hallmark of the domain is the absence of humans. While it is important to comprehend that the human is an integral part of the design of any unmanned system, humans are particularly important to the MAV platform where operators are thought to have more active roles in commanding the craft, and systems have a wider spread across the levels of automation. This reality can complicate system design and function allocation. A failed systems approach that deemphasizes the human in the loop will result in human-related challenges such as unmanageable operator workload or poor situational awareness (SA) culminating in compromised safety for warfighters.

The Defense Acquisition Guidebook indicates that a total system approach “includes not only the prime mission equipment, but also the people who operate, maintain, and support the system; the training devices; and the operational and support infrastructure” (United States Department of Defense 2012). It would be a grave error for designers to assume that just because a human is not physically present onboard the vehicle that they do not need to be knowledgeable about human attributes and limitations. Testing and evaluating these attributes and limitations is a major aspect of Human Factors Engineering (HFE), and given the uniqueness of the manner in which MAVs are deployed, special design considerations must be given to elements such as human anthropometry and cognitive, physical, and sensory abilities. While not every engineer needs to be an expert in these areas, they should at least be able to discuss them intelligently with HFE/HSI practitioners and understand their extreme importance to successful system design, as it is their job to guarantee that the human is considered throughout every portion of the design process.

54.2 Terms

Introducing system designers and operators to HSI and HFE considerations associated with MAVs requires one to recognize the finer distinctions between commonly used terms within the systems engineering domain.

A brief discussion of terms used in this work is necessary for clarity since many use some of the common terms interchangeably while the author does not. Note that while there are many definitions of Human Systems Integration, the author ascribes to the United States Air Force's definition of Human Systems Integration which is "the integrated and comprehensive analysis, design, and assessment of requirements, concepts and resources for system manpower, personnel, training, environment, safety, occupational health, habitability, survivability, and human factors engineering, with the aim to reduce total ownership cost, while optimizing total mission performance" (United States Department of Defense 2009). While different branches of the armed forces emphasize varying arrangements of domains (see MANPRINT, for example), the underlying theme is that HSI is a vital element that optimizes system design to enhance system performance while maximizing the abilities and mitigating the limitations of humans. In so doing, the abilities of the warfighter are enhanced and total ownership costs of a system are reduced (Air Force Human Systems Integration Office 2009). It is important to emphasize that HSI from an engineering standpoint not only deals with requirements and concepts but people and equipment. In this way, HSI should be understood to be a broader concept than HFE although many incorrectly use the two terms interchangeably. The major motivation for incorporating HSI into MAV programs is to control total ownership costs while maximizing system effectiveness.

Note that Human Factors Engineering is included as one of the domains of HSI. HFE deals more exclusively with the realization of design criteria, psychological elements of human behavior, as well as physical and mental limitations of humans. By understanding human attributes (anthropometry, for example), HFE analysis seeks to enhance performance (to required levels), increase safety, and increase user satisfaction (Wickens et al. 2003). The science of human factors can be said to rest upon generalization and prediction. With respect to unmanned systems, researchers should be interested in generalizing common problems and predicting solutions – all while accounting for the humans in the loop and reducing design-induced failures.

The importance of incorporating HSI and HFE analyses into the design process early cannot be stressed enough. The Department of Defense specifies that design changes can cost 1,000 and 10,000 times more in initial production phases than the same change would cost during a product's earlier design phases (United States Department of Defense 2005). While the human element is a vital consideration throughout the entirety of the design process, it has been shown time and time again that design changes made early are less costly because they do not involve the modification of existing hardware and software. A human-centered approach to MAV development should be early and iterative but also should span throughout the entirety of the design process.

54.3 Incorporating an HSI-Centric Design Approach

A proper approach to system design must include HSI as a consideration of prime importance. So much so that the Department of Defense's acquisition policy indicates that HSI should be central to the formation of integrated product teams (IPTs), which are a vital part of the integrated product and process development method. The Defense Acquisition Guidebook states that the product and process development method is a "management technique that integrates all acquisition activities starting with capabilities definition through systems engineering, production, fielding/deployment and operational support in order to optimize the design, manufacturing, business, and supportability processes. At the core of the IPDD technique are IPTs. Human Systems Integration should be a key consideration during the formation of IPTs" (United States Department of Defense 2012). It is important for HSI experts who are considering the human element for each HSI domain to be included as members of the various IPTs so that the human element is considered throughout the entire design process. This design concept places those charged with including the human element in positions to comprehensively impact the system's design (United States Department of Defense 2012). HSI professionals should be included in the development of a MAV platform's concept of operations rather than being invited into the design later to comment. In reality this rarely occurs, but doing so allows HSI considerations to impact the concept of operations and in turn create systems where total ownership costs are reduced and system effectiveness is maximized.

It is not to be unexpected for engineers to resist the involvement of HSI professionals. Those that do are likely victims of a bad definition of HSI and do not understand how it will positively impact their work, or they think it is an unnecessary cost not realizing that without it their program will be more expensive. Unfortunately, many engineers treat HSI and HFE as checkboxes for after their work is considered complete. A tendency may exist among some engineers to place less emphasis on designing for the human when managing unmanned systems programs such as MAVs. The development of an HSI plan is one way to make sure that human considerations are not deemphasized. Managers must develop an HSI strategy for their program. This takes the form of an HSI plan that should be established early in the design process before any manufacturing takes place (United States Department of Defense 2012). Each phase of the program should include how the HSI plan will address issues associated with each HSI domain. In systems where new technologies are expected to replace or supplement human activities, HSI plans should anticipate the possibility that delays in development might still prevent such measures from being implemented, and therefore, program managers should have alternatives that still account for appropriate levels of human operator workload.

While the finer points of the accepted defense acquisition strategy are beyond the scope of this work, a brief understanding of the context in which MAVs are being developed is valuable. Within the unmanned systems category, incorporating good HSI plans is particularly important for MAVs where typically more involvement is expected of the human operator in terms of direct control of the craft.

This distinction of the MAV platform requires a close look at all of the HSI domains but specifically those that deal with the cognitive, physical, and sensory abilities of humans and their knowledge, skills, abilities, and experience levels as system operators (the personnel, training, and human factors engineering domains).

54.4 MAV Concept of Operations

In contrast to larger, more traditional unmanned systems being widely deployed around the world, MAVs are deployed in a distinctive manner. DARPA's definition of a MAV specifies that the vehicle be smaller than 15 cm (about 6 in.) in any dimension (McMichael and Francis 1997). The MAV was originally thought to be a military asset that could be included in every warfighter's loadout, enabling him/her to engage in intelligence, surveillance, and reconnaissance (ISR) missions while in the field (Michelson 2008). This methodology stands in contrast to larger-scale systems with expansive ground control stations used with popular unmanned aerial vehicles (UAVs) such as Northrop Grumman's Global Hawk or General Atomics' Predator. In such systems, multiple human operators are involved in the command and control of the system from either homeland-based command centers or mobile trailer style units.

From a technical standpoint, MAVs, being craft measuring in mere centimeters, represent a number of issues that have been written about in detail elsewhere. Some of these issues include robustness in inclement weather, endurance, antenna aperture, and strains on command and control due to limited storable energy on board the craft. While these are very real challenges deserving consideration, one cannot look to solve these problems while ignoring or merely paying lip service to the importance of a human-centered approach to design of MAV platforms. The aforementioned issues should be handled considering their effects on humans, but there are other specific issues that are perhaps not often thought of regarding the human's interaction with MAV systems. The remainder of this chapter is dedicated to addressing some common considerations to MAV design and implementation from a human-centered perspective while comparing and contrasting their uniqueness against larger high-altitude UAV systems. HSI professionals involved in the development of a MAV's concept of operations should emphasize that a system is driven by humans. Unmanned systems should be just as concerned with designing for the human in the loop as manned systems. The main difference between the two is that a human is not physically onboard the craft, but UAVs such as Predator have hundreds of humans involved in their operation, maintenance, and support.

54.5 MAVs and the Man-Machine System

As the popularity of UAVs has increased, so has the interest in shrinking their size. This reduction in size has expanded the range of applications for which humans can utilize them. No longer are UAVs limited to outdoor flight at altitude.

Recent developments in research and technology have afforded MAVs the prospect of flying into buildings, down hallways, through air ducts, sewer systems, or even acting as micro spies no bigger than a housefly on the wall. All of these abilities present new challenges not only to the MAV platform's hardware and software but also to the human system designers and operators. The challenges associated with indoor flight such as obstacle avoidance and navigating environments without the assistance of GPS are difficult to overcome and are only enhanced by difficulties such as transmitting valuable location information back to human operators from inside of a structure.

Engineers may feel inclined to meet these challenges with a reductionist approach to systems design. This traditionally popular approach is characterized by a focus on each physical and technical component of a system alone while not emphasizing the behavioral components of systems. This approach has been associated with catastrophic system failures on the order of magnitude of oil spills and commercial aircraft crashes because it does not consider the interaction among all of a system's parts and does not view system components in terms of their relation to accomplishing overall system goals (Czaja and Nair 2012). The best way to avoid this pitfall is to incorporate a good HSI plan which dictates that all the various domains of HSI are considered together.

Since HFE is concerned with enhancing the interactions between humans and all other components of a system, good systems theory sees the understanding of the entirety of a system's components being in concert as implicit to good systems design (Czaja and Nair 2012). As the levels of autonomy continue to increase with MAVs, the complexities of automation and the system components required to support them also increase. As the emphasis continues to be on advancing the state of MAV technology, equal consideration in the realm of system design must be given to the human element or systems will underperform. This is an accepted tenant within the HFE field (Czaja and Nair 2012).

While there are many ways to classify a system, classifying MAV systems in terms of levels of feedback mechanisms is particularly useful to the characterization of the platform. A closed-loop system provides feedback to an operator that is continuous for error correction. This type of system provides human operators with real-time information so that they can determine the difference between actual and desired system states. An open-loop system does not afford an operator the feedback necessary for continuous control (Czaja and Nair 2012). MAVs exhibit levels of autonomy ranging across the spectrum from 100 % remotely piloted robots to entirely autonomous robots making their own decisions in real time based on their environment apart from a human operator. For this reason, MAVs as a category can be closed-loop systems, open-loop systems, or both.

This can complicate the design strategy one should employ since open- and closed-loop systems require varying design strategies. Designers should pay close attention to the consequences of dynamic function allocations to ensure that realistic expectations are being placed on human operators in certain circumstances where the human has more control. Extensive experimentation and testing using realistic scenarios that cover the breadth of possible human and computer workload should be undertaken to mitigate potential workload-related errors. For a more full

discussion of scenario development strategies and workload analysis techniques that extend beyond the scope of this work, see *Mental Workload and Situation Awareness* (Vidulich and Tsang 2012).

When considering the capabilities and limitations of humans for MAV systems, it is important to recognize the way in which advancing vehicle autonomy has changed the nature of man-machine systems. Not only have new technologies enhanced the manner in which human operators can control their vehicles (spoken commands, for instance), but also the capacity for intelligence of the vehicles themselves has changed the relationship between the human and the system. This trend will continue as flying robots are increasingly able to perform more tasks previously restricted to humans. Historically the model for human machine interfaces has been based around control. Humans were said to interact with a system by controlling it. Under such circumstances, the system was subservient to the human. The current state of affairs is that intelligent MAVs are evolving into entities that can extend the capabilities of their human partners (Czaja and Nair 2012). With respect to MAVs, the robot should be viewed and treated as a member of the team working to accomplish the stated goals of the system. This reality of the human's relationship with MAVs will only be enhanced further as each human operator becomes responsible for swarms of MAVs in the future.

Perhaps the most prominent example of MAVs advancing the state of the art in robotic flight behavior is the International Aerial Robotics Competition. It has consistently highlighted the importance of the interdependent relationship of human operators and their aerial robots to accomplish tasks. Starting in 1991, the International Aerial Robotics Competition is the longest ongoing university-based robotics competition in the world. In addition to exposing the next generation of aerial robotics engineers to the process of system design from concept to reality, the competition has always been driven by pushing the advancement of autonomous robotic technology to levels previously unattainable by governments or industry. While earlier competition challenges involved outdoor flight tasks including mapping hazardous environments, searching for disaster survivors, collecting and moving objects, locating specific buildings and their attributes, and transmitting information back to command and control stations, the most recent challenges have involved smaller flying robots that are tasked with penetrating structures and negotiating confined spaces full of obstacles with no GPS guidance. Small air vehicles are designed to penetrate openings in structures, deactivate security systems, read signs (in languages foreign to the human operators), avoid obstacles (furniture and debris), negotiate disturbances to the airspace (such as fans), and successfully retrieve and replace an object that must be located in a limited amount of time.

These advanced robots systems are fully autonomous, making decisions on their own with no human operator control inputs, and are typical of the relationship between humans and their MAVs because they assist the human to accomplish a task that cannot be achieved without both entities working together. An attribute typical of all International Aerial Robotics Competition missions has been using unmanned technology to enhance the capabilities of humans by going places and doing things that are either not possible or too dangerous for humans to attempt. In this way,

the missions when accomplished become excellent case studies exemplifying the relationships of man-machine systems that can extend the capabilities of humans.

For a detailed examination of some considerations that can assist one to create, test, and evaluate autonomous systems that are designed to extend human capabilities and not merely mimic human attributes, the author recommends *Ten Challenges for Making Automation a “Team Player” in Joint Human Activity* (Klein and Bradshaw 2004). This essay highlights the importance of focusing research objectives on promoting healthy human robot teamwork, not merely on how to make systems more autonomous.

54.6 Automation, Situation Awareness, and Workload

It has been established that human operators and MAVs should be thought of as team members working toward the satisfaction of mission goals. But how does one determine what should be an automated function and what should not with MAV systems? Traditionally, designers might reference a Fitts List (an exhaustive list outlining the abilities of humans against the abilities of machines) to attain an overall understanding of the types of activities humans excel at versus the types of activities at which machines excel. However, one should do so with caution, making sure that by examining individual activities their interrelation as a whole within the system is not overlooked.

There are several primary reasons why a function might be considered a good candidate for automation. The first is when a task is considered too dangerous for a human to attempt without resulting in bodily harm or death. Another is when the nature of a task is something at which humans do not excel. A final example is when the safe and successful operation of a system requires a reduction in the workload of the human operator. To achieve optimal system performance, the capacity of the human to take on tasks successfully as well as the system's limitations must ultimately direct which functions should be allocated to which team member and when (Hopcroft et al. 2006).

Recent studies in the area of workload and adaptive automation have revealed that automation should be thought to redistribute workload, not reduce it (Vidulich and Tsang 2012). Too much automation on the other hand could lead to operator boredom, which could separate the human operator from the current state of the system resulting in difficulty for the human if they had to take over control to mitigate unusual crisis events. The concept behind adaptive automation is to implement automation that negotiates the fine balance between appropriate workload levels for human operators (the author advocates 75–80% workload as a rule of thumb) without inhibiting situational awareness (SA) (Vidulich and Tsang 2012). Particularly when human operators are monitoring MAVs in flight, it is essential that ground control stations provide solid feedback in real time lest the operator lose track of actions taken by the vehicle resulting in incorrect mental models of the system state. Such a situation enhances the likelihood of errors and reduces human

trust in automation. Conversely, human trust in automation might be too high due to poor feedback resulting in less attentive monitoring and the overlooking of error (Hopcroft et al. 2006).

Advocates of adaptive automation prescribe that predetermined fixed (or static) allocations do not compliment complex dynamic systems because individual factors can rapidly and unexpectedly change confounding human operators (Vidulich and Tsang 2012). This is a particularly important observation for MAVs deployed on the battlefield where the operational environment can quickly change not just in terms of threats from enemy combatants but also from environmental changes such as sandstorms or blinding rainfall.

Since MAV operators on the battlefield may be involved in direct and supervisory control of aerial robots during a firefight, one should expect that the cognitive workload demands are enhanced greatly as a variety of stressors have the opportunity to overwhelm the operator (Parasuraman et al. 2009). In such cases, automation should be employed effectively to support the human's performance with the system. Limitations such as mistrust in automation, a poor balance of workload between the human and the system, overreliance, and SA reductions are just a few of the challenges that can prevent automation from successfully aiding the warfighter in such high-stress operational environments (Parasuraman et al. 2009).

Dynamic environments such as battlefields are excellent case studies for why static function allocations are not ideal. The solutions to the limitations the warfighter faces in the operational environment with MAVs can be addressed with proper adaptive automation where conditions in which the system intervenes are not fixed but rather avail themselves at the appropriate time depending on the context of the battlefield environment. This sort of context-sensitive system assistance can be triggered by mission events, human operator performance levels, or even the human operator's physiological status (Parasuraman et al. 2009). For a further discussion of adaptive automation, reference *Adaptive Automation for Human Supervision of Multiple Uninhabited Vehicles: Effects on Change Detection, Situation Awareness, and Mental Workload* (Parasuraman et al. 2009). It encompasses a more thorough examination of the means of deploying effective adaptive automation with unmanned systems in general for warfighters on the battlefield.

To fully understand the relationship between SA and workload for warfighters remotely piloting vehicles on the battlefield, human operators who have sent MAVs inside of structures that they cannot see, or human operators who are piloting MAVs that have left their line of sight outdoors, a brief discussion of what SA entails is warranted. The author's description of SA offered below is designed to familiarize the reader with the basic concepts of SA, which is a topic that has been widely researched and written about. For a more full discussion of SA and the common challenges associated with it, reference *Situation Awareness* (Endsley 2012).

SA is a human attribute derived from cognitive processes and is therefore not an attribute of computers. Endsley's widely accepted definition of situational awareness is "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the

near future” (Endsley 1995). Using this definition, one must accept that situational awareness has three vital elements (perception, understanding, and prediction) and that each of them must be applied to exact circumstances.

Each of the three elements of situational awareness (perception, understanding, and prediction) requires different levels of human operator skill. Perception relies on selective attention, understanding relies on working and long-term memory, and prediction encompasses plan generation based on processed cues. Operator expertise is a major factor in SA (Endsley 2012). Applying the elements of situational awareness to the task of MAV flight monitoring, evaluators should expect to consider the following human tasks:

1. The assimilation of new information into preexisting knowledge bases within the human operator’s working memory
2. The perception of relationships and their significance
3. The projection of what will occur in the future given current information so that appropriate actions can be taken

It is being suggested that evaluations studying ground control station effectiveness address each of the three tasks in order by representing information so that it can be comprehended, depicting relationships among stimuli and integrating with heuristics to aid the human in making predictions about future system states. Human cognition naturally attempts to organize and make sense of visual and auditory stimuli. Human cognition is attuned to spatial and temporal relationships. Specifically with regard to perception, the human mind organizes stimuli into sources and streams that have identity and continuity over time (Endsley 2012). Harnessing these tendencies to create appropriate system feedback and data presentations to the human operator is important. In assessing successful user outcomes, it is important to highlight that good SA is not the same as good performance. Good SA is measured as appropriate and timely responses to unexpected events taking place within the system being studied.

An appropriate framework that considers the team members within a man-machine system working toward the accomplishment of mission goals must be adopted if one is to effectively manage the issue of automation with varying degrees of SA with a MAV system. One such framework has been proposed, which underscores the importance of considering which team members need awareness of other team members’ states. Using the warfighter as an example, human operators need information about the MAV’s state, humans need to know about each other (other soldiers involved with the mission), the MAV needs to know about the human’s state, and MAVs need to know about each other (Drury and Scott 2008). The human-UAV awareness framework should “accommodate the asymmetrical information needs of people and UAVs, be independent of any particular instantiation of a UAV, and be specific to the types of information needed in the UAV domain” (Drury and Scott 2008). While geared toward UAVs at large, the framework is well suited to MAVs and even refers to specific MAV platforms. The framework does well to address a common pitfall of SA analyses which is to paint an entire system’s human interface as either having good or bad SA because it considers a deeper level analysis of the

types of awareness various agents exhibit and require within a system in the context of levels of automation (Drury and Scott 2008).

As stated earlier, HFE is concerned with generalization and prediction. One challenge worth noting with regard to conducting workload and SA analyses on MAV systems is that generalizing can be challenging. In the example of the warfighter, while one can simulate weather effects to some degree, it is not possible to simulate the stresses of live fire from a battlefield into one's HFE evaluation protocol when studying a MAV system. Therefore, the actual levels of workload and stress observed during system evaluation cannot be considered to mirror what actual conditions would be. This reality is furthered by individual differences in human operator's skill and SA needs in a given situation.

Another generalization issue associated with MAVs is that one should be careful not to generalize workload assessments conducted on large-scale UAV platforms to MAV platforms because the concept of operations and the manning concepts are different. For instance, researchers at the Georgia Tech Research Institute have conducted workload analyses of large-scale high-altitude UAV platforms to determine the optimal crew size and workload levels under varying conditions for realistic operational scenarios (Georgia Tech Research Institute 2008). It would not be appropriate to generalize their findings about crew size and workload to a MAV because MAV flights are shorter in duration and typically do not involve handoffs of control as one would expect to see with a crew shift in order to support a large system for days at a time. Human operators within the large control centers serving UAVs such as Global Hawk find themselves in a more routine work cycle than the warfighters launching ISR missions with backpack MAVs from dynamic battlefields to see what is over the next hill. Depending on the MAV platform, soldiers may find themselves handing control off to one another to accomplish a mission, but this would not be because of operator fatigue like in large command and control centers. Two elements of workload that are often overlooked are what other routine tasks an operator was already involved in prior to the introduction of a new set of tasks for the MAV system, and how the introduction of the MAV system effects other agents (maintainers for example). Conducting a detailed task/job analysis as part of a good HSI plan should mitigate such an error.

54.7 Test, Evaluation, and Training

A unique attribute of MAV platforms distinguishing them from their larger unmanned counterparts is their low risk in terms of testing and evaluation. The testing of MAV platforms begins in wind tunnels but progresses to outdoor test flights much more rapidly than larger platforms that can ill afford costly crashes (Michelson 2008). AeroVironment's Wasp MAV with a radius of just 5 nm was selected for Disruptive Technology Fund by the Navy which, among others things, specified a cost goal of 5,000 USD per vehicle (United States Department of Defense 2005). Using the Wasp as an example, testing a vehicle with this price point is a very

affordable prospect. For perspective, the popular large platforms being deployed today cost well over 10 million USD. At such a nominal cost in comparison, one could crash thousands of Wasps before equaling the cost of one of those systems.

The affordability of test flights with MAVs is an advantage to the platform that is uniquely suited to its needs. Assessments of MAV performance during flight tests is generally completely reliant on stored information from testing instrumentation because the vehicle's responses to control prompts become unobservable to operators on the ground. Fine-tuning stability can often be a trial and error process (Michelson 2008), and therefore the ability to crash early test flights with fewer financial consequences is an advantage of the platform. The often-repeated mantra of "crash early, crash often" can be adopted with MAVs, allowing designers and researchers to conduct a higher number of test flights early on in the design process than with larger-scale unmanned systems.

Although there are advantages to testing with the MAV platform in comparison to larger craft, there are drawbacks. Perhaps one of the most challenging drawbacks is how one handles test instrumentation. As mentioned previously, MAVs generally have to rely on test instrumentation onboard to store flight data. It is often difficult to outfit MAVs with this special test instrumentation, and given that the majority of them operate with 50 % of the gross takeoff weight dedicated to propulsion and energy and the other 50 % to airframe and payload (Michelson 2008), some MAVs have very little payload to spare. The test instrumentation needs to be unique in size and weight, which has numerous implications on its cost, power, and versatility. In this way, MAV test instrumentation exists at the cost of fuel and endurance.

Because these platform-specific challenges to test and evaluation are unique, program managers need to have tasked HSI practitioners to address human concerns for each of the domains of HSI. Testing and evaluation falls within the bounds of the HFE domain because it deals not only with the satisfaction of design criteria but also with making sure that systems operation, maintenance, and support are appropriately suited to the capabilities and limitations of human operators and maintainers of the system (Air Force Human Systems Integration Office 2009). Due to the technical and manning requirements specific to MAVs, one should never attempt to reuse or adjust an existing test plan for a larger system and apply it to a MAV system. Doing so could result in major deficiencies in the test plan resulting in key attributes not being tested appropriately or at all.

When considering the training and selection for UAV operators, it is important to note that there are no common standards across the branches of the U.S. military. The Air Force restricts UAV pilots to military pilots only, while the Navy, Marines, and Army require a pilot to have only a private pilot's license. While human factors research has shown that appropriate levels of positive knowledge transfer is possible from manned flight to UAV control, further research is required to ascertain whether manned flight experience should be required (McCarley and Wickens 2004). These questions surrounding training standards and qualifications become more complex for MAVs where the desired model may be to place the common foot soldier in command of one or many vehicles. Further analysis, and ethnographic research,

for instance, should be required to determine what levels of training is appropriate for the piloting of MAVs from the battlefield. Manned flight experience may prove to be unnecessary as novel approaches to the command and control of MAVs are developed. Ideally, ground control station interface designs should incorporate good usability attributes that enhance the operator's ability through good affordances to remotely pilot vehicles without extensive prior flight experience.

54.8 MAV Ground Control Station Considerations

If the design of a new system or product is to be successful, a considerable amount of time must be spent trying to understand who the users will be and what their needs are. Failure to do so will result in products that do not accommodate their intended users or their environments. One of the greatest pitfalls for designers is to design for themselves and fail to consider their actual users. Ground control stations for MAVs are more mobile in nature compared to other UAV platforms. Many of them are carried in backpack rigs by soldiers or deployed from trunks of vehicles. Further, a real possibility exists that MAV ground control stations will be used in harsh outdoor battlefield environments. Although incorporating solid user interface design principles into ground control station concepts, such as element recognition over recall, clear affordances, and continuous feedback, is crucial for successful outcomes in general, this section is concerned with addressing a few of the unique challenges that MAV ground control stations have to overcome in order to be successful.

The most popular types of ground control stations serving MAVs are laptop-style designs. These come in many forms, and some of them are packed into ruggedized plastic boxes to prevent damage to system components. An advantage to the laptop style as a design is that training can be reduced because operators most likely already have familiarity with laptops in general. This style system is also advantageous because it can be rapidly deployed if necessary.

Recently, there has been an interest within industry to develop universal ground control stations that promote interoperability. Achieving interoperability increases the efficiency of MAV systems and capabilities of military units to share information. Designers should pay attention to preexisting standards and requirements for interoperability established by various branches of the armed forces as well NATO's standardization agreement (STANAG) when developing their own ground control station requirements.

Even the best interoperable ground control station designs are subject to degradation under certain conditions. In bright urban environments or desert battlefields, direct sunlight can render a ground control station's display useless. While hoods and shields can be helpful, they come at the cost of safety to the operator. If a MAV operator has his/her head buried in a display to interpret it or to maintain direct control over the vehicle, they can no longer observe their immediate physical surroundings. On battlefields or other environments where threats to the operator may abound, flying a MAV can be a high-risk activity. Minimally, the operator's

colleagues will have to guard the operator. In other situations, the operator may be able to operate the MAV from within an armored vehicle, offering protection as well as reduction in direct sunlight on the display. Designers should consider how obvious their ground control station designs make an operator to potential enemies in order to avoid increasing their likelihood of becoming a special target. Therefore one should not discount the manner in which the introduction of MAVs at the platoon level impacts the operators' preexisting job descriptions. Accounting for HSI early in a program (during the development of the concept of operations) can deter design induced manning and safety issues.

There are other environmental concerns apart from sunlight. Ground control stations can fall victim to excessive heat, cold, moisture, sand, and dust to name a few. These sorts of design considerations are often addressed in standards and guidelines. Designers should reference these when making sure that their hardware is appropriately engineered to handle these sorts of threats to functionality. What might be overlooked however is that the human operator interacting with the ground control station is also subject to the adverse effects of those elements. Soldiers, for instance, may not have the luxury of choosing their operational environment, and their performance can be limited due to these elements degrading their sensory and physical abilities. As mentioned previously, proper human interface design principles can help to reduce these negative effects by ensuring that displays and controls are simple to use.

It is important for designers to remember that the human operator while remotely piloting a MAV from a ground control station is separated from the sensory inputs that pilots receive in manned aircraft. This means that displays showing flight information need to be easy to read and operators should be able to trust them because their physical bodies are not actually feeling the effects of the forces that the vehicle is. For this reason, one should never assume that findings from studies about workload and cockpit control layout are scalable from manned aircraft to MAV ground control stations. The MAV ground control station should be considered as a unique entity with its own challenges in terms of workload and design layout.

While a pilot's inability to feel the forces of flight can be seen as a disadvantage, it can also be seen to enable the human operator, who may be able to remain calmer in riskier situations where his/her physical well being is not tied to the fate of the MAV. Designers might consider implementing haptic feedback to augment the physical effects that the MAV may be encountering. This can greatly enhance flight condition feedback to the operator.

The display of ground control stations should not overburden a human operator's mental resources. Selective attention and working memory are required because these two processes are particularly important to enhancing SA (Vidulich and Tsang 2012). An important consideration for MAVs that will be remotely piloted is the viewpoint that the display will show the operator. There are advantages and disadvantages to choosing an egocentric (one where the viewer sees what they would see as a pilot physically sitting in the vehicle) versus exocentric (a view from directly behind the vehicle) display, and one should consider the tradeoffs carefully. From one point of view, the environment is moving and the vehicle is

seen to be stationary, and from another, the environment is fixed and the vehicle is moving. An egocentric display may be preferable for obstacle avoidance, while an exocentric display may be better for acquiring an overall awareness of the terrain and environment (Vidulich and Tsang 2012). When designing for certain applications, it may be possible and advisable to create a display that affords the user both views simultaneously or the ability to switch easily. Whichever display type is selected, the design should support the human by accounting for his/her cognitive resources.

Another consideration for designers of ground control stations is the design of controls. Designing for gloved operation is normally a good idea for joysticks, buttons, and latches. Any testing and evaluation of a system should reflect as closely as possible the actual operational environment in which the system will be deployed, and therefore test participants should be outfitted with all their equipment for an appropriate context including gloves or other elements which could inhibit mobility or fine motor control required for delicate articulation. For some applications, the adherence to certain MILSPEC guidelines and standards might be required with regard to design features such as gloved operation, button size, and spacing.

In certain environments where the warfighter will be launching and commanding MAVs, light and noise discipline may hinder ground control station effectiveness. Systems that rely on light and audio to function are poor designs for this application. One should not design command and control to be limited to a single type of control input such as speech recognition. Alerts and alarms should not be audible in such conditions and displays should be designed to consider the potential for immediate muting. The necessity of night vision integration may also prove useful in certain applications. Another element to consider for the battlefield is that if the recognition of audible alerts and alarms is vital to proper system use, they can easily be muffled by the sounds produced by a firefight.

54.9 Ruggedization

The primary attribute of MAVs that distinguish them from other man-made flying things is their small size. This means that MAVs are more fragile than other UAVs, and how they transported and deployed in unforgiving environments such as the battlefield is worthy of consideration. Many MAVs are designed to fit into protective bags and cases that can be attached to the warfighter's loadout. Many MAVs are designed to be packable within the standard Modular Lightweight, Load-carrying Equipment (MOLLE) system.

Providing bags for compatibility with common equipment attachment methods is desirable but does not guarantee the level of protection that may be necessary to protect the smallest and most fragile MAVs. In the future, MAVs are going to continue to shrink in size, and it stands to reason that their fragility will continue to increase. This reality presents a challenge for the future as manufacturers of military components have had difficulty in the past keeping up with the speed of the evolution of commercial off-the-shelf components. Components get smaller, faster, and more rugged, while military systems can get bogged down in expensive

and time-consuming tests to make sure they meet military standards (Military and Aerospace Electronics 2004).

Applied Research Associates' Tactical Mini-Unmanned Air Vehicle (TACMAV) was a novel approach to storage and carrying. TACMAV had a 20.9-in. wingspan, and its flexible wings folded up around its carbon fiber fuselage. The entire system was storable in a 22×5 in. tube that attached to a soldier's backpack and weighed 0.8 pounds (United States Department of Defense 2009).

Designers should consider not only the vehicle itself but its ground control station's protection for harsh environments. The ruggedness of a MAV system should be driven by its intended use. Systems deployed inside of structures should be seen in contrast to those deployed outdoors. Standards and guidelines can be useful to designers as they determine to what degree a system's packaging needs to repel water, heat, cold, or other elements. When examining ruggedization from an HSI perspective, one should consider the survivability domain and its interdependence on the other HSI domains.

54.10 Conclusion

Remaining mindful of the distinction between Human Systems Integration and Human Factors Engineering with regard to MAV development is pivotal to correctly accounting for the human in the design process. The uniqueness of MAVs and how they are deployed is forging a new understanding of man-machine systems as their intelligence expands the capabilities of humans in diverse operational environments. Teamwork in the future among humans and MAVs will depend on systems appropriately accounting for operator-specific elements such as workload, situational awareness, and training. Developing and implementing a Human Systems Integration Plan throughout acquisition is part of a total systems approach and will ensure that the human is considered throughout the entire design process.

While specific design choices regarding ground control station interfaces or ruggedization will vary from system to system, one must remember that Human Factors Engineering is comprised of a series of tradeoffs. The individual choices made should support the operational performance objectives set forth for a specific MAV system to accomplish its mission. The relative merits of the issues discussed in this work are driven by how a particular system is intended to be deployed. For example, MAVs intended for urban combat environments will require a different set of design criteria than ones intended for agricultural surveying.

As civilian and military operators continue to appreciate the valuable differences MAVs can make in protecting human life and accomplishing formidable missions, much of the focus moving forward will be on meeting the technical and political challenges of achieving interoperability, enhancing communications, managing system affordability, furthering autonomous behavior, increasing endurance, and integrating into national airspace. MAVs will no doubt continue to be an

increasingly popular solution for warfighters as they operate in dynamically challenging environments, and therefore the means of further integrating MAVs at the platoon level will be a concern. In a time when so much of the media attention surrounding MAVs is focused on advances in technical capability and policy considerations, one must not forget to pay attention to the needs, wants, and desires of human operators within the design space. How well designers implement a human-centered approach to MAVs in the future will largely dictate the platform's success.

References

- Air Force Human Systems Integration Office, *United States Air Force FY09 Human Systems Integration Management Plan* (U.S. Government Printing Office, Washington D.C., 2009)
- S. Czaja, S. Nair, *Human Factors Engineering and Systems Design in Handbook of Human Factors and Ergonomics*, 4th edn. (Wiley, New Jersey, 2012), pp. 38–56
- J. Drury, S. Scott, Awareness in unmanned aerial vehicle operations. *Int. C2 J.* **2**(1) (2008), pp. 1–28
- M. Endsley, Toward a theory of situational awareness in dynamic systems. *Hum. Factors* **37**(1), 32–64 (1995)
- M. Endsley, *Situation Awareness in Handbook of Human Factors and Ergonomics*, 4th edn. (Wiley, Hoboken, 2012), pp. 553–568
- Georgia Tech Research Institute, Human systems integration division (2008), www.gtri.gatech.edu
- R. Hopcroft, E. Burchat, J. Vince, *Unmanned Aerial Vehicles for Maritime Patrol: Human Factors Issues* (DSTO Defence Science and Technology Organisation, Victoria, 2006)
- G. Klein, J. Bradshaw, Ten challenges for making automation a “Team Player” in joint human-agent activity. *IEEE Intell. Syst.* **19**(6), 91–95 (2004)
- J. McCarley, C. Wickens, *Human Factors Concerns in UAV Flight* (Institute of Aviation, University of Illinois, Urbana-Champaign, 2004)
- J. McMichael, S. Francis, Micro air vehicles – toward a new dimension in flight, federation of American Scientists (1997), http://www.fas.org/irp/program/collect/docs/mav_aovsi.htm. Accessed 1 May 2012
- R. Michelson, Test and evaluation of fully autonomous micro air vehicles. *ITEA J.* **29**(4), 367–374 (2008)
- Military and Aerospace Electronics, Rugged Computers Become Everyday Battlefield Equipment **15**(1), 22 (2004)
- R. Parasuraman, K. Cosenzo, E. De Visser, Adaptive automation for human supervision of multiple uninhabited vehicles: effects on change detection, situation awareness, and mental workload. *Mil. Psychol.* **21**(2), 270–297 (2009)
- United States Air Force, *Air Force Human Systems Integration Handbook* (U.S. Government Printing Office, Washington D.C., 2009)
- United States Department of Defense, *Unmanned Aircraft Systems Roadmap 2005–2030* (U.S. Government Printing Office, Washington D.C., 2005)
- United States Department of Defense, *Unmanned Systems Integrated Roadmap 2009–2034* (U.S. Government Printing Office, Washington D.C., 2009)
- United States Department of Defense, *Defense Acquisition Guidebook* (U.S. Government Printing Office, Washington D.C., 2012)
- M. Vidulich, P. Tsang, *Mental Workload and Situation Awareness in Handbook of Human Factors and Ergonomics*, 4th edn. (Wiley, Hoboken, 2012), pp. 243–273
- C. Wickens, J. Lee, Y. Liu, S. Gordon-Becker, *Introduction to Human Factors Engineering* (Prentice Hall, Upper Saddle River, 2003)

Shigeru Sunada, Hao Liu, Hiroshi Tokutake, and Daisuke Kubo

Contents

| | | |
|--------|---|------|
| 55.1 | Introduction | 1330 |
| 55.2 | Flapping Wing | 1332 |
| 55.2.1 | A Hummingbird-Sized Flapping-Wing MAV | 1333 |
| 55.2.2 | The MAV's Flexible-Wing Kinematics | 1335 |
| 55.2.3 | The MAV's Flexible-Wing Aerodynamics | 1336 |
| 55.3 | Rotary Wing | 1344 |
| 55.3.1 | Ducted-Fan UAV | 1344 |
| 55.3.2 | Quadro-Rotor UAV | 1345 |
| 55.4 | Tail-Sitter MAV | 1346 |
| 55.4.1 | Sato's Spherical Air Vehicle | 1347 |
| 55.5 | Flow Sensor on the Wing Surface Mimicking Hairs on an Insect's Body | 1348 |
| 55.6 | Effects of Flexibility and Corrugation on Wing Performance | 1353 |
| 55.7 | Conclusion | 1356 |
| | References | 1356 |

S. Sunada (✉)

Department of Aerospace Engineering, Osaka Prefecture University, Sakai, Osaka, Japan

e-mail: sunada@aero.osakafu-u.ac.jp

H. Liu

Graduate School of Engineering, Chiba University, Chiba, Japan

Shanghai Jiao Tong University and Chiba University International Cooperative Research Center (SJTU-CU ICRC), Shanghai, China

e-mail: hliu@faculty.chiba-u.jp

H. Tokutake

Faculty of Mechanical Engineering, Kanazawa University, Kanazawa, Ishikawa, Japan

e-mail: tokutake@se.kanazawa-u.ac.jp

D. Kubo

Aviatios Program Group, Japan Aerospace Exploration Agency, Tokyo, Japan

e-mail: kubo.daisuke@jaxa.jp

Abstract

MAVs (micro air vehicles), with a maximal dimension of 15 cm and nominal flight speeds around 10 m/s, operate in a Reynolds number regime of 10^5 or lower, the same as that of most natural flyers, including insects, bats, and birds. Due to the light weight and low flight speed, the MAV's flight characteristics are substantially affected by environmental factors such as wind gust and demonstrate distinguished features with fixed, rotary, and flapping wings. Like natural flyers, the wing structures of MAVs are often flexible and tend to deform during flight. Consequently, the aero/fluid and structural dynamics of these flyers are closely linked to each other, making the entire flight vehicle difficult to analyze. In this chapter, a prototype bio-inspired flapping MAV with flexible wings is described with a specific focus on the flexible-wing aerodynamics. The flapping-wing MAV has a weight of 2.4–3.0 g and a wingspan of 10–12 cm, which is comparable to hawk moths and hummingbirds. The MAV's flexible-wing aerodynamics is analyzed by combining an *in-house* computational fluid dynamics (CFD) method and wind tunnel experiments (EXP). In addition, fixed-wing and rotary-wing MAVs with elements that enable the miniaturization of an aerial vehicle will be introduced.

55.1 Introduction

A large earthquake occurred in the Tohoku area of Japan on March 11, 2011. A devastating tsunami caused by the earthquake then hit the area, destroying all the power supplies at the nuclear power plant at Fukushima. The plant subsequently lost the ability to cool down the reactor core. The building covering the reactor vessel was destroyed by the explosion of H_2 , and a great amount of radioactive materials were released from the power plant. Various methods have been undertaken to clean up after this accident. Acquiring visual information of the power plant from a high altitude using a small aerial vehicle is instrumental in this regard. The Global Hawk (Northrop Grumman Corporation), an airplane of the Fuji IMVAC Corporation, and the T-Hawk (Honeywell Corporation), a ducted-fan air vehicle, were used for this purpose. A photograph taken by Air Photo Service, Inc. by using the Fuji IMVAC Corporation airplane is shown in Fig. 55.1. The importance of unmanned aerial vehicles was confirmed by the activities of these meter-sized aerial vehicles.

Smaller aerial vehicles, in particular, the micro air vehicle (MAV), have also been studied and developed. MAVs are now the subjects of an active and well-integrated area of research, attracting participation from a wide range of talent. With a maximal dimension of 15 cm and nominal flight speeds of around 10 m/s, MAVs are desired for performing missions such as environmental monitoring, surveillance, and assessment in hostile situations. MAVs normally operate in a Reynolds number regime of 10^4 – 10^5 or lower, the same as that of most natural flyers, including insects, bats, and birds. The prominent feature of the MAVs' aerodynamics, in general, is characterized by a large-scale vortex flow structure and, hence, is highly unsteady



Fig. 55.1 Overview of the nuclear power plant at Fukushima after the earthquake on March 11, 2011 (Courtesy of Air Photo Service, Inc.)

(Shyy et al. 2007). Further, with the flexible wing structures and wing deformation of MAVs, the aero/fluid and structural dynamics of these flyers are closely linked to each other, making the entire flight vehicle difficult to analyze and design. In the past decade, there has been a remarkable increase in the research and development of MAVs, and numerous vehicle concepts, including the fixed wing, rotary wing, and flapping wing, have been proposed (Mueller 2001; Platzer et al. 2008; Stanford et al. 2008; Liu et al. 2010).

AeroVironment, Inc. has developed a very tiny aerial vehicle with flapping wings, the NANO hummingbird. Keennon et al. (2012) at AeroVironment state about the NANO hummingbird as follows: The NANO hummingbird differs from other ornithopters in its ability to control attitude in roll, pitch, and yaw during hover through the actuation of the same flapping wings it uses for propulsion, whereas other hovering ornithopters have used large tail surfaces to deflect the wake of the main wings. By performing flight and control with its wings alone, the NANO hummingbird moves one step closer to nature's flyers, which commonly demonstrate the ability to hover without large tail areas or large tail deflections.

According to online documents about the NANO hummingbird (<http://www.avinc.com/nano>), this flapper can:

1. Demonstrate precision hover flight
2. Demonstrate hover stability in a wind gust flight that requires the aircraft to hover and tolerate a 2-m/s (5 miles per hour) wind gust from the side without drifting downwind more than 1 m

3. Demonstrate a continuous hover endurance of 8 min with no external power source
4. Fly and demonstrate a controlled transition flight from hover to 11 miles per hour fast forward flight and back to hover flight
5. Demonstrate flying from outdoors to indoors and back outdoors through a normal-size doorway
6. Demonstrate flying indoors “heads-down” where the pilot operates the aircraft looking only at a live video image stream from the aircraft, without looking at or hearing the aircraft directly
7. Fly in hover and fast forward flight with a bird-shaped body and bird-shaped wings

The NANO Hummingbird has high flight capabilities, as stated above. When its flight abilities are increased, it will be used to accomplish a true mission. Is this high performance of the NANO Hummingbird caused by the flapping wings? At the present stage, one cannot deny the possibility that a tiny aerial vehicle with a configuration other than an entomopter/ornithopter has high performance flight capabilities. The following flights are required for a tiny aerial vehicle to accomplish a real mission:

1. Flights under a wide range of forward velocities including hovering
2. Flights with high acceleration and angular acceleration
3. Flights that are very robust to wind gusts

Independent of configuration, when an aerial vehicle becomes smaller, it is strongly affected by wind gusts, and the frequencies of some oscillated motions are larger, while the time constants of some converged/diverged motions are smaller. By considering these characteristics of reduced-size aerial vehicles, insect-sized aerial vehicles, which can accomplish the flights stated above, must have the following abilities:

1. Large and quick control force/moment
2. Small variations of aerodynamic force/moment when wind gusts are encountered
3. Sensors and actuators with short time-delay and high accuracy

In this chapter, recent studies and developments of the tiny entomopter/ornithopter and an aerial vehicle with other configurations that are thought to be suitable for downsizing will be introduced.

55.2 Flapping Wing

An airplane, which flies with a fixed wing, and a helicopter, which flies with a rotary wing, are the primary human-passenger aircrafts. Flying living creatures, such as birds and insects, fly with flapping wings. What type of aircraft is suitable for an insect-sized airplane? One answer might be an ornithopter, which is a vehicle with flapping wings. This is because birds and insects with very high flight capabilities have flapping wings. Sigthorsson et al. (2012) showed by their analysis that the flapping-wing vehicle is less sensitive to velocity perturbations in all aerodynamic forces and moments than the rotorcraft, except for the forward/backward

direction force. The following are the presumed advantages over an airplane with a propeller(s):

1. An ornithopter does not require a thruster because flapping wings can generate thrust as well as lift. This contributes to reducing the total weight of the vehicle.
2. A propeller generates pitching and yawing moments when it encounters side and vertical winds, respectively (Ribner 1943). This disturbs the attitude of an airplane with propellers. This does not happen when an ornithopter encounters side and vertical wind gusts.

Moreover, Iida pointed out at 2011 Annual Meeting, Japan Society of Fluid Mechanics that the lift generated by an ornithopter is independent of turbulence. On the other hand, lift on a fixed wing at a small angle of attack of 6 deg is decreased as turbulence is increased. Though more research is required, this result raises the possibility that an ornithopter is more robust to turbulence than a fixed-wing airplane.

All of the successful flapping-wing MAVs developed up to this point have flexible and light wings such as those observed in natural biological flyers (Wootton 1981), which indicates that wing flexibility is likely to have a significant influence on the resulting aerodynamics, as well as the flight stability (Young et al. 2009; Mountcastle and Daniel 2009; Shyy et al. 2010; Liu et al. 2010). In a sense, flapping flexible-wing aerodynamics is of great importance not only in uncovering the novel mechanisms of insect and bird flight but also in designing efficient flapping flight vehicles. Recently, a biologically inspired flapping-wing MAV is developed, which has four flexible wings. In this prototype MAV, the clap and fling mechanism is applied and achieved by a prototype crank system, not only because such a mechanism is observed in insect flight and thought to be capable of enhancing the aerodynamic force generation (Weis-Fogh 1973) but also because such physical interaction can affect the in-flight deformation of flexible flapping wings and, hence, aerodynamic performance.

55.2.1 A Hummingbird-Sized Flapping-Wing MAV

A prototype flapping MAV developed at Chiba University (Nakata and Liu 2011; Nakata et al. 2011) is illustrated in Fig. 55.2a. The wingspan is designed to be around 12 cm – the size observed in hawk moths and hummingbirds. The wing has a semielliptic planform that is made of a polyethylene film with a thickness of 0.03 mm and a carbon rod with a diameter of 0.3 mm at the leading edge. The wing length is 60 mm, and the wing chord length is 30 mm at the wing base. The mean chord length is calculated as 23.6 mm. Two pairs of wings are attached to a crank system that is designed to achieve the clap and fling mechanism (Weis-Fogh 1973). As illustrated in Fig. 55.2b, the wings touch thrice in a wing beat on the top by upper wings and on the side by upper and lower wings. The gearbox is fabricated by cutting the acrylonitrile-butadiene-styrene (ABS) resin so as to ensure a nice match with the motor (MK04S-10, DIDEL), the gears, and the wing hinges. A 12-tooth pinion gear made from polyacetal with a module of 0.3 is attached to the motor.

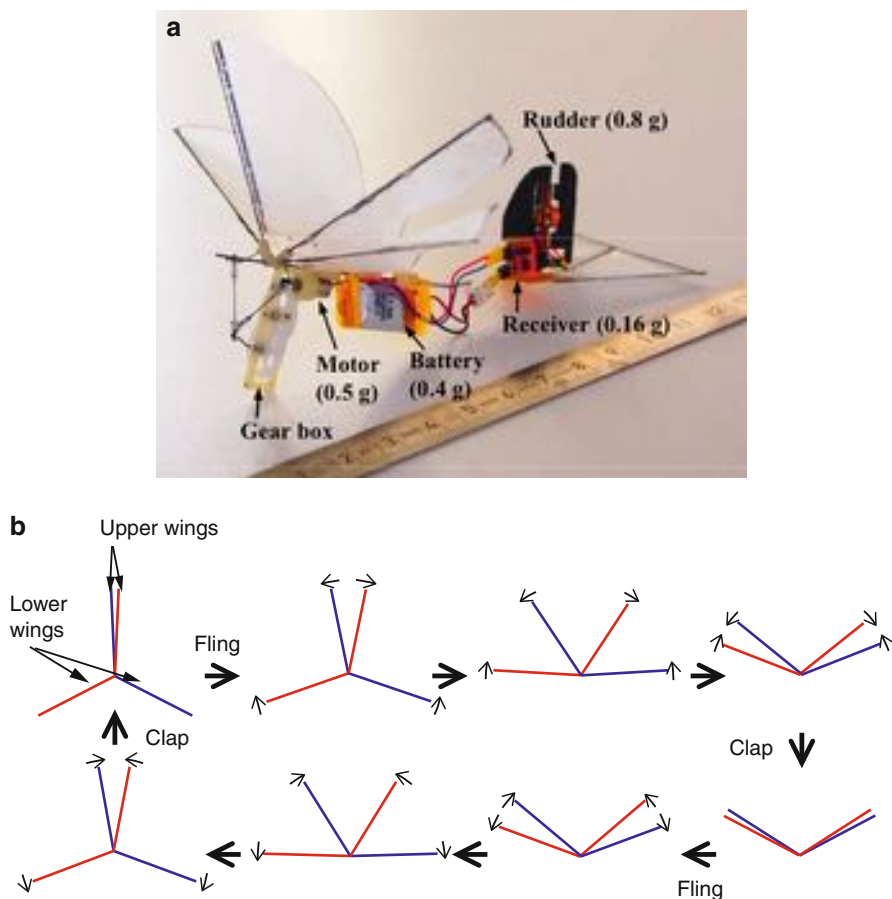


Fig. 55.2 (a) A prototype flapping micro air vehicle (MAV): a bio-inspired MAV. (b) Schematic of wing kinematics in an X-type wing MAV (viewed from the leading edge)

With a speed-reduction ratio of 60/12 teeth of the idler gear, the crank is mounted to link and actuate the two pairs of wings on the 60-tooth final gear. The gearbox system, the crank, and the wings are connected by a carbon rod with a diameter of 0.5 mm with the tail, the rudder, the receiver, and the remote control. The rudder is controlled by a magnetic actuator (hinge act, Plantraco) that moves laterally, weighs 0.23 g, and can provide sufficient control power. The remote control with infrared ray offers two channels to control both the motor frequency and the rudder angle. A rechargeable lithium polymer battery (FR30SC, Fullriver) is utilized as the power source. With all the parts mounted together, their flapping MAV weighs less than 3 g in toto and is able to fly with a time duration of up to 6 min, a maximum height over 10 m, and a region of 20×20 m.

55.2.2 The MAV's Flexible-Wing Kinematics

The flexible-wing aerodynamics is measured with a high-speed camera filming system as depicted in Fig. 55.3a–d by tracking the body-wing motions and deformations during tethered flapping flights. Three high-speed cameras (Miro, Vision Research Ltd.) with an image of 800×600 pixels at a frequency of 1,000 Hz are synchronized and operated for 2 s. Given that the MAV's flapping frequency normally varies over a range of 20–35 Hz, the recorded image sequences are able to provide sufficient temporal resolution for the current flexible-wing kinematics. Calibration is done in a global three-dimensional system by a right-angled portable calibration frame, as shown in Fig. 55.3b, c. The markers (the white dots) are arranged in an array with an interval of approximately 100 mm, which is sufficient to reasonably resolve the flexible-wing kinematics during a complete wing beat.

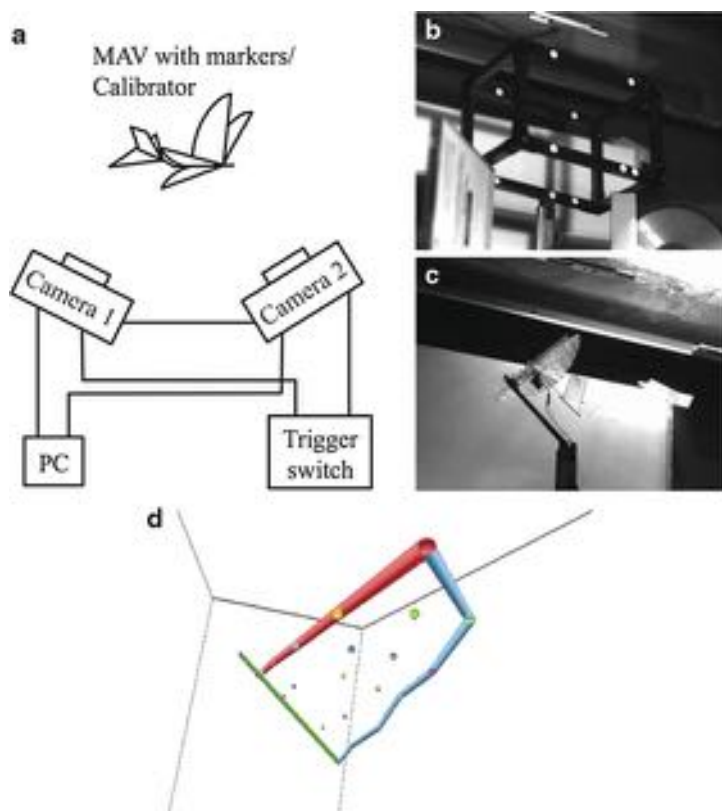


Fig. 55.3 (a) High-speed camera filming system for wing kinematics measurement. (b) A right-angled portable calibration frame. (c) An image of the flapping MAV captured by a high-speed camera. (d) Reconstructed coordinates on the wing surface based on a sequence of images in a complete beat cycle

The recorded image sequences are then downloaded to a computer, and the three-dimensional coordinates of these marked points are reconstructed utilizing commercial software, DippMotion (Ditect) (Fig. 55.3d).

A realistic kinematic model of the MAV's flexible wing is constructed by interpolating the reconstructed coordinates of the markers on the flapping wings. The displacements $\mathbf{u}(t, x, y)$ at some point of the wing (x, y) (Fig. 55.4d) are interpolated by using a function of Fourier series, such as

$$\mathbf{u}(t, x, y) = \sum_{l=0}^{n_x} \sum_{m=0}^{n_y} \sum_{n=0}^n (\mathbf{a}(l, m, n) x^l y^m \cos(n\omega t) + \mathbf{b}(l, m, n) x^l y^m \sin(n\omega t)), \quad (55.1)$$

where terms \mathbf{a} and \mathbf{b} are derived by the least square method. A realistic morphological model of the MAV's wing for computational fluid dynamics (CFD) analysis is constructed by tracing the outline of the wing planform. A uniform thickness is defined, but with elliptic smoothing at the leading and trailing edge, as well as at the tip. Figure 55.4b illustrates the computational geometric models and grid systems of the MAV. The CFD analysis is performed by introducing a symmetric plane as depicted in Fig. 55.4b under the assumption that the left and right wings move and deform symmetrically.

Figure 55.5b shows the time courses of both the upper and lower wings at a wing cross section of $0.7 R$ from the wing base (Fig. 55.5a), in which the cross sections at the stroke reversal are shown by a dotted line. Note that those markers where the coordinates cannot be reconstructed due to the overlapping of upper and lower wings are excluded (Fig. 55.5b). Since the crank system can only generate the one-axial flapping motion, the wing rotation or the feathering motion is apparently induced passively due to the wing flexibility. The angles of attacks of both the upper and lower wings are, however, observed to somehow be in good timing, contributing to the lift force production during each half stroke. The upper wing is nearly vertical to the horizontal axis at both stroke reversals, which implies a “symmetric” rotational phase of the upper wing (Dickinson et al. 1999), and hence is likely a result of the clap and fling at each stroke reversal. The lower wing keeps the attitude throughout the half stroke while showing some “phase delay” at the stroke reversal, which may lead to lowering the aerodynamic performance of flapping wings.

55.2.3 The MAV's Flexible-Wing Aerodynamics

55.2.3.1 CFD Modeling of Flexible Flapping Wing

Evaluation of the aerodynamic performance of the bio-inspired flapping MAV is conducted by using an insect dynamic flight simulator (Liu 2009; Liu et al. 2010; Nakata and Liu 2011) that is designed to integrate the modeling of realistic wing-body morphology, realistic flapping wing and body kinematics, and unsteady aerodynamics in biological flight. The CFD study is performed under the assumption of

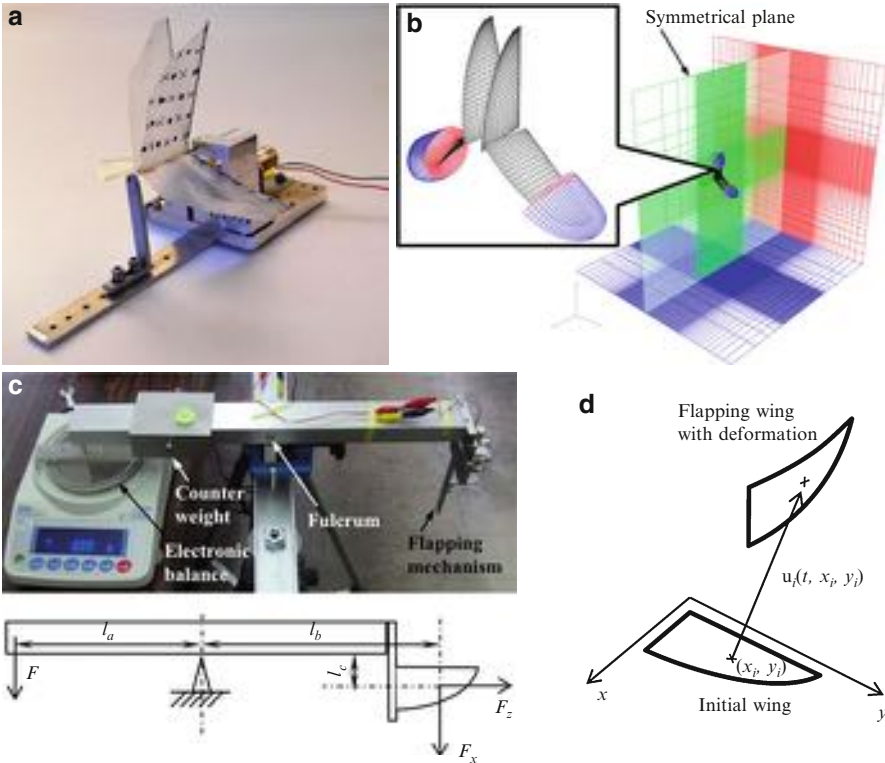


Fig. 55.4 (a) A mechanical flapping-wing MAV model. (b) A computational fluid dynamic model of MAV wings and a multiblock grid system (single wing grid: $33 \times 37 \times 19$, background grid: $89 \times 129 \times 81$). (c) Force measurement system for the mechanical flapping-wing MAV model. (d) Definition of displacement \mathbf{u} at point (x_i, y_i) on the wing surface at time t

a hovering flight condition. Given the mean chord length c_m as the reference length L_{ref} and the mean wing tip velocity in hovering flight as the reference velocity U_{ref} , which is proportional to $U_{\text{ref}} = \omega R$, where R is the wing length and ω is the mean angular velocity of the wing ($\omega = 2\Phi f$, where Φ is the wing beat amplitude and f is the flapping frequency), the Reynolds number in hovering flight can be reformed as

$$\text{Re} = \frac{U_{\text{ref}} L_{\text{ref}}}{\nu} = \frac{2\Phi f R c_m}{\nu} = \frac{\Phi f R^2}{\nu} \left(\frac{4}{\text{AR}} \right), \quad (55.2)$$

where the aspect ratio AR is in the form of $\text{AR} = (2R)^2/S$ with a wing area of $S = 2Rc_m$. Note that the Reynolds number here is proportional to the wing beat amplitude, Φ , the flapping frequency, f , and a square of the wing length, R^2 , but proportional inversely to the aspect ratio of the wing, AR.

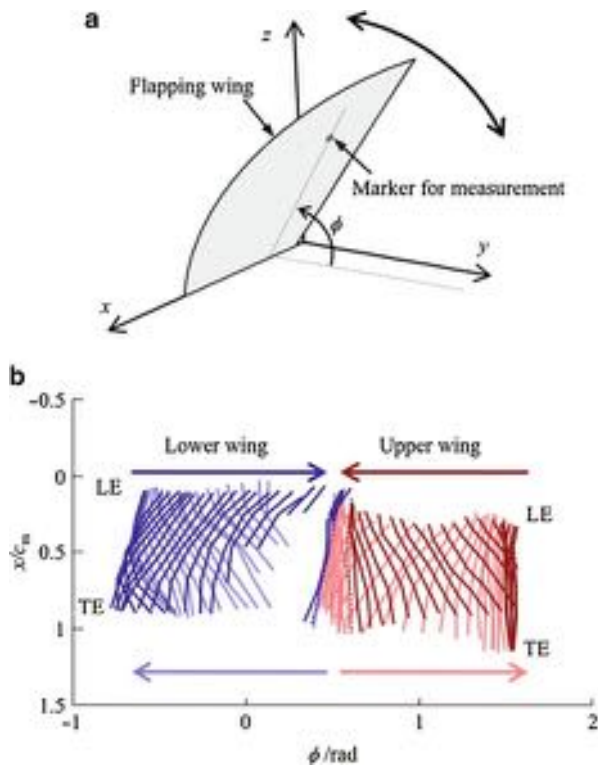


Fig. 55.5 (a) Definition of angle position ϕ . (b) Time courses of both *upper* and *lower wings* at a wing cross section of $0.7 R$ from the wing base

The reduced frequency that normally characterizes rotational versus translational speeds is defined in the case of hovering flights as

$$k = \frac{\pi f L_{\text{ref}}}{U_{\text{ref}}} = \frac{\pi c_m}{2\Phi R} = \frac{\pi}{\Phi \text{AR}}. \tag{55.3}$$

Note that the reduced frequency k is proportional inversely to the beat amplitude Φ and the aspect ratio AR of the wing. According to the measured data of the MAV’s mechanical model ($c_m = 23.6 \text{ mm}$, $R = 60 \text{ mm}$, $F = 1 \text{ rad}$, $f = 18.5 \text{ s}^{-1}$, $n = 1.5 \times 10^{-5} \text{ m}^2/\text{s}$), Re and k are calculated to be about 3,400 and 0.59, respectively.

The CFD modeling of unsteady flows around the MAV’s flexible wings undergoing flapping is performed for a single flexible-wing model in which realistic geometric and kinematic models are utilized, as described in the preceding sections. The CFD-based results show that a leading-edge vortex (LEV) and, hence, a strong negative pressure region are generated on the upper and lower wings during both half strokes (Fig. 55.6a). As observed in insect flapping flight (Ellington et al. 1996),

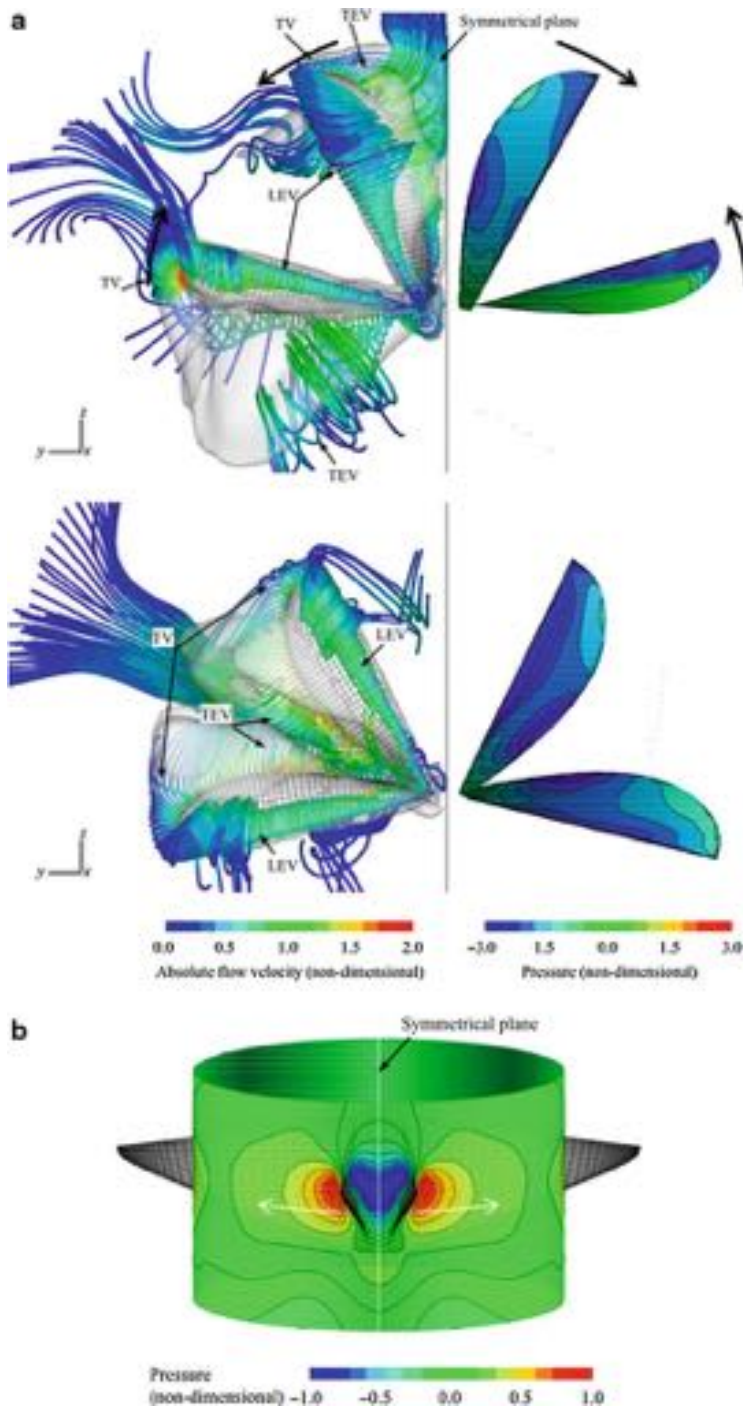


Fig. 55.6 (continued)

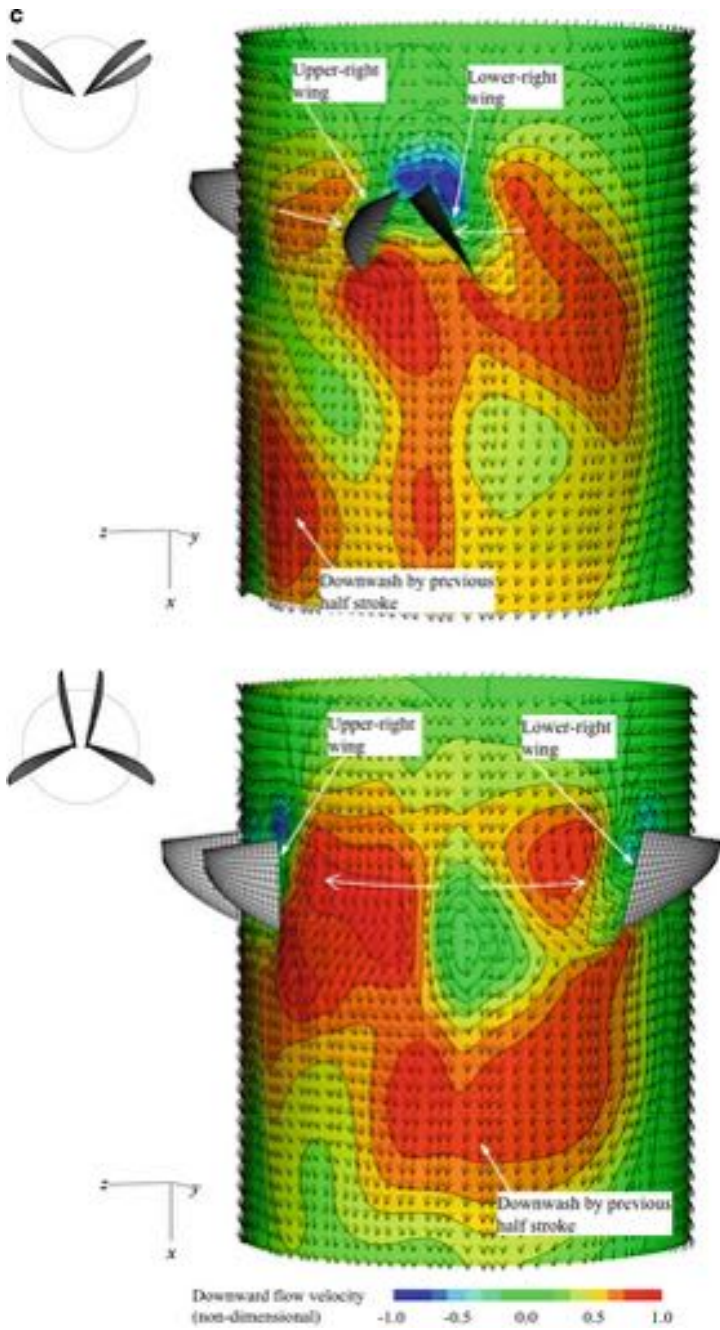


Fig. 55.6 (a) Instantaneous streamlines, iso-vorticity surface, and pressure contours on the upper surface of flapping wings at each half stroke. (b) Pressure contours on a virtual cylindrical surface at $2.0 c_m$ from the wing base. (c) Wake topologies at the end of each half stroke. Velocity vectors and contours are visualized at a virtual cylindrical surface at $2.0 c_m$ from the wing base

this LEV apparently plays a crucial role in the lift and/or thrust force production in the MAV flight. The vortex rings that are formed from the LEV, the tip vortex (TV), and the trailing-edge vortex (TEV) are also observed, showing a similar pattern to those of insect flight (Liu and Aono 2009). Obviously, strong negative pressure regions are detected between the upper right and left wings, as illustrated in Fig. 55.6b, which are likely induced by the clap and fling mechanism.

Figure 55.6c illustrates the wake topology on a virtual cylindrical surface at the end of each half stroke. An intense downwash is generated behind the path of the wings identical to the center of the vortex rings (Liu 2009). The downwash disks are connected by the clap and fling with a pitching up motion, resulting in a pronounced downwash in the far wake below. In contrast, it is interesting to see that the downwash generated by the lower wings without the clap and fling is rather weak and there is almost no high-speed zone visible downward in the far wake.

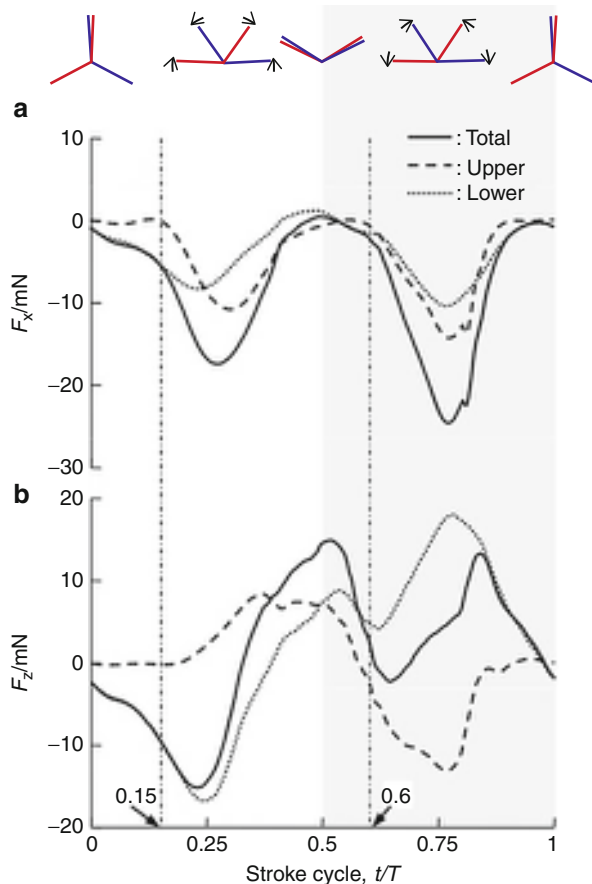
Figure 55.7 shows the time courses of aerodynamic forces generated by the upper and lower wings in a wing beat. Clearly, the upper wing generates very few aerodynamic forces until $0.15 T$ after the clap and fling (the vertical dashed line in Fig. 55.7), while the aerodynamic forces generated by the lower wing show a stable increase after the stroke reversal. However, the horizontal force F_x by the upper wing shows a subsequent rapid rise having a higher peak than the lower wing. Then, force F_x continues to decrease to a low level until $0.6 T$ after the next stroke reversal (the vertical dashed line in Fig. 55.7), while the upper wing shows a larger peak again at the half stroke. Since the present flapping-wing mechanism generates large forces at each half stroke, one peak in the total F_x is observed at each half stroke.

Furthermore, the mean aerodynamic force is calculated to be 23.3 mN, which is in good agreement with the measurement of a value of 26.46 mN acting on the electronic balance. The mean force components of F_x and F_z generated by the upper wing are -4.2 and 0.2 mN and those by the lower wing are -3.8 and 2.0 mN, respectively. The mean aerodynamic power consumed by the upper and lower wings is estimated to be 14.2 and 14 mW, respectively.

55.2.3.2 Wind Tunnel Experiments: Lift, Drag, and Flexible-Wing Kinematics

The aerodynamic performance of the present prototype MAV (Fig. 55.3b, c) and a mechanical flapping-wing MAV model (Fig. 55.4a, c) that can measure forces is evaluated by means of wind tunnel experiments. The forces acting upon the MAV undergoing the tethered flight are measured by utilizing the load cells. Figure 55.8 shows the mean lift and drag forces plotted against the body inclination angle of q_b , which varies from 0 to 70 deg. The lift force is seen to increase with increasing the body angle, the flapping frequency (with two frequencies of 10 and 20 Hz), and the wind velocity (when q_b is greater than 40 deg), which reaches a value of 25 mN at a flapping frequency of 20 Hz and a wind velocity of 1.6 m/s, very close to the MAV's weight. The drag force also shows an increase with increasing the body angle, but a negative drag force, i.e., the thrust force, is observed when the body angle is less than 50 deg. Note that the thrust force is generated merely in the case of no wind velocity and turns to the drag force at the larger wind velocity of 3.0 m/s.

Fig. 55.7 Time courses of force components (a) F_x and (b) F_z on the right wings of the four-winged MAV



One key reason that the present MAV utilizes the X-type wing (Fig. 55.2) with four wings is because it can generate more lift and/or thrust forces than a two-winged MAV. Comparison of the lift forces between the two MAV types indicates that the present four-winged MAV does generate more lift force at almost twice the lift forces of the two-winged MAV when q_b is greater than 40 deg. However, two times the lift force by the two-winged MAV with the thin Mylar wing shows a larger value than that of the present four-winged MAV.

55.2.3.3 The Clap and Fling Mechanism in Four-Winged MAVs

The flapping-wing mechanism utilizing an X-type wing with four wings is employed for the present four-winged MAV, which is inspired by the smart two-winged natural flyer, the hummingbird, but obviously has crucial differences. Though there is a great variety of biological flyers, such as insects, bats, and birds, most fly with one pair of wings. The present flapping-wing mechanism is quite unique

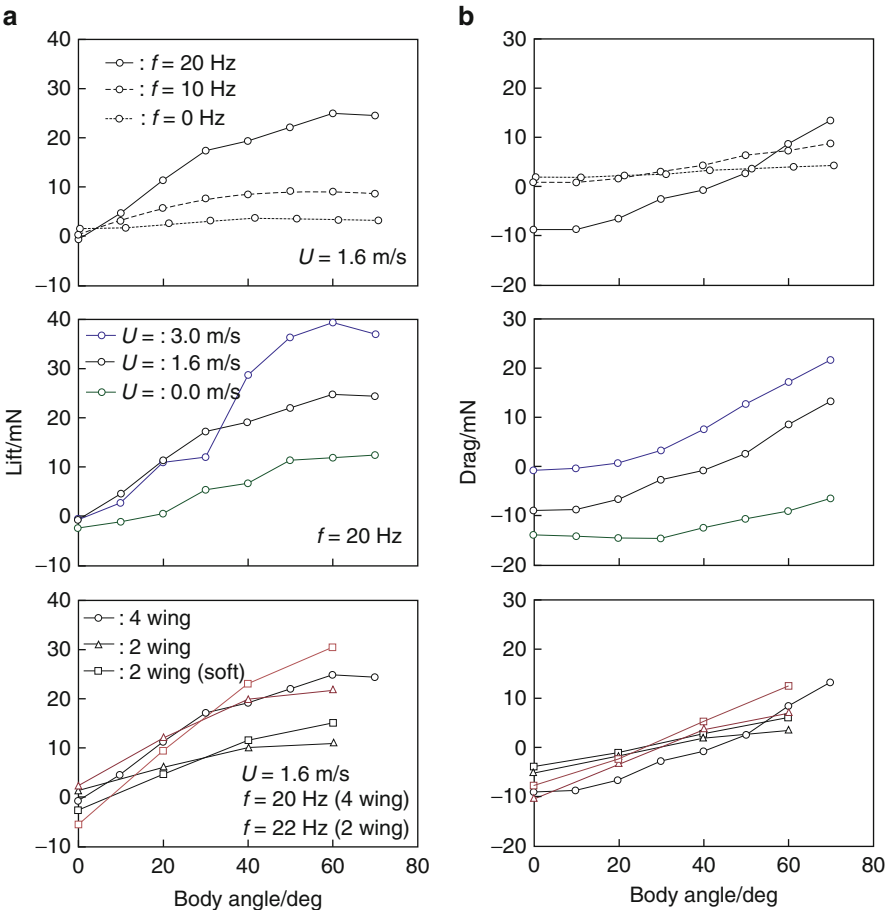


Fig. 55.8 (a) Lift and (b) drag forces plotted against body angles for four-winged and two-winged MAVs. The red line represents two times of lift and drag forces generated by the two-winged MAV

because of the multiple physical interactions in a wing beat cycle. Here, it is found that the four-winged MAV is able to generate two times greater lift force than that of a two-winged MAV at body angles larger than 40 deg where the MAV actually flies (Fig. 55.8). This implies that the present flapping-wing mechanism very likely utilizes the clap and fling mechanism effectively and hence enhances force production more than the two-winged flapping mechanism. On the other hand, it is seen that a two-winged MAV with a softer Mylar wing generates even more lift force than the four-winged MAV in the case of body angles larger than 30 deg (Fig. 55.8), which further points to the importance of the choice of an appropriate wing structure in the MAV design (Heathcote and Gursul 2007; Heathcote et al. 2008).

55.3 Rotary Wing

A lot of meter-sized rotary-wing unmanned aerial vehicles (UAVs) are helicopters with attitudes controlled by cyclic pitch. The following two types of rotary-wing UAVs without cyclic pitch control seem to be successful.

55.3.1 Ducted-Fan UAV

A ducted-fan MAV is a vehicle having a lift-augmented ducted fan driven by a reciprocating engine or an electrical motor at the center of the vehicle and aerodynamic surfaces (vanes) close to the outlet of the duct for flight control. This entails less danger to the operator and has an acoustic signature rather than a rotary-wing-type MAV because of the shrouded rotor.

In 2001, the Defense Advanced Research Projects Agency (DARPA) of the United States Department of Defense started the Organic Air Vehicle (OAV) program following the DARPA MAV program (1996–2000) (DARPA 2003). The OAV program is one of the technologies supporting the DARPA/ARMY Future Combat Systems (FCS) program. The smallest category of the FCS is a backpack-sized vertical takeoff and landing (VTOL) MAV for platoon-level operations (FCS class I). One of the contractors of the OAV program is Honeywell, which developed the “T-Hawk” (originally called the “Honeywell MAV”) ducted-fan MAV (Daly 2011). It was selected for the FCS I platform vehicle, and further development included a high performance heavy fuel engine. The U.S. Army and also the U.S. Navy awarded T-Hawk contracts to Honeywell for deployment to Iraq. The U.S. military used the T-Hawk in search missions for roadside bombs in Iraq. The T-Hawk was also used to conduct surveillance of the damaged Fukushima Dai-ichi nuclear power station in Japan in 2011.

Ducted-fan vehicles are unique in many respects: They can hover but are unlike helicopters, and they can dash at high speeds but are unlike airplanes. Much of the design intuition within the aerospace community is no longer applicable when venturing into the realm of VTOL ducted-fan vehicle design (Ohanian et al. 2010). Therefore, a wide variety of research and development has been conducted since the year 2000. Even just focusing on “micro” size, there is considerable research and development of ducted-fan MAVs all over the world (Table 55.1).

Advanced methods for improving the aerodynamic characteristics of ducted fans are being investigated. Barrett (2004) applied piezoceramic actuator technology that enables higher bandwidth control than the conventional servomotor actuator for XQ-138 ducted-fan MAVs. Bilgen et al. (2010) applied a variable camber airfoil via smart materials for the control vanes in wind tunnel tests. Ohanian et al. (2011) applied synthetic jet actuators to the duct lip and the trailing edge to control the force and moment of the duct. Colman et al. (2011) experimentally investigated a cyclic and collective pitch control of a proprotor shrouded with an asymmetric duct.

Table 55.1 Developed ducted-fan MAVs

| MAV name | Duct dia. (in.) | Gross weight (g) |
|--------------------------------|-----------------|------------------|
| XQ-138-6'' (Barrett 2004) | 6 | <378 |
| iMAV (Ohanian et al. 2010) | 6 | 567 |
| SEMi (Kubo et al. 2009) | 6 | 600 |
| RMIT Univ. (Zhao and Bil 2009) | 5.5 | – |
| XQ-138-4'' (Barrett 2004) | 4 | <115 |

Fig. 55.9 MD4-200 by Microdrones (Courtesy of Microdrones GmbH)



55.3.2 Quadro-Rotor UAV

A quadro-rotor-type UAV has been constructed by companies such as Microdrones GmbH(<http://www.microdrones.com>), Ascending Technologies GmbH(<http://www.ascotec.de>), and Dragonfly Innovations Inc. (<http://www.rctoys.com>). The MD4-200 by Microdrones weighs about 900 g with a 37-cm rotor diameter. The X-3D BL by Ascending Technologies weighs about 740 g with a 37-cm rotor diameter. Dragonfly Innovations Inc. developed a UAV with three coaxial rotors, the Dragonfly VI as well as the Dragonfly IV and V, which are quadro-rotor-type UAVs.

Figure 55.9 shows the MD4-200 by Microdrones as one example. This rotorcraft does not have a swashplate and does not need any blade pitch control. The two pairs of rotors (1,3) and (2,4) rotate in opposite directions. The gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight. Increasing or decreasing the four rotors' speed generates vertical motion. Changing the 1 and 3 rotors' speed conversely creates roll motion with lateral motion. Similarly, changing the 2 and 4 rotors' speed conversely causes pitch motion with longitudinal motion. Yaw motion is established by increasing (decreasing) the 2 and 4 rotors' speed and decreasing (increasing) the 1 and 3 rotors' speed while keeping the thrust constant. As the variation of rotational speeds of the rotors is increased, the control moments can be increased. Furthermore, the control moment can be increased by increasing the distance of the motors from the center of the vehicle. The control moment per the

moment of inertia increases with the increase of the distance of motors from the center of the vehicle. This is because the greatest weight is located at the center of a quadro-rotor-type UAV. Therefore, angular accelerations can be increased by increasing the distance of the motors from the center of the vehicle.

55.4 Tail-Sitter MAV

A tail-sitter is an aircraft that takes off and lands on its tail section, with the fuselage pointing upward. In the 1950s, several manned tail-sitter fighter airplanes were developed, and prototypes were constructed in the USA. However, these projects were abandoned because of significant control difficulty during the landing phase for a human pilot. However, tail-sitters have the advantage of eliminating the need for variable mechanisms for making the transition between hovering and cruising modes when compared to tilt-rotor and tilt-wing aircraft; therefore, this configuration is particularly suitable for small UAVs and MAVs that have severe weight constraints because of their small size.

Certain ducted-fan vehicles can be also categorized as “tail-sitter” vehicles because they are designed not only for hovering flight but also for forward high-speed dash performance. The ducted-fan MAV XQ-138 can be considered as a ducted-fan vehicle but is defined as a coleopter in the original paper (Barrett 2004). A coleopter is an aircraft having a fuselage surrounded by an annular wing, which is usually associated with a tail-sitter. Furthermore, several ducted-fan vehicles have small wings to augment cruising efficiency (Zhao and Bil 2009). Considering this, there are no explicit borders between the designs of the ducted-fan and tail-sitter MAV.

For a “fixed-wing” tail-sitter aircraft design, transitional flight – middle air speed flight between zero speed hover to high-speed dash flight – is one of the biggest challenges. Especially for small sized vehicles, the effect of wind during the takeoff and landing phase is critical. For zero wind conditions, tail-sitter vehicles hover with the fuselage pointing in the nearly vertical direction; however, in nonzero wind conditions, the vehicles must compensate the wind drag by tilting the thrust, i.e., the pitch angle of the vehicle. Therefore, the angle of attack will be significantly large under these conditions even when using propeller slipstream effects. Thus, small tail-sitter vehicles must be designed with stall constraint relaxation, such as propeller slipstream effects and high lift devices on the main wing like a leading-edge slat (Kubo and Suzuki 2008).

On the other hand, for the MAV tail-sitter design, the low Reynolds number and aspect ratio wing with propeller slipstream makes the effective stall angle of attack (AoA) significantly larger, which makes transitional flight easier. One of the examples of the fixed-wing VTOL MAVs is the “Hoverfly,” which was developed in the DARPA MAV program by AeroVironment Inc. This is a battery-powered flying wing with coaxial proprotors. Another example is the Mini-Vertigo MAV of the University of Arizona (Chu et al. 2009).

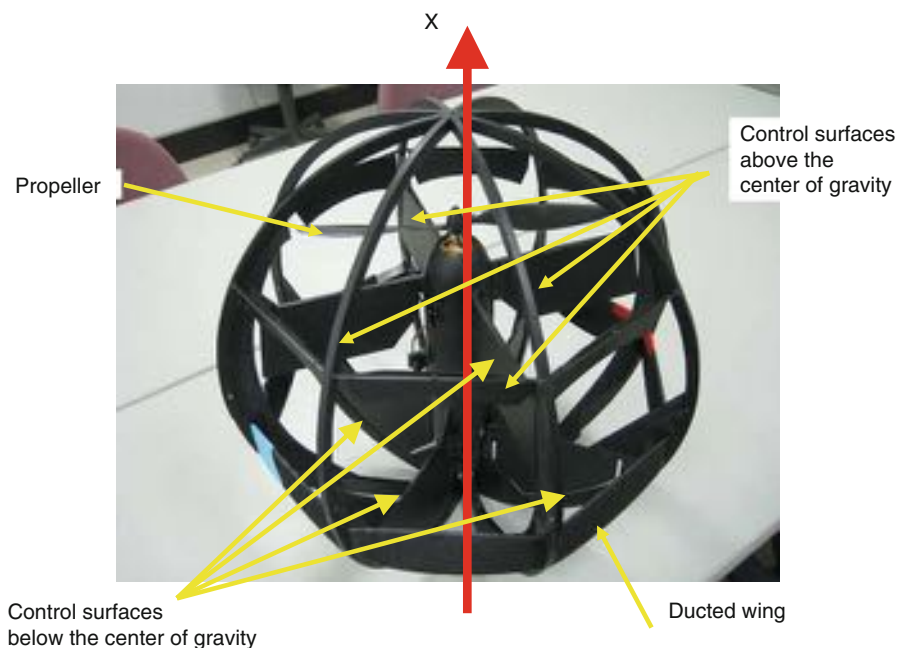


Fig. 55.10 Spherical air vehicle developed by Sato (http://www.mod.go.jp/trdi/research/youshi2010_oral1.pdf)

55.4.1 Sato's Spherical Air Vehicle

Recently, Sato developed an aerial vehicle as shown in Fig. 55.10 (Sato 2010). This vehicle is composed of a tail-sitter airplane and a spherical structure around the tail-sitter airplane. The diameter of the spherical structure is 42 cm, and the total weight is about 350 g. An ordinary tail-sitter airplane is at risk of being toppled by a side wind when it is standing. However, the present vehicle does not have this potential problem. It can roll on the ground, take off vertically, land at any attitude, and make hovering and level flights. Flight speed is between 0 and about 60 km/h, and flight duration at hover is about 8 min.

This airplane has the following characteristics:

1. The center of gravity is aft, and the airplane can stand on the ground stably.
2. The rotational direction of the propeller can be varied, and the direction and magnitude of thrust by the propeller can be controlled.
3. The four control surfaces arranged crosswise are above the center of gravity, and the other four control surfaces arranged crosswise are below the center of gravity. These control surfaces generate moments about the X, Y, and Z axes. These surfaces can generate control moment because the surfaces are in the wake of the propeller. The angle between the two sets of four control surfaces is 45 deg

about the X axis. The aerodynamic interaction between these sets of four control surfaces is reduced.

4. The aerodynamic center of the airplane is below the center of gravity. This provides stability for a level flight. The ducted wing on the spherical structure moves the aerodynamic center more aft.
5. When this airplane takes off and lands, the two control surfaces above the center of gravity act as an elevator. Then, the direction of deflection of the elevator for pitching the airplane up/down remains the same during flight and while rolling on the ground. This provides ease in controlling the vehicle. Note that the airplane rolls on the ground in the direction of the moment about the point where the airplane touches the ground, and the airplane pitches in the direction of the moment about the center of gravity of the airplane.
6. This vehicle can cling to a vertical wall.

55.5 Flow Sensor on the Wing Surface Mimicking Hairs on an Insect's Body

The flight performance of a small airplane can be expected to be improved by using the flow data around the airplane (Tokutake et al. 2011). However, conventional air data systems, such as the pitot tube and the angle of attack sensor, are too heavy to be installed on small airplanes and are not sensitive to their small-flow condition changes. For these reasons, a tiny flow sensor was developed to control the attitude of small airplanes. The flow velocities on the wing surface are measured by the sensor system, which utilizes microelectromechanical systems (MEMS) technology. This sensor system is similar to hairs on an insect. This flight control system was designed to attain stable flight at a high angle of attack with maximum lift coefficient using flow sensor output.

The MEMS flow sensor D6F-W04A1 manufactured by Omron Corporation, which detects the temperature distribution of a heated sensor head, has been improved and adopted. The MEMS flow sensor on the market was disassembled,

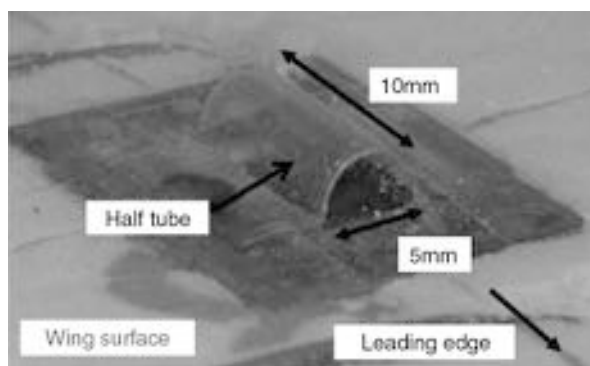


Fig. 55.11 Flow sensor on the wing surface

Fig. 55.12 Flow sensor output

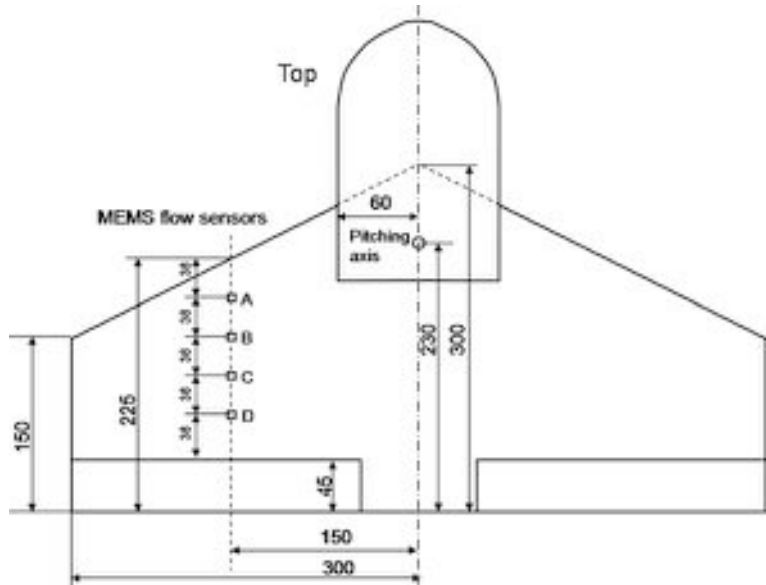
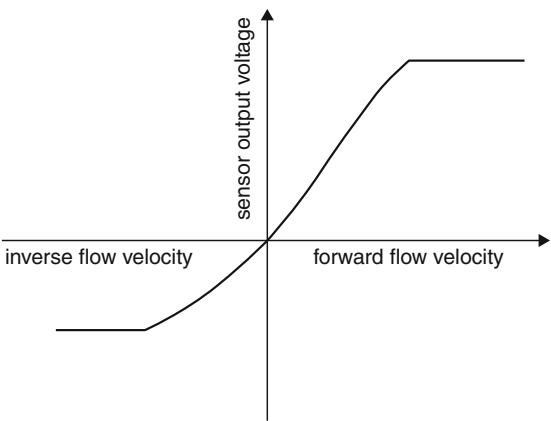


Fig. 55.13 Sensor assignments

revealing the sensor head. The sensor head is a 4×4 mm square, and it was used with a half tube cover having a length of 10 mm and a diameter of 5 mm for protection (Fig. 55.11). This MEMS flow sensor can measure the strength of the surface velocity and can detect inverse flow. The weight of the sensor is 1.3 g. Figure 55.12 shows the relationship between the flow velocity and the sensor output. Because the flow conditions at the maximum lift coefficient are of interest in these experiments, the sensor output at the maximum lift coefficient was investigated. Since these sensors are light, they can be mounted on the flight airplane model.

Fig. 55.14 Lift coefficient

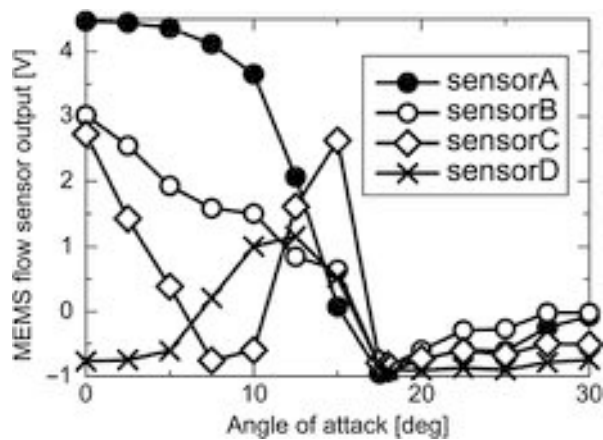
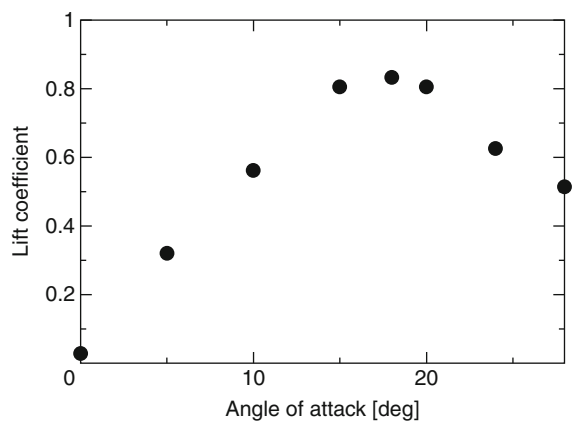
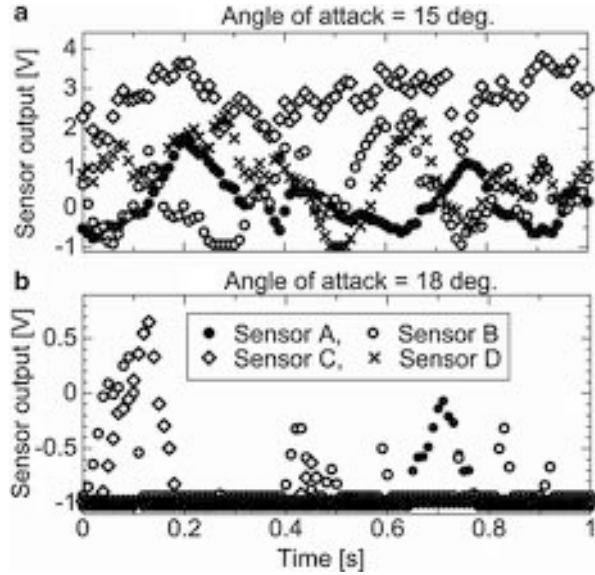


Fig. 55.15 Flow sensor output – angle of attack

A tailless small airplane was studied. MEMS flow sensors were located on the upper wing surface at the quarter span of the right wing while being aligned chordwise with the same distance in-between (Fig. 55.13).

Several static and dynamic wind tunnel tests were performed to investigate the aerodynamic characteristics. Figure 55.14 shows the relations between the angle of attack and the lift coefficients of this small airplane, which reach a maximum value around the angle of attack of 18°, and Fig. 55.15 indicates the flow sensor outputs subject to the angle of attack. Figure 55.16 shows the MEMS flow sensor output for 1 s at a static AOA of 15° and 18°. For these cases, the flow on the entire wing surface at an AOA of 18° is disturbed. At an AOA of 15°, the MEMS flow sensors detect forward flow velocity. The control requirement is then defined as maintaining the output of MEMS flow sensor C as 0. If this requirement is satisfied

Fig. 55.16 Flow sensor output at steady angle of attack



constantly, the flow condition resembles the desired one. As a result, the airplane has a maximum lift coefficient. The purpose of the controller design is to maintain the high lift coefficients. Therefore, the stabilizing controller is designed to maintain the output of flow sensor C at zero under disturbance and model uncertainties.

Flow responses around the wing are determined by airplane attitude. The dynamics between attitude input and flow sensor output are identified via dynamic wind tunnel testing. The pitch angle of the airplane was varied dynamically, and the pitch angle and the sensor outputs were measured. From the experimental results, the relations between the angle of attack and the sensor outputs were identified as a transfer function. The transfer function from the angle of attack to the output of sensor C is as follows:

$$G_a(s) = \frac{-0.02177s^3 - 8.685s^2 - 711.3s - 4.946 \times 10^{-4}}{s^3 + 229.2s^2 + 1.456 \times 10^{-4}s + 1.382 \times 10^{-5}}.$$

From this dynamic model of flow sensor output and longitudinal airplane dynamics, a generalized plant was formulated. The linearized longitudinal dynamics around a trim condition AOA of 18° and forward velocity of 6.2 m/s is as follows:

$$\begin{aligned}\dot{x}_L &= A_L x_L + B_{L1} w_L + B_{L2} \delta \\ x_L &= [u \ \alpha \ q \ \theta]^T\end{aligned}$$

$$A_L = \begin{bmatrix} -2.31 & -5.53 & 0 & -9.8 \\ 0.06 & -5.29 & 0.94 & 0 \\ 6.93 & -28.84 & -0.71 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B_{L1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_{L2} = \begin{bmatrix} -2.69 \\ -2.35 \\ -27.80 \\ 0 \end{bmatrix}.$$

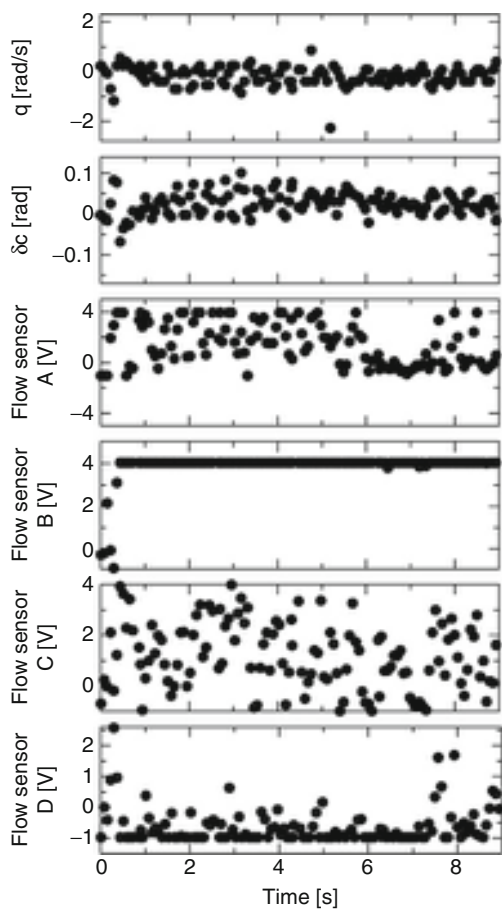
Here, u (m/s) is the forward velocity, α (rad) is the angle of attack, q (rad/s) is the pitch rate, θ (rad) is the pitch angle, and δ (rad) is the elevon deflection angle. These values are perturbed around the trim condition. W_L is an aerodynamic disturbance that accounts for the flow condition change and sensor noise. The values of matrices A_L and B_{L2} are ascertained from the static wind tunnel experiment. The actuator dynamics are formulated as a first-order lag with a time constant of 0.1 s. The airplane dynamics, actuator dynamics, and sensor output dynamics are combined. The H_∞ controller is designed for the developed dynamics so that the pitch rate responses, output of flow sensor C, and elevon deflection angle are restricted under gust input and model uncertainties.

The obtained controller was discretized and installed onto a microcomputer, and flight testing was performed. Figure 55.17 shows the constructed flight model. The flow sensors are installed at the same position as in Fig. 55.13. The airplane was flown at the trimmed condition, and the designed controller was engaged. Figure 55.18 shows the recorded flight data. From these data, the airplane is clearly stabilized, and the output of flow sensor C is maintained around zero. The sensor output near stall of the steady wind tunnel test oscillates (Fig. 55.16). Therefore, it can be said that the output of sensor C in this flight experiment was well stabilized.



Fig. 55.17 Flight model

Fig. 55.18 Flight data



55.6 Effects of Flexibility and Corrugation on Wing Performance

At a low Reynolds number (less than 10^5), a thin wing has a large lift-drag ratio (Kunz and Kroo 2001). Corrugation can make a thin wing more rigid (Rees 1975; Sunada et al. 1998). Thus, a corrugated wing can be thin, and its aerodynamic characteristics can be higher accordingly. Furthermore, Hu and Tamai (2008) pointed out that corrugation prevents large-scale flow separation and improves the flight agility and maneuverability of MAVs at high angles of attack. Figure 55.19 shows the flows around a dragonfly forewing model in a steady flow (Obata and Sinohara 2009) at a Reynolds number of 7,000 and an angle of attack of 5 deg, where the exposure time in Fig. 55.19a and b is 0.02 and 5 s, respectively. In the former picture, unsteady vortices from the airfoil can be observed; in the latter picture, a time-averaged flow pattern around the airfoil can be observed. They also

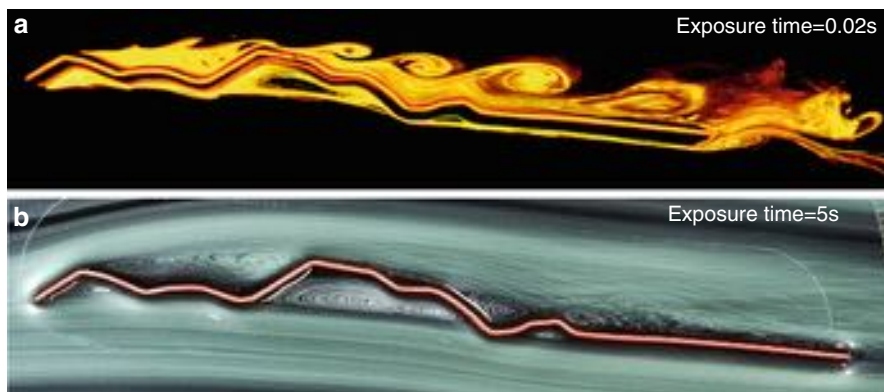


Fig. 55.19 Flow around a corrugated wing (Obata and Sinohara 2009). (a) Exposure time = 0.02 s. (b) Exposure time = 5 s

pointed out that the flow around the corrugated airfoil with acceleration is similar with that without acceleration. This implies that corrugation can suppress the effect of disturbance for airfoil characteristics, and the corrugated wing is capable to enhance the stability of a small aerial vehicle. Till now, the fixed corrugated wing is the main subject in most studies of aerial vehicles. With respect to a corrugated wing undergoing flapping motion, Meng et al. (2011) studied numerically the aerodynamic effects of corrugation by using several model insect wings, which mimic insect hovering. The results show that the aerodynamic performance of the corrugated wing model is very close to that of the flat-plate wing model. In their computational models, the corrugated wing model performs feathering motion but is rigid without wing deformation during flapping motions. Since the wing corrugation normally shows a strong relationship with wing flexibility as observed in nature, more studies are needed associated with effects of the wing corrugation on aerodynamic performance of flapping wings.

Recently, an important subject about flapping wings is the effect of wing flexibility on the aerodynamic performance of flapping wings. The wing flexibility normally leads to passive wing deformation, which may enhance aerodynamic performance, even in cases with no active control and/or with no extra power consumption. Shyy et al. (2010) gave a review on recent studies associated with the subject about a flapping wing and classified it into three categories: “chordwise-flexible-wing structure,” “spanwise-flexible-wing structure,” and “combined chordwise- and spanwise-flexible-wing structures.” Among these studies, Ho et al. (2003) compared the aerodynamic performance of a cicada wing, which is rigid in spanwise, with a flexible titanium alloy wing but without leading-edge support. They pointed out that stiffness distribution is an important factor for vortex interaction and thrust production. High stiffness is required at the outboard region

of the wing to enhance lift generation, whereas wing flexibility is required at the inward region to enhance thrust.

Heathcote and his colleagues made some experiments about a flexible flapping wing. Heathcote et al. (2004) used a flexible airfoil with plunging motion at 0 freestream velocity. They concluded in two-fold: (1) there is an optimum airfoil stiffness for a given plunge frequency and amplitude, and (2) the thrust/power ratio is greater for the flexible airfoils than for the rigid airfoil. Heathcote and Gursul (2007) used a flexible airfoil with plunging motion at zero and non-zero freestream velocity. They found that stronger trailing-edge vortices and weaker leading-edge vortices correspond to higher thrust coefficients and higher efficiencies, respectively. Furthermore, a degree of flexibility increases both thrust coefficient and propulsive efficiency. Heathcote et al. (2008) further carried out an experiment in a water tunnel to study the effect of spanwise flexibility on aerodynamic performance of a plunging flexible-wing configuration in forward flight.

Yong et al. (2009) analyzed a forward flight of locust by using a computational fluid dynamic solver and by prescribing the three-dimensional wing shape changes based on measurements. They found that the full-fidelity wing model achieved greater power economy than the uncambered model, which performed better than the untwisted model.

A computational FSI (fluid-structure interaction) analysis that is performed by coupling a CFD solver and a computational structural dynamics (CSD) solver is capable of providing detailed information on both flow fields and structural dynamics of flexible wings. Zhu (2007) made a simulation-based FSI analysis. The boundary element method, which is based upon the potential flow framework was used for fluid dynamics, and a two-dimensional nonlinear thin-plate model was used for structural dynamics. He found that (1) chordwise flexibility and fluid-driven deformation can increase efficiency and (2) spanwise flexibility and inertia-driven deformation can increase thrust.

Gopalakrishnan and Tafti (2010) made a computational FSI analysis to analyze the effect of aeroelastic cambering on flapping flight performance using a linear elastic membrane model. The unsteady aerodynamics of the flapping wing was solved using LES (large-eddy simulation) on a deforming body-fitted fluid mesh using a multiblock finite volume methodology. The analysis was carried out for a rigid wing and flexible wings with different prestress values. Flexible wings were found to have a positive and negative camber during the downstroke and upstroke, respectively. The camber was observed to enhance the lift and thrust production and hence propulsive efficiency. Moreover, it was found that the membrane with a low prestress along the chord and a high prestress along the span demonstrates the best aerodynamics and aeroelastic performance.

Nakata and Liu (2011) carried out a simulation-based FSI analysis by coupling a Navier-Stokes equation solver and a finite-element method-based structural dynamics solver, which is further applied to hawkmoth hovering with realistic flexible wings. The wing motion is composed of flapping motion and feathering motion being prescribed at the wing base. The wing is modeled as a flexible flat plate with consideration of realistic properties of material in a hawkmoth wing,

which bends in spanwise and twists in chordwise with some cambers formed as a result of the inertia and aerodynamic forces on the wing. Their results indicate that (1) a dynamic wing bending can delay the breakdown of leading-edge vortices near the wing tip, and hence responsible for augmenting the aerodynamic force prediction, and (2) a combination of the dynamics change of the wing bending and twist can modify the wing kinematics in the distal area in an appropriate way, which can lead to the aerodynamic force enhancement immediately before stroke reversal.

Kang et al. (2011) analyzed the effects of seven nondimensional parameters on the aerodynamic performance of a flapping wing, which includes the Reynolds number, the reduced frequency, the Strouhal number, the aspect ratio, the effective stiffness, the thickness ratio, and the density ratio. With an analytical model, they confirmed that the maximum propulsive force can be obtained when the wing flapping occurs near the resonance, while the optimal propulsive efficiency can be realized when the flapping frequency is about half of the natural frequency.

55.7 Conclusion

Insect-sized MAVs are expected to be capable to perform specific missions in a manner that is superior to other larger air vehicles. At present, the miniaturization of an ornithopter-type MAV such as the NANO hummingbird by AeroVironment, Inc. seems to have more advantages in maneuverability when compared with those of other MAVs. However, an optimal configuration that is most suitable for the insect-sized MAV has not been achieved yet. On one hand, MAVs with potentially high flight performance configurations, such as quad rotor or tail-sitter air vehicles, will also be candidates for insect-sized MAVs. On the other hand, some unusual MAV with a unique configuration such as the Georgia Tech Entomopter (Michelson et al. 2002) or the Embry-Riddle SamarEye (Hockley et al. 2010) may come to fascinate one with remarkable flight performance. Breakthroughs in miniaturization and the achievement of brand-new MAV configurations will only be achieved through a synthesis of various disparate disciplines such as electronics, robotics, aerodynamics, and material science.

References

- R.M. Barrett, Adaptive aerostructures, in *AIAA 2004-1886* (American Institute of Aeronautics and Astronautics (AIAA), 2004), <https://www.aiaa.org/>
- O. Bilgen, K. Kochersberger, D. Inman, O. Ohanian, J. Aircr. **47**(1), 303 (2010)
- D. Chu, J. Sprinkle, R. Randall, S. Shkarayev, *AIAA-2009-5875* (American Institute of Aeronautics and Astronautics (AIAA), 2009), <https://www.aiaa.org/>

- M. Colman, S. Suzuki, D. Kubo, *AIAA-2011-6379* (American Institute of Aeronautics and Astronautics (AIAA), 2011), <https://www.aiaa.org/>
- M. Daly, in *Jane's Unmanned Aerial Vehicles and Targets Issue 36th*, ed. by M. Daly, M. Streetly (Janes Information Group, Alexandria, 2011), p. 332
- DARPA, in *FACT FILE: A Compendium of DARPA Programs*, ed. by DARPA (United States Defense Advanced Research Projects Agency, Arlington, 2003), p. 20
- M.H. Dickinson, F.-O. Lehmann, S.P. Sane, *Science* **284**, 1954 (1999)
- C.P. Ellington, C. van den Berg, A.P. Willmott, A.L.R. Thomas, *Nature* **384**, 626 (1996)
- P. Gopalakrishnan, D.K. Tafti, *AIAA J.* **48**(5), 865 (2010)
- S. Heathcote, I. Gursul, *AIAA J.* **45**(5), 1066 (2007)
- S. Heathcote, D. Martin, I. Gursul, *AIAA J.* **42**(11), 2196 (2004)
- S. Heathcote, Z. Wang, I. Gursul, *J. Fluids Struct.* **24**, 183 (2008)
- S. Ho, H. Nassef, N. Pornsinsirak, Y.-C. Tai, C.-M. Ho, *Prog. Aerosp. Sci.* **39**, 635 (2003)
- C. Hockley, M. King, R. Khatri, C. Kirby, C. Sammet, M. Bakula, C. Reinholtz (2010), http://iarc.angel-strike.com/2010SymposiumPapers/ERAU_2010.pdf
- H. Hu, M. Tamai, *J. Aircr.* **45**(6), 2068 (2008)
- C.-K. Kang, H. Aono, C.E.S. Cesnik, W. Shyy, *J. Fluid Mech.* **689**, 32 (2011)
- M. Keennon, K. Klingebiel, H. Won, A. Andriukov, *AIAA-2012-0588* (American Institute of Aeronautics and Astronautics (AIAA), 2012), <https://www.aiaa.org/>
- D. Kubo, S. Suzuki, *J. Aircr.* **45**(1), 292 (2008)
- D. Kubo, N. Nagasaka, S. Suzuki, *AIAA-2009-1968* (American Institute of Aeronautics and Astronautics (AIAA), 2009), <https://www.aiaa.org/>
- P.J. Kunz, I. Kroo, in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, ed. by T.J. Muller (AIAA, Reston, 2001), p. 35
- H. Liu, *J. Comput. Phys.* **228**, 439 (2009)
- H. Liu, *Micro air vehicles*, in *Encyclopedia of Aerospace Engineering* ed. by R. Blockrey, W. Shyy (Wiley, Chichester, 2010)
- H. Liu, H. Aono, *Bioinsp. Biomim.* **4**, 015002 (2009)
- A.M. Mountcastle, T.L. Daniel, *Exp. Fluids* **46**, 873 (2009)
- X.G. Meng, L. Xu, M. Sun, *J. Exp. Biol.* **214**, 432 (2011)
- R.C. Michelson, M.A. Naqvi, A. Colozza, F. Porath, C. Smith, R. Banerjee, K. Isaac, P. Shivaram, T. Dalbello, C. Kory, T. Scott, NASA institute for advanced concepts project NAS5-98051 (2002), http://angel-strike.com/entomopter/NIACFinal_Report_10-31-02.pdf
- T.J. Mueller, *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*. Progress in Astronautics and Aeronautics, vol 195 (AIAA, Reston, 2001)
- T. Nakata, H. Liu, *Proc. R. Soc. B* **279**(1729), 722 (2011)
- T. Nakata, H. Liu, Y. Tanaka, N. Nishihashi, X. Wang, A. Sato, *Bioinsp. Biomim.* **6**, 045002 (2011)
- A. Obata, S. Sinohara, *AIAA J.* **47**(12), 3043 (2009)
- O.J. Ohanian III, P.A. Gelhausen, D.J. Inman, *AIAA-2010-1052* (American Institute of Aeronautics and Astronautics (AIAA), 2010), <https://www.aiaa.org/>
- O. J. Ohanian III, E. D. Karni, W. K. Londenber, P. A. Gelhausen, *J. Aircr.* **48**(2), 514(2011)
- M. Platzter, K. Jones, J. Young, J. Lai, *AIAA J.* **46**, 2136 (2008)
- C.J.C. Rees, *Nature* **145**, 200 (1975)
- S.H. Ribner, *NACA TR 820* (1943)
- F. Sato, Japanese Patent 2010-52713, 5 Mar 2010
- W. Shyy, Y. Liang, J. Tang, D. Viieru, H. Liu, *Aerodynamics of Low Reynolds Number Flyers* (Cambridge University Press, New York, 2007)
- W. Shyy, H. Aono, S.K. Chimakurthi, P. Trizila, C.-K. Kang, C.E.S. Cesnik, H. Liu, *Prog. Aerosp. Sci.* **46**, 284 (2010)
- D.O. Sigthorsson, M.W. Oppenheimer, D.B. Doman, *AIAA 2012-0028* (American Institute of Aeronautics and Astronautics (AIAA), 2012), <https://www.aiaa.org/>
- B. Stanford, P. Ifju, R. Albertani, W. Shyy, *Prog. Aerosp. Sci.* **44**, 258 (2008)

- S. Sunada, L. Zeng, K. Kawachi, *J. Theor. Biol.* **193**, 39 (1998)
- H. Tokutake, S. Sunada, J. Fujinaga, *J. Syst. Des. Dyn.* **5**(1), 1 (2011)
- T. Weis-Fogh, *J. Exp. Biol.* **59**, 169 (1973)
- R.J. Wootton, *J. Zool.* **193**, 447 (1981)
- J. Young, S.M. Walker, R.J. Bomphrey, G.K. Taylor, A.L.R. Thomas, *Science* **325**, 1549 (2009)
- H.W. Zhao, C. Bil, *AIAA-2009-7097* (American Institute of Aeronautics and Astronautics (AIAA), 2009), <https://www.aiaa.org/>
- Q. Zhu, *AIAA J.* **45**(10), 2448 (2007)

Flapping-Wing Propelled Micro Air Vehicles 56

Kevin D. Jones and Max F. Platzer

Contents

| | | |
|--------|---|------|
| 56.1 | Introduction | 1360 |
| 56.2 | Lift Generation on Wings Flying at a Steady Angle of Attack | 1361 |
| 56.3 | Thrust and Lift Generation on Flapping Wings | 1367 |
| 56.3.1 | Flapping Single Wings | 1367 |
| 56.3.2 | Flapping Tandem Wings | 1370 |
| 56.3.3 | Flapping Biplane Wings | 1371 |
| 56.3.4 | Two Wings Flapping in the Clap-and-Fling Mode | 1371 |
| 56.3.5 | Hovering Single Wings | 1372 |
| 56.4 | Flapping-Wing Micro Air Vehicles | 1373 |
| 56.4.1 | Microbat (AeroVironment Inc.) | 1373 |
| 56.4.2 | “NPS Flyer” (Naval Postgraduate School) | 1374 |
| 56.4.3 | Delft University DelFly | 1377 |
| 56.4.4 | Wright State University Micro Air Vehicle | 1378 |
| 56.4.5 | “Nano Hummingbird” (AeroVironment Inc.) | 1379 |
| 56.5 | Concluding Remarks | 1381 |
| | References | 1382 |

Abstract

In this chapter, a brief history of the major discoveries in the scientific exploration of flapping-wing flight is presented. This is followed by a short review of the basic concepts of lift generation on wings in low-speed, steady flight, which leads into a discussion of the generation of thrust due to the flapping of wings. The aerodynamics of single flapping wings in forward and hovering flight, of flapping tandem and biplane wings, and of dual wings using the clap-and-fling effects are discussed. The chapter concludes with an overview of the major characteristics of five representative flapping-wing propelled micro air

K.D. Jones (✉) • M.F. Platzer

Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, USA

e-mail: jones@nps.edu; mplatzer@nps.edu

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

1359

DOI 10.1007/978-90-481-9707-1_11,

© Springer Science+Business Media Dordrecht 2015

vehicles developed to date, including models developed at the AeroVironment Company, the Naval Postgraduate School, Wright State University, and Delft University.

56.1 Introduction

Anyone observing the seemingly effortless flight of birds and insects cannot help but be impressed by their abilities. Indeed, bird flight greatly inspired early aviation pioneers, such as Lilienthal (1992). The bird or insect wing represents what is referred to in modern aeronautical engineering language as a fully integrated lift/propulsion/control system. From today's vantage point, it is understandable that the attempts by Lilienthal and others to imitate bird flight remained unsuccessful for quite some time. There were simply too many phenomena which had to be understood, foremost the aerodynamics of flapping wings. The Wright brothers correctly recognized that the lifting, propulsion, and control elements had to be separated in order to build a successful flying machine. They also recognized that it was sufficient to imitate the shape of a bird wing in order to generate lift but to refrain from flapping it. It took several more years until it was understood that a thick wing with a well-rounded leading edge was to be preferred over a typical birdlike, thin wing.

For many decades, aeronautical scientists had little interest in the study of flapping-wing flight, leaving this field mostly to ornithologists. Nevertheless, Knoller in Vienna and Betz in Göttingen provided a first elementary explanation for thrust generation by wing flapping in independent studies not too long after the Wright brothers' first flight (Knoller 1909; Betz 1912). Inspired by the work of Knoller and using Knoller's wind tunnel, Katzmayer (1922) found that an airfoil mounted in an oscillating airstream experienced a thrust force (Katzmayer 1922). This was the first experimental verification of the Knoller-Betz effect and, although he most likely had not realized it, probably the first experimental study of dynamic soaring, a common flight technique found in the animal kingdom. At about the same time, a more comprehensive explanation of the Knoller-Betz effect was obtained by Birnbaum in his Göttingen dissertation under the guidance of Prandtl (Birnbaum 1924). He applied a concept first proposed by Prandtl in 1922 at a scientific congress in Innsbruck, Austria (Prandtl 1924). Prandtl showed that it was possible to formulate a linearized theory for incompressible, two-dimensional flow past oscillating thin airfoils using the method of singularities. He explained that a vortex must be shed from an airfoil's trailing edge whenever the airfoil's lift changes, in order to satisfy Kelvin's theorem of constancy of the total circulation. Birnbaum applied Prandtl's theory and succeeded in obtaining results for the oscillatory lift in response to a harmonic small amplitude airfoil oscillation. He also showed that a chordwise force is being generated. This force may be a thrust rather than a drag under certain conditions, thus confirming the Knoller-Betz-Katzmayer effect. Another fundamental insight into lift generation was obtained by Wagner in a dissertation submitted at the Technical University in Berlin (Wagner 1925).

It treated the vortex generation and the lift buildup in response to an airfoil's sudden acceleration to a constant speed or to a step change in angle of attack.

This brief history shows that within about 20 years after the first powered flight by the Wright brothers in 1903 and the recognition of the fundamental relationship between lift and circulation found by Kutta (1902) and Joukowski (1907) at about the same time (the Kutta-Joukowski theorem), the mechanism and prediction of lift generation on a finite-span wing in steady flight had been clarified by Prandtl during World War I with his lifting-line theory (Prandtl 1927), and the clarification of the mechanism and prediction of unsteady lift and thrust generation on thin airfoils had been achieved by Knoller, Betz, Katzmayr, Prandtl, Birnbaum, and Wagner. In the following years, the Prandtl-Birnbaum approach became of great importance for the analysis of wing flutter, pioneered by Kuessner in Göttingen (Kuessner 1936) and Theodorsen at NACA-Langley (Theodorsen 1935). Theodorsen's associate Garrick applied the analysis to the prediction of thrust on airfoils executing harmonic pitch or plunge oscillation (Garrick 1936). In the following years, the emphasis remained on flutter and gust response analysis and on the extension of Prandtl's theory to finite-span wings of general plan form in both steady and unsteady flow. As already stated, the use of flapping wings for flight propulsion was considered impractical by most aeronautical engineers for the remainder of the twentieth century. This perception changed only at the end of the century with the recognition that flapping-wing propulsion might have advantages when used on small air vehicles.

In this chapter, an overview of the basic physics of lift generation on wings flying at a constant speed and angle of attack is presented, followed by a discussion of the basics of flapping-wing aerodynamics. Then the major design features and flight characteristics of five of the many successful flapping-wing propelled air vehicles which have been developed during the past 15 years are discussed.

56.2 Lift Generation on Wings Flying at a Steady Angle of Attack

Because of the small flight speeds of flapping-wing propelled air vehicles, the discussion is limited to incompressible flow considerations. Consider the flow over a typical airfoil, shown in Fig. 56.1. The airfoil has a well-rounded leading edge and a sharp trailing edge. The airfoil chord line is inclined relative to the oncoming freestream. The typical explanation for the generation of lift on this airfoil makes use of Bernoulli's equation for incompressible flow:

$$p + \frac{1}{2}\rho u^2 = \text{const} \quad (56.1)$$

where p is the local pressure, ρ is the density, and u is the local flow speed.

The forward stagnation point on this airfoil is seen to be located at a point close to the foremost point near the nose but somewhat below this point on the airfoil lower surface. Consider two neighboring fluid particles, one flowing slightly above

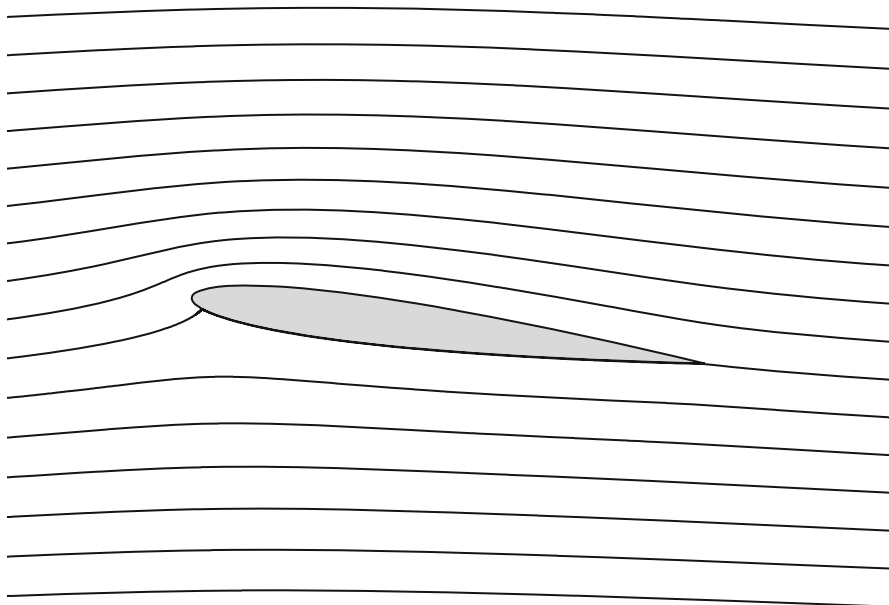


Fig. 56.1 Flow over a stationary airfoil at an angle of attack. Flow is from *left to right*, and the streamline that appears to run into the airfoil is referred to as the *stagnation streamline*

the stagnation streamline and the other one slightly below this line. If they are to arrive at the trailing edge at the same time, it is “readily” seen that the upper fluid particle has to travel a larger distance than the lower particle. Hence it has to flow at a higher speed which, in turn, means according to Bernoulli that the pressure on the upper surface is lower than on the lower surface. This pressure difference between lower and upper surface integrates to a finite lift force.

Although this account is widely used as an explanation of lift generation, the term “readily” is put between quotation marks because it disregards important physical features. The key to a fuller understanding was found by Kutta (1902) and Joukowski (1907). They derived the important equation, now known as the Kutta-Joukowski law:

$$L = \rho U \Gamma \quad (56.2)$$

which relates the airfoil lift, L , to a quantity called *circulation*, given by Γ . Consider the photo shown in Fig. 56.2.

This photo, taken by Prandtl, shows the changes in flow pattern near the airfoil leading and trailing edge as the airfoil is suddenly started from rest while holding it at a small but finite angle of attack. A vortex is observed to form at the trailing edge which, after a while, starts to detach. Kutta and Joukowski recognized the importance of the relationship between the strength of this vortex (its circulation) and the airfoil lift. They recognized that the airfoil can be regarded as a vortex (more precisely as a series of vortices of varying strength extending from the

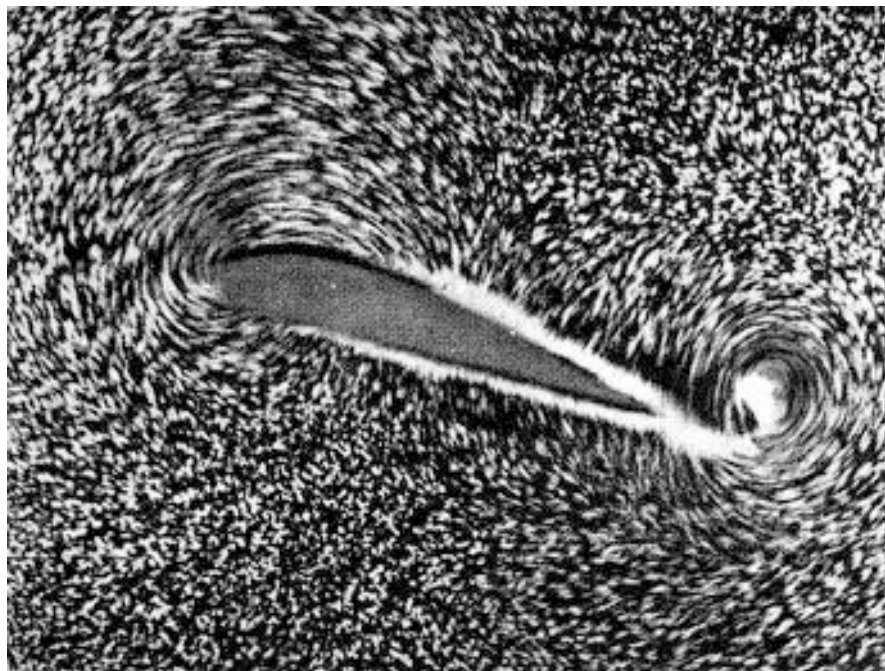


Fig. 56.2 Vortex generation on an airfoil shortly after an impulsive start. Visible is the counter-clockwise trailing edge vortex

leading to the trailing edge) which induces a circulatory flow. Using ideal inviscid flow assumptions, they could solve the governing equation (the Laplace equation) subject to the proper boundary conditions on the airfoil surface. One condition is the obvious requirement that the airfoil surface is a streamline (i.e., flow cannot pass through the airfoil wall). However, there is a second condition which is crucial for the correct lift prediction. It is the Kutta trailing edge condition requiring that the rearward stagnation point be located at the sharp trailing edge. The use of this condition yields very good agreement with the measurements, as is shown in Fig. 56.3.

The predicted lift is seen to be in excellent agreement with the measurements on a NACA 0012 airfoil up to angles of attack of about 16° but starts to deviate sharply as soon as the flow stalls due to flow separation. Visitors of the National Air and Space Museum will notice that on their airplane the Wright brothers used a much thinner airfoil than the 12 % thick NACA 0012 airfoil. Indeed, all the airplanes used until 1917 had thin airfoils because it was believed that a good airfoil was one that imitated bird wings as closely as possible. However, Prandtl showed that a thicker airfoil with a well-rounded leading edge delayed the onset of stall to a higher angle of attack and therefore had a higher maximum lift coefficient. The famous airplane designer Anthony Fokker adopted this type of airfoil on the Fokker D.VII airplane,

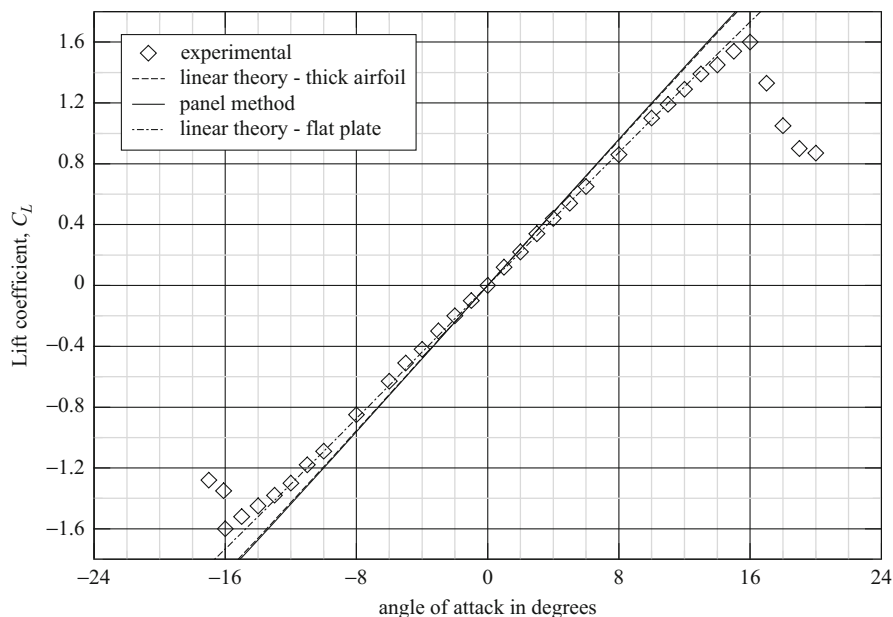


Fig. 56.3 Lift coefficient as a function of angle of attack for a NACA 0012 airfoil. Experimental data due to (Abbott and Von Doenhoff 1959)

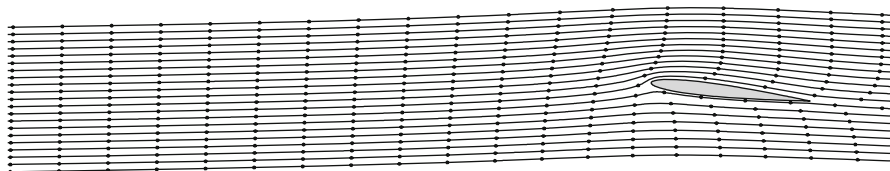
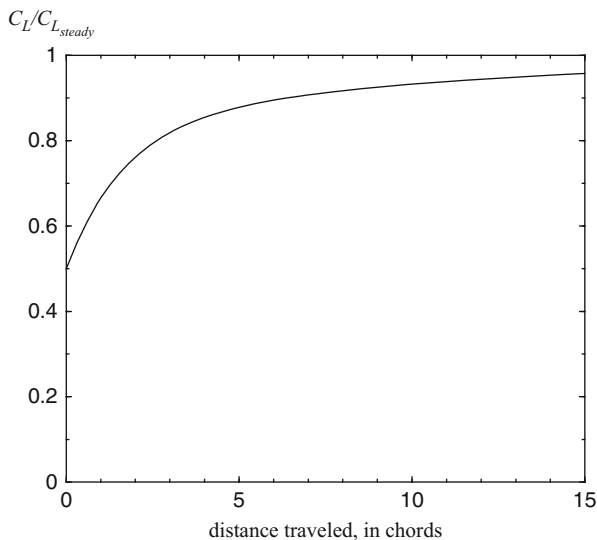


Fig. 56.4 Flow over an airfoil at a small angle of attack. Columns of *dots* on the streamlines are released simultaneously far upstream and travel downstream as a wave front. The bound circulation on the airfoil speeds up the particles traveling *above* the wing and slows down the particles traveling *below* the airfoil such that the particles taking the *upper route* actually arrive at the trailing edge before the particles taking the *lower route*

making it possible for the German pilots to outclimb and outmaneuver their French and British opponents. The thicker wings allowed for thicker spars providing much stronger wings, allowing designers to do away with flying wires, and providing a secondary benefit of significant drag reduction.

The vortex formation and detachment from the airfoil trailing edge caused by the sudden start from rest shown in Fig. 56.2 is the key to a full understanding of lift generation. It is this starting vortex that induces a circulatory flow around the airfoil. The two neighboring fluid particles mentioned earlier therefore arrive at different times at the trailing edge, as shown in Fig. 56.4.

Fig. 56.5 Lift as a function of the distance the starting vortex has traveled downstream, due to Wagner



The particle flowing over the upper surface flows at a faster speed, and therefore, applying the Bernoulli equation, the pressure on the upper surface is lower than on the lower surface. It is important to remember the crucial role of a sharp trailing edge. A rounded trailing edge weakens the starting vortex formation. Furthermore, it is important to keep in mind that the formation and detachment of the starting vortex takes a finite amount of time. For this reason, the lift cannot instantaneously jump to its steady-state value in response to an impulsive airfoil start or a sudden change in angle of attack. Instead, as first shown by Wagner (1925) in Fig. 56.5, the airfoil needs to travel some 20 chord lengths before the steady-state lift value is reached.

Having recognized that any change in flow direction (due to flight in gusty air or in an oscillating air stream, as generated by Katzmayr) causes the shedding of starting vortices, it is clear that a flapping wing also causes the shedding of starting vortices due to the continuously changing angle of attack. The K  nner-Betz explanation of thrust generation by wing flapping therefore is incomplete. The complete physics of thrust generation involves the shedding of starting vortices.

The full understanding of lift and thrust generation by a *finite-span wing*, however, requires knowledge of the three-dimensional flow phenomena generated by a finite-span wing. During World War I, Prandtl developed a simplified model for the steady flow past a lifting wing which describes the actual flow amazingly well. It is the famous lifting-line model which became the basis for future more accurate models. Prandtl suggested that the lifting flow over a typical rectangular high-aspect ratio wing can be modeled by replacing the wing by a single vortex line (the bound vortex). Since a vortex line cannot simply end at the wing tips, he postulated that the bound vortex line must be continued at the wing tips by two trailing vortices aligned with the flow direction. In the case of an impulsively starting wing, a closed vortex

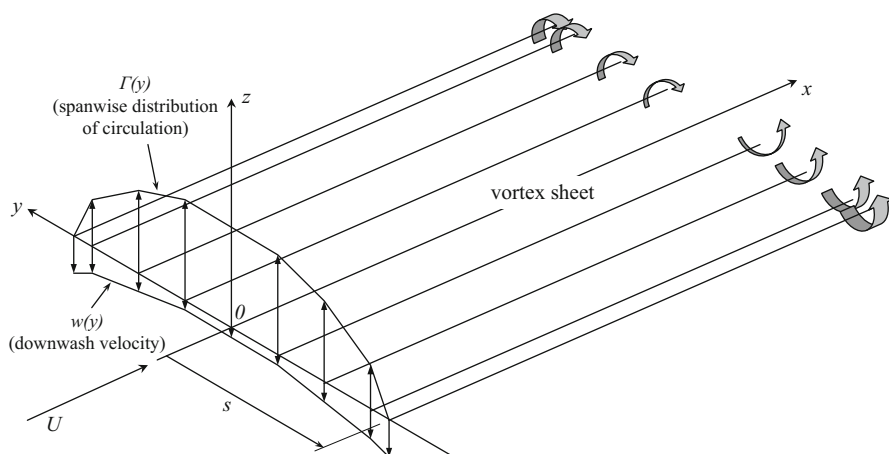


Fig. 56.6 Prandtl's lifting-line theory for finite wings

loop is then formed by the bound, trailing, and starting vortices. As in the two-dimensional case, the influence of the starting vortex diminishes with increasing distance between wing and starting vortex. A lifting wing in steady flight therefore can be modeled by the horseshoe model of one bound vortex and two trailing vortices that extend downstream to infinity. However, this model is still incomplete because it implies that the circulation (and therefore the lift) of a single bound vortex is constant along the wing span, requiring the lift to drop to zero abruptly at the wing tips. This being physically impossible, Prandtl refined the single horseshoe model by postulating the shedding of a trailing vortex with every change in lift in the spanwise direction. This lifting-line model is illustrated in Fig. 56.6. The physics of lift generation now becomes clear. The trailing vortex sheet (especially the strong vortices shed near the wing tips) captures a certain amount of oncoming flow and gives it a downward momentum. The reaction to this downward momentum flow is the wing lift. As in the two-dimensional case, lift generation requires the generation of vortices.

Prandtl succeeded in analyzing this flow model mathematically. He showed that the induced downward velocity is related to the wing lift by the formula

$$\frac{w}{U} = \frac{C_L}{\pi AR} \quad (56.3)$$

which can also be written as

$$L = 2\pi s^2 \rho U w \quad (56.4)$$

where w is the induced downward velocity, AR is the finite wing aspect ratio, s is the semispan of the wing, and C_L is the lift coefficient, given by $C_L = 2L/(\rho U^2 S)$, where S is the wing area.

This formula specifies the amount of freestream flow that is captured by the trailing vortex sheet and given a downward velocity. It is the air mass flowing through a circle with a radius equal to the semispan. The downward velocity is twice the downward velocity induced at the wing, another important result of this theory. Most importantly, it is seen that lift generation is brought about by the rate of change of momentum in agreement with Newton's second law. Lift can only be maintained if the wing continues to generate trailing vortices, thus requiring power. Another important result derived by Prandtl was the recognition that the required power becomes a minimum for constant spanwise downwash distribution and elliptic spanwise lift distribution. Equations (56.3) and (56.4) also show that they contain the two-dimensional flow case as the asymptotic limit case. The induced downward velocity becomes zero for infinite aspect ratio, while the captured mass flow becomes infinite so that the product of mass flow and downward velocity yields a finite lift.

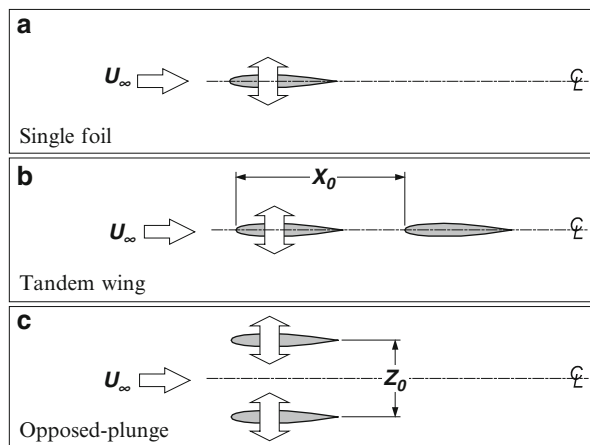
56.3 Thrust and Lift Generation on Flapping Wings

In this section, the aerodynamics of single flapping wings and several flapping-wing combinations will be discussed. These configurations are sketched in Fig. 56.7. In each case, the flapping motion can include just a plunge motion or a combination of pitch and plunge with some phase offset, and if the freestream flow drops to zero, the wing is assumed to be hovering.

56.3.1 Flapping Single Wings

For a first understanding of the aerodynamics of flapping wings, it is again advantageous to consider first the two-dimensional flow physics. The visualization

Fig. 56.7 Common flapping-wing configurations. The single flapping wing models the flight of birds and many modern insects. The tandem wing is found in dragonflies and less developed insects. The opposed-plunge model is not found in nature except when a bird is flying near the ground, where the symmetry plane between the two wings is replaced by the ground



of the vortical wake shed from the trailing edge of an airfoil (visible on the left-hand side) is shown in Fig. 56.8 for increasing flapping frequency. In Fig. 56.8a, the airfoil is held steady. Two vortex rows are seen to be shed from the airfoil trailing edge such that the upper row consists of clockwise-turning vortices and the lower row of counterclockwise vortices.

This type of vortex street is known as a Kármán vortex street, and it typically forms in the flow over cylinders and airfoils at low Reynolds numbers. Interesting changes in vortex street behavior occur when the airfoil is being oscillated in a pure plunge (up and down) motion. As the frequency and amplitude of oscillation are increased, the mushroomlike vortices of Fig. 56.8b are generated which switch to the vortical wake shown in Fig. 56.8c and finally to the vortex street of Fig. 56.8d. This vortex street is usually denoted as the reverse Kármán vortex street because the upper row of vortices now turns counterclockwise and the lower row turns clockwise.

It is apparent from the fact that the upper vortices turn counterclockwise and the lower vortices turn clockwise that a relatively high velocity is being induced between the two rows. Thus, the time-averaged flow at a given station downstream from the airfoil trailing edge resembles that of a jet flow, as shown in Fig. 56.9. This jet flow could be predicted with an inviscid flow analysis (panel code). Although it is not generally realized, birds employ the same jet propulsion principle used for the propulsion of airplanes and missiles. Wing flapping provides the energy needed to obtain a momentum increase. A flapping wing therefore can be regarded as a “rectangular” propeller which draws air into the propeller and gives it a momentum increase, which must result in a thrust force according to Newton’s second law.

In Figs. 56.8 and 56.9, the airfoil was oscillated in a pure plunge mode. A similar but weaker propulsion effect is obtained by pitching the airfoil about, say, the quarter chord point. It turns out that a thrust is generated only above a certain threshold reduced frequency (reduced frequency is the product of chord length times flapping frequency divided by the freestream speed). However, a combined pitch-plunge oscillation is advantageous because the propulsive efficiency is a maximum when the pitch leads the plunge oscillation by roughly 90° . It also turns out that at the small Reynolds numbers, typical for micro air vehicles, viscous effects can be quite strong, especially at small reduced frequencies. For a more detailed discussion and computational analysis of these effects, refer to Platzer et al. (2008), Young and Lai (2004, 2007), and Shyy et al. (2008).

As in the steady lifting case, discussed above, the oscillating finite-span wing sheds additional vortices, especially near the wing tips. In the case of a single-step change in angle of attack, a single vortex ring is generated consisting of the bound vortex on the wing, two trailing vortices that reach from the wing tips back to the starting vortex that was generated at the moment of the step change. As seen in Fig. 56.10, the oscillating finite-span wing generates alternating vortex rings. The flow shown in Fig. 56.10 was obtained using an inviscid three-dimensional panel code. Viscosity effects change this picture considerably, especially at increased distance from the trailing edge. The flow in the mid-span vertical plane is seen to resemble the two-dimensional reverse Kármán vortex shedding shown in Fig. 56.8d.

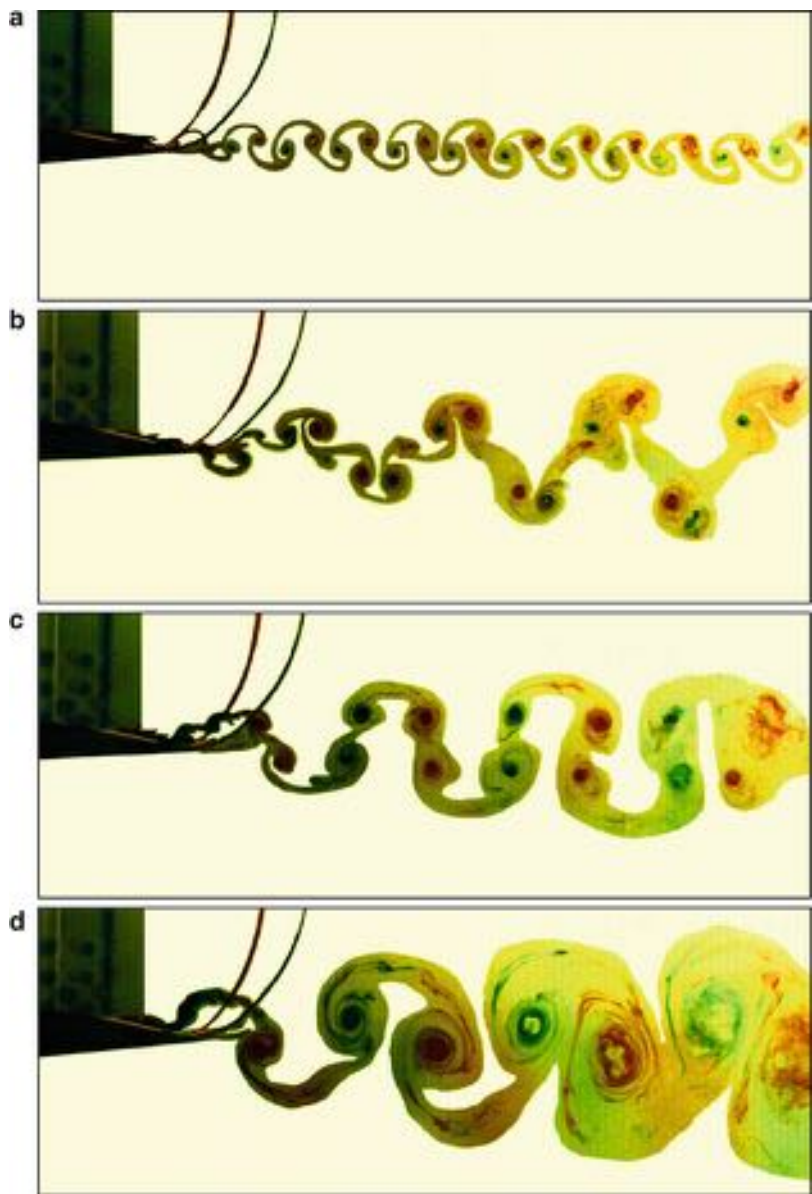


Fig. 56.8 Transition from normal to reverse Kármán vortex street with increasing frequency and amplitude of airfoil oscillation. (a) Stationary airfoil shedding a Kármán vortex street, indicative of drag. (b) Flapping with a low frequency and amplitude such that net thrust is still negative. (c) Flapping with a frequency and amplitude where the net thrust is essentially zero. (d) Flapping with a frequency and amplitude where thrust exceeds drag, producing a reverse Kármán vortex street

Fig. 56.9 Time-averaged streamwise velocity profile in the wake of a flapping wing measured experimentally and predicted numerically

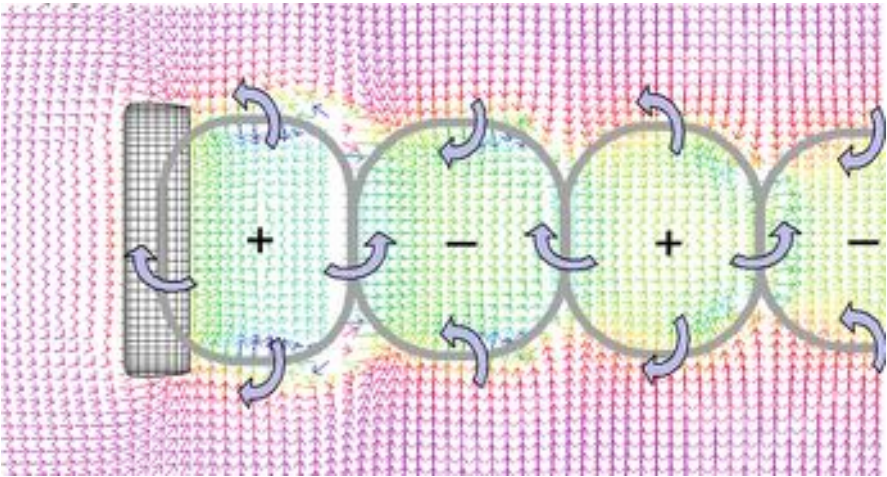
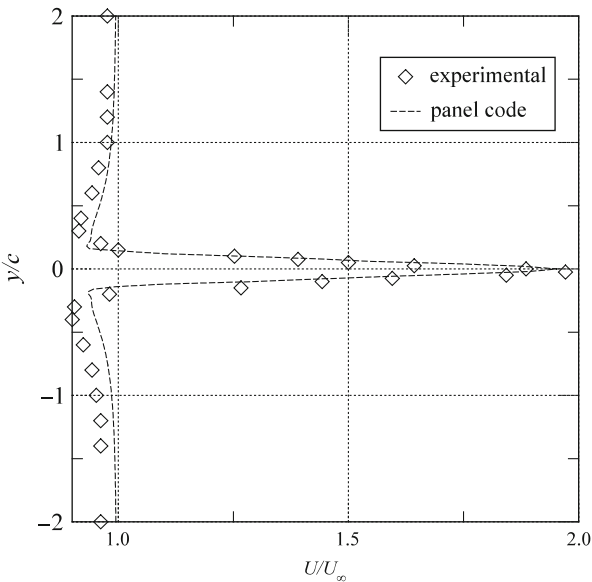


Fig. 56.10 Alternating vortex rings in the wake of a finite-span flapping wing

56.3.2 Flapping Tandem Wings

Consider the two-foil tandem arrangement shown in Fig. 56.7b. If the second airfoil is stationary, an oscillation of the leading airfoil will expose the non-oscillating trailing airfoil to an oscillatory flow. This is the flow situation first studied by Katzmayr (1922), where he measured a thrust on an airfoil mounted in an oscillatory air stream. Schmidt appears to have been the first one during and after World War II

to use this arrangement for the development of his *wave propeller* (Schmidt 1965). It is now well recognized that birds can minimize their wing flapping and, therefore, energy expenditure by seeking to fly in wavy air, a form of dynamic soaring. Similarly, fish can remain stationary with minimum tail flapping in wavy water flow, such as the Kármán vortex street behind a rock in a river. A more complicated interaction effect occurs when both foils are oscillated. Depending on the phase angle and amplitude of oscillation of the two airfoils, rapid thrust and lift changes can be achieved, giving insects such as the dragonfly astounding maneuverability. For more details, refer to Isogai et al. (2004) and Yamamoto and Isogai (2005).

56.3.3 Flapping Biplane Wings

It is well known that a wing flying very close to the ground experiences a substantial increase in lift. This favorable ground effect occurs also on flapping wings. Both the thrust and the propulsive efficiency are significantly increased on wings flapping close to the ground, as shown in Fig. 56.11. Birds make use of this effect when flying close to a water surface, for example, as shown in Fig. 56.12. The same effect can be achieved in flight out-of-ground effect when two wings are used that flap in counterphase, as shown in Fig. 56.7c. According to Wootton and Kukalova-Peck (2005), large insects existed a long time ago that had forewings and hind wings which overlapped extensively, thus making use of this favorable biplane effect. However, no present-day insects are known to use it. Hence micro air vehicles, such as the one developed by the present authors (Jones et al. 2005), must be classified as biomorphic vehicles, where the characteristics of natural flyers are modified considerably, in contrast to biomimetic vehicles which try to simulate natural flyers as closely as possible. Further details about computational and experimental data on flapping biplane wings can be found in Platzer et al. (2008) and Jones et al. (2000, 2005).

56.3.4 Two Wings Flapping in the Clap-and-Fling Mode

There exists another thrust and lift generation mechanism used by natural flyers. Consider the typical hovering insect configuration depicted in Fig. 56.13. As the two wings close up as shown in Fig. 56.13b, c, e, the wings force air out of the wing trailing edges, thus creating a jet flow and hence lift. Then, as the wings begin to open up at the top, as shown in Fig. 56.13f, h, i, a low pressure is induced between the wings, causing the formation of strong leading edge vortices, thus again generating a substantial amount of lift. Parts a, d, and g correspond to parts f, h, and i, illustrating what the insect is doing in three dimensions. This effect was first observed and described by Weis-Fogh in 1973 and is often referred to as the Weis-Fogh effect (Weis-Fogh 1973). Lehmann et al. report a recent experimental investigation of this effect (Lehmann et al. 2005). Figure 56.13 is a composite image using sketches by

Fig. 56.11 Predicted thrust and efficiency for flapping wings in ground effect

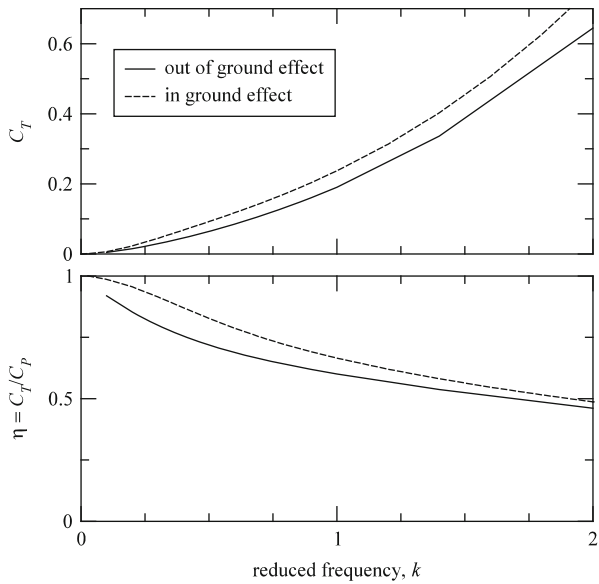


Fig. 56.12 A brown pelican and a great egret demonstrating the use of ground effect to minimize operator workload

Weis-Fogh (1973) and Sane (2003), available online at the Journal of Experimental Biology (Sane 2011).

56.3.5 Hovering Single Wings

Hummingbirds and many insects are able to hover by moving their wings in an approximately horizontal plane. Although the actual wing motion more closely follows a figure eight trajectory, the basic effect can be understood by studying the vortex shedding caused by the motion in a horizontal plane. As shown in Fig. 56.14,

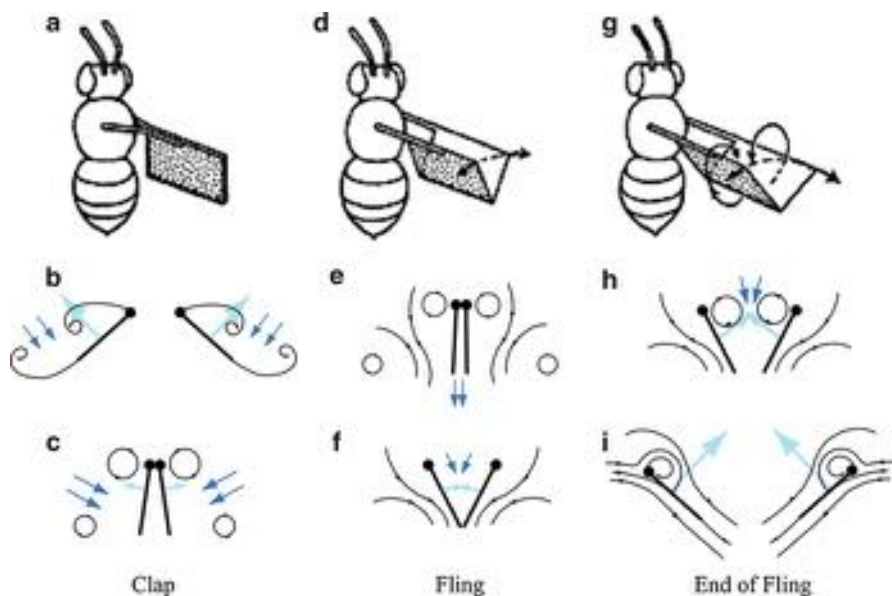


Fig. 56.13 Clap-and-fling or Weis-Fogh effect to enhance lift/thrust (Reproduced/adapted with permission from the Journal of Experimental Biology Weis-Fogh 1973; Sane 2003, 2011)

the airfoil moves to the left at constant positive angle of attack, then rotates at the end of the stroke so that the leading edge points in the opposite direction, proceeds to the right with positive angle of attack and then rotates back at the right end of the stroke. Strong vortices are shed during the airfoil rotation, thus generating substantial amounts of lift. Other hovering modes are possible, as discussed for example in Shyy et al. (2008).

56.4 Flapping-Wing Micro Air Vehicles

56.4.1 Microbat (AeroVironment Inc.)

While model airplane enthusiasts had been building rubber band-powered ornithopters for many decades, the AeroVironment Microbat, shown in Fig. 56.15, was the first known remotely controlled MAV ornithopter design. The design philosophy applied by the AeroVironment Company was to take a functional rubber band-powered vehicle and substitute an electric motor drivetrain for the rubber band and add a radio for control. The Microbat model uses a 2-cell lithium polymer (Li-poly) battery and a three-channel radio. It has a span of 23 cm and a total weight of 14 g. Its endurance is approximately 25 min. In trying to mimic bird flight, the Microbat is an example of a biomimetic design.

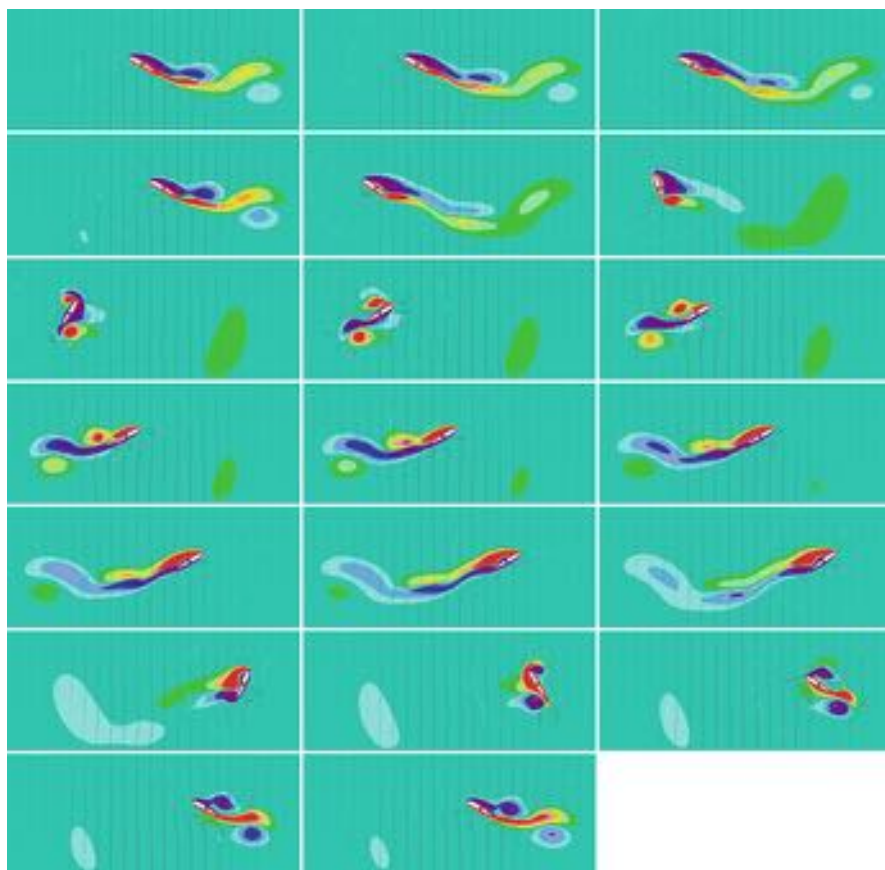


Fig. 56.14 Numerical prediction of a single wing flapping in a figure eight hovering mode (Kurtulus et al. 2005)

56.4.2 “NPS Flyer” (Naval Postgraduate School)

In contrast, the present authors decided to use flapping wings primarily for propulsion while generating most of the lift by a stationary (non-flapping) wing, hence to pursue the concept of a biomorphic design. Having chosen to separate thrust and lift generation, it remained to choose the type of flapping-wing propeller and its location. These decisions were greatly facilitated by knowledge about the superiority of flapping biplane wings and by the possibility of flow separation control through a closely coupled wing-propeller arrangement. Since it is well known that propellers mounted close to the wing trailing edge entrain the flow and therefore delay wing stall, the flapping-wing propeller was positioned downstream of the fixed wing. Another key decision was the choice of flapping mechanism which would make it possible to build a sufficiently lightweight biplane flapper.

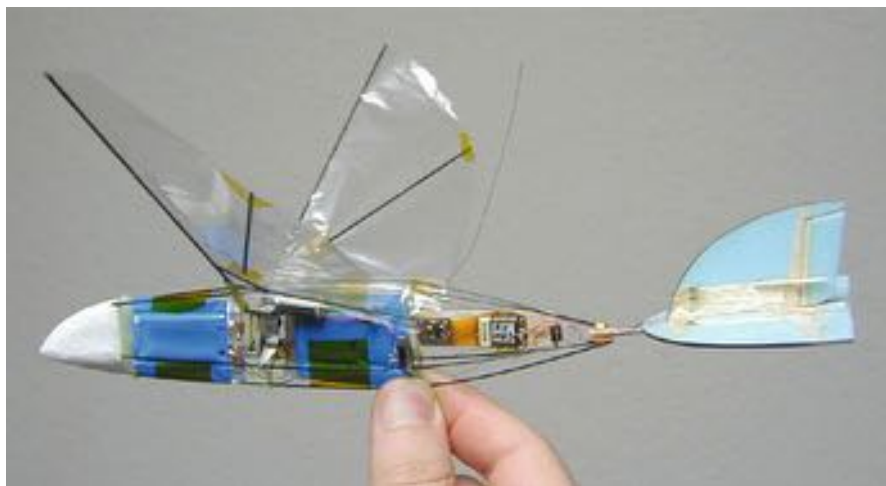


Fig. 56.15 Microbat flapping-wing vehicle (AeroVironment Inc.) (Keennon and Grasmeyer 2003)



Fig. 56.16 MAV-sized biplane propulsion test model

The configuration shown in Fig. 56.16 consists of two swing arms which flap about a point several chord lengths ahead of the biplane wings, driven by a crankshaft that is connected to a tiny geared electric motor. The wings are elastically coupled to the swing arms such that aerodynamic loads create a passive coupled pitching motion.

As already mentioned, conventional propellers located close to but downstream of a wing's trailing edge have a favorable effect on the flow over the wing's upper surface. This consideration is particularly important for low Reynolds number air vehicles due to their increased tendency toward flow separation. High-aspect ratio flapping-wing propellers have the advantage over conventional propellers that the flow along the whole wing span can be entrained. The effectiveness of this type

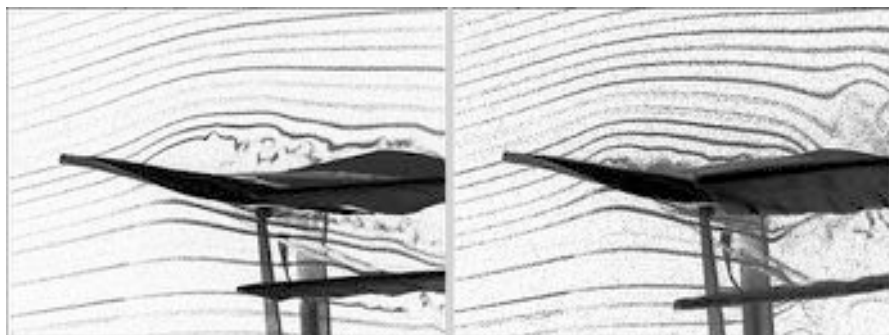


Fig. 56.17 Visualization of separation control on the fixed/flapping-wing test model. In the *left photo*, the trailing flapping wings are at rest, and a massive, turbulent separated region above the main wing is visible. In the *right photo*, the biplane flapping wings are active, and the separated region above the main wing is gone. The reattachment process is very fast, requiring about four flapping strokes to complete

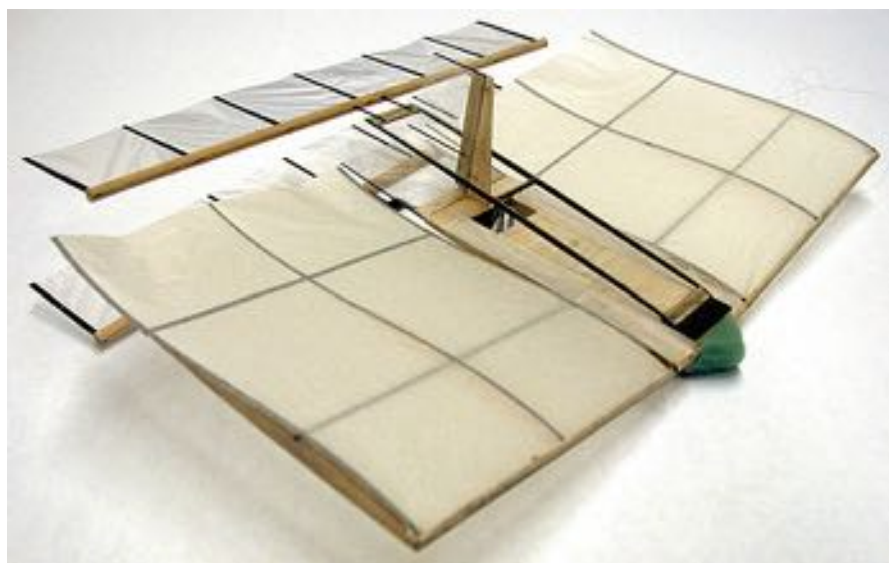


Fig. 56.18 The first flyable flapping-wing MAV at NPS

of flow control can be clearly seen in Fig. 56.17. The upper image shows the flow over the wing for the case where the two small biplane wings are stationary. Flow separation over the wing upper surface is clearly visible. The lower figure shows the case where the biplane wings are flapping. The flow separation over the upper wing surface is seen to be greatly reduced due to the entrainment effect of the flapping-wing propeller.

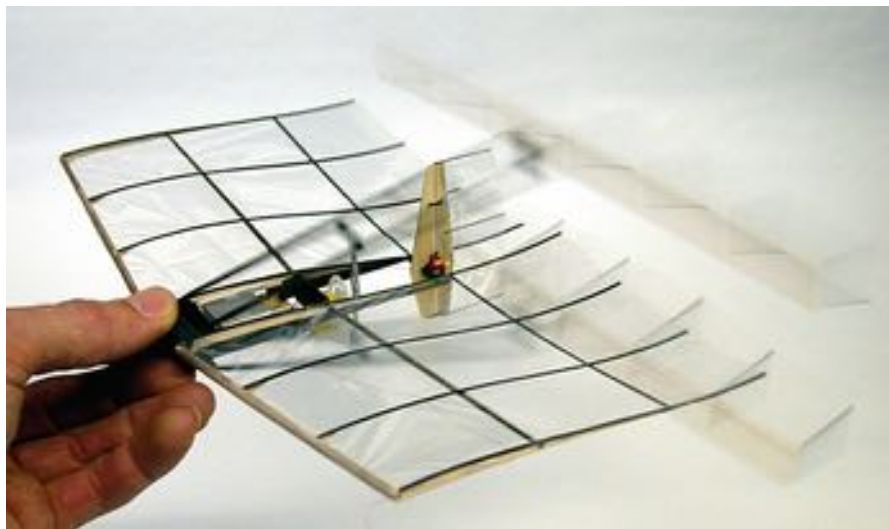


Fig. 56.19 An improved NPS flapping-wing model

The first complete, flyable micro air vehicle is shown in Fig. 56.18. It first flew in December 2002. It had a main wing with a 30 cm span and 14.5 cm chord and flapping wings with a 25 cm span and 4 cm chord. The weight was 14.4 g. Subsequent development made it possible to reduce the span to 23 cm and the weight to 11 g, as shown in Fig. 56.19. Minor trim changes allowed for flight speeds between 2 and 5 m/s. The endurance on the single rechargeable lithium polymer battery was about 12 min. Further details can be found in Jones et al. (2005).

56.4.3 Delft University DelFly

As previously mentioned, the rubber-powered model airplane community had been experimenting with flapping-wing aircraft for the better part of a century. Most successful aircraft looked pretty much like birds, but in 1985 Frank Kieser came up with the design shown in Fig. 56.20. It used four wings that flapped in counterphase, providing the same mechanical and aerodynamic symmetry as the NPS flyer, but with wings that were hinged at the root, more like birds and insects. His design was very successful and held endurance records for many years.

The DelFly, developed at the Technical University of Delft, is a variation on the Frank Kieser's design, moving the flapping wings to the front and adding the empennage at the rear (Lentink et al. 2009; de Croon et al. 2009). The DelFly I, shown in Fig. 56.21, had a 50-cm span and 21 g mass and was only capable of cruise flight. The DelFly II, shown in Fig. 56.22, had a 28-cm span and 16 g mass, carried a wireless video camera, and was able to hover and take off and land vertically using

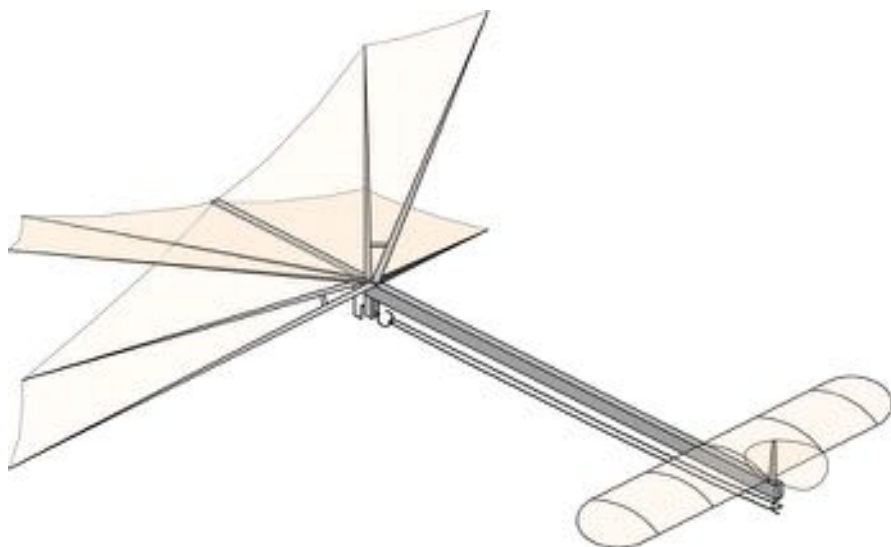


Fig. 56.20 Frank Kieser's record-breaking rubber-powered x-wing flapper



Fig. 56.21 DelFly I – 50-cm span, 21 g mass

its tailfins as landing gear. On the DelFly II, the upper set of wings was designed to come together at the top to make use of the Weis-Fogh effect.

The latest addition to the fleet was the DelFly Micro, shown in Fig. 56.23, with a 10-cm span and 3.07 g total mass including video. It is the smallest known flapping-wing MAV.

56.4.4 Wright State University Micro Air Vehicle

At the Wright State University, a flapping-wing micro air vehicle was developed that makes use of the Weis-Fogh clap-and-fling effect (Hsu et al. 2010;

Fig. 56.22 DelFly II in hovering flight – 28-cm span, 16 g mass, with video camera



Dawson et al. 2011). The design is based on a three-way clap-and-fling mechanism. Two wings on either side are flapping in such a way that they come together on both ends of the stroke as the wings are in the horizontal and vertical positions, as shown in Fig. 56.24. The wings are covered with a thin polyester film with a thickness of $25\ \mu$. The maximum flapping frequency is 20 Hz, and the wing span is 20 cm. The total weight including a 0.67 g camera is 12.56 g. The vehicle was demonstrated to be capable of hovering and of performing simple aerobatic maneuvers.

56.4.5 “Nano Hummingbird” (AeroVironment Inc.)

The design requirement, specified by the Defense Advanced Research Projects Agency (DARPA), was to develop a biomimetic vehicle that would simulate a hummingbird as closely as possible. Therefore, it had to be capable of hovering for a minimum of 8 min as well as to transition into forward flight. As shown in Fig. 56.25, both body and wings closely resemble a hummingbird. To achieve

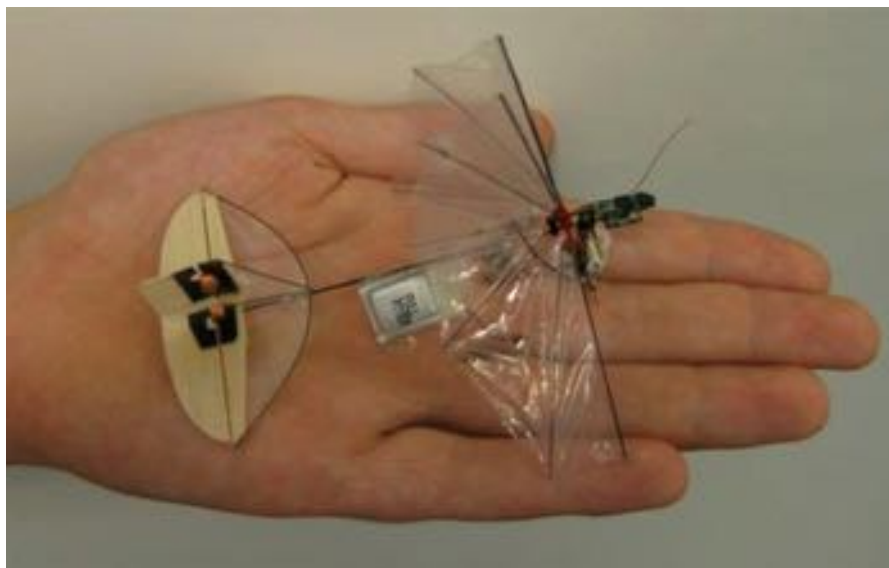


Fig. 56.23 DelFly Micro – 10 cm span 3.07 g mass

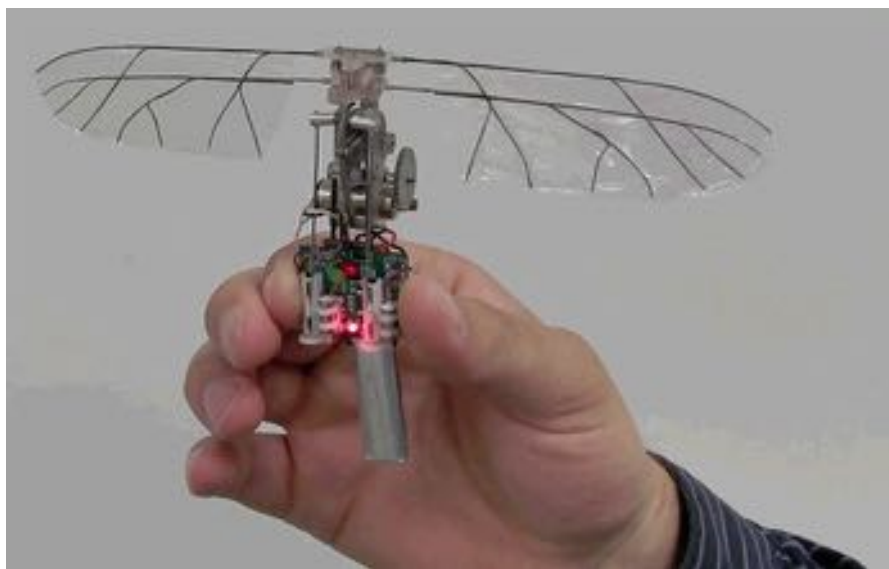


Fig. 56.24 Clap-and-fling micro air vehicle developed at Wright State University

this biomimetic design, very lightweight flexible wings had to be developed which allowed the camber and twist to reverse symmetrically from forestroke to backstroke to modulate lift and drag for control purposes while looking in shape and color like its natural counterpart. A major challenge was the development of the wing



Fig. 56.25 Nano Hummingbird (AeroVironment Inc.) sitting in the palm of a hand and in hovering flight (Photos courtesy of the AeroVironment Corporation)

flapping mechanism. A string mechanism was selected. Two strings, attached to a continuously rotating crankshaft driven by an electric motor, are each connected to two pulleys which are mounted on the wing hinge flapping axes. As the crankshaft turns, the pulleys oscillate. Two additional strings connecting the pulleys keep the two pulleys (and wings) matched in phase. This mechanism made it possible to generate control moments in the roll, pitch, and yaw axes using only the flapping wings. This was accomplished by modulating the wing rotation and the wing twist. The complete vehicle weighs 19 g and has a wing span of 16.5 cm. It can hover for several minutes and fly forward up to 6.7 m/s. Further details are described in the recent paper by Keennon et al. (2012). The Nano Hummingbird is the only known model that hovers with just two wings and provides all stability and control through the same wings with no tail surfaces at all.

56.5 Concluding Remarks

Twenty years ago, flapping-wing-propelled *micro air vehicles* were of primary interest only to children and enthusiasts who wanted to fly rubber band-powered airplanes. This situation changed rapidly after the Defense Advanced Research Projects Agency's call for micro air vehicle proposals in 1996. As described in the preceding pages, the basic physics of flapping-wing aerodynamics, acquired in the early twentieth century, was sufficiently well understood to demonstrate several flapping-wing-propelled air vehicles for hovering and forward flight. The amazing progress in battery power density and electric motor miniaturization made it possible to switch from rubber band power to electric power.

The above-described five micro air vehicles were largely developed by an iterative cut-and-try approach. However, great progress has been made over the past 20 years in acquiring experimental data and developing computational methods to improve knowledge of the complex unsteady three-dimensional low Reynolds number aerodynamic phenomena occurring on various flapping-wing propelled vehicles. This progress is evidenced by the large number of papers published in recent years on various facets of flapping-wing vehicles. It will undoubtedly lead to further innovative designs of such vehicles in the near future.

Disclaimer

This chapter is declared a work of the U.S. government and is not subject to copyright protection in the U.S.

References

- I.H. Abbott, A.E. von Doenhoff, *Theory of Wing Sections* (Dover, New York, 1959)
- A. Betz, Ein Beitrag zur Erklarung des Segelfluges. Zeitschrift fuer Flugtechnik und Motorluftschiffahrt **3**, 269–272 (1912)
- W. Birnbaum, Das ebene Problem des schlagenden Fluegels. Zeitschrift fuer angewandte Mathematik und Mechanik (ZAMM) **4**, 277–292 (1924)
- J.C. Dawson, T.J. Smith, P.G. Huang, D.B. Doman, Figure-8 flapping micro air vehicle, AIAA 2011–3792 (2011)
- G.C.H.E. de Croon et al., Design, aerodynamics, and vision-based control of the DelFly. Int. J. Micro Air Veh. **1**(2), 71–97 (2009)
- I.E. Garrick, Propulsion of a flapping and oscillating airfoil. NACA report 567 (1936)
- C.K. Hsu, J. Evans, S. Vytla, P.G. Huang, Development of flapping wing micro air vehicles—design, CFD, experiment and actual flight, AIAA 2010–1018 (2010)
- K. Isogai, S. Fujishiro, T. Saitoh, M. Yamamoto, M. Yamasaki, M. Matsubara, Unsteady three-dimensional flow simulation of a dragonfly hovering. AIAA J. **42**(10), 2053–2059 (2004)
- K.D. Jones, T.C. Lund, M.F. Platzer, Experimental and computational investigation of flapping wing propulsion for micro air vehicles, Chapter 16, in Progress in Astronautics and Aeronautics, Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications, AIAA, vol. 195 (2000)
- K.D. Jones, C.J. Bradshaw, J. Papadopoulos, M.F. Platzer, Bio-inspired design of flapping-wing micro air vehicles. Aeronaut. J. **109**(1098), 385–393 (2005)
- N. Joukowski, On the adjunct vortices. Obshchestvo ljubitelei estestvoznaniia, antropologii i etnografi, Moskva, Izvestiia, 112. Trans. Phys. Sect. **13**, 12–25 (1907)
- R. Katzmayr, Effect of periodic changes of angle of attack on behavior of airfoils, NACA TM 147 (1922)
- M. Keennon, J.M. Grasmeyer, Development of the black widow and microbat MAVs and a vision of the future of MAV design, AIAA 2003–3327 (2003)
- M. Keennon, K. Klingebiel, H. Won, A. Andriukov, Development of the nano hummingbird: a tailless flapping wing micro air vehicle, AIAA 2012–0588 Jan 2012
- R. Knoller, Die Gesetze des Luftwiderstandes. Flug- und Motortechnik (Wien) **3**(21), 1–7 (1909)
- H.G. Kuessner, Zusammenfassender Bericht ueber den instationaeren Auftrieb von Fluegeln. Luftfahrt-Forschung **13**, pp. 410–424 (1936)

- D.F. Kurtulus, A. Farcy, N. Alemdaroglu, Unsteady aerodynamics of flapping airfoil in hovering flight at low Reynolds numbers, *AIAA* 2005–1356 (2005)
- W.M. Kutta, Auftriebskraefte in stromenden Fluessigkeiten. *Illustrierte Aeronautische Mitteilungen* **6**, 133–135 (1902)
- F.O. Lehmann, S.P. Sane, M.H. Dickinson, The aerodynamic effects of wing-wing interaction in flapping insect wings. *J. Exp. Biol.* **201**, 3075–3092 (2005)
- D. Lentink et al., The scalable design of flapping micro air vehicles inspired by insect flight, chap. in *Flying Insects and Robots*, ed. by D. Floreano et al. (Springer, Heidelberg/New York, 2009)
- O. Lilienthal, *Der Vogelflug als Grundlage der Fliegekunst* Harenberg Edition, 3rd Printing, R. Gaertner's Verlagsbuchhandlung, 1992
- M.F. Platzer, K.D. Jones, J. Young, J.C.S. Lai, Flapping-wing aerodynamics: progress and challenges. *AIAA J.* **46**(9), 2136–2149 (2008)
- L. Prandtl, Ueber die Entstehung von Wirbeln in der idealen Fluessigkeit, mit Anwendungen auf die Tragfluegeltheorie und andere Aufgaben, in *Hydro-und Aeromechanik*, ed. by Th. Von Karman, T. Levi-Civita (Julius Springer, Berlin, 1924)
- L. Prandtl, “Tragfluegeltheorie”, *Goettinger Nachrichten*, mathematisch-physikalische Klasse (1918), 451–477. Reprinted in *Vier Abhandlungen zur Aerodynamik und Hydrodynamik*, Kaiser Wilhelm Institut fuer Stroemungsforschung, Goettingen, 1927
- S.P. Sane, The aerodynamics of insect flight. *J. Exp. Biol.* **206**, 4191–4208 (2003)
- S.P. Sane, Steady or unsteady? Uncovering the aerodynamic mechanisms Of insect flight. *J. Exp. Biol.* **214**, 349–351 (2011). Available online at <http://jeb.biologists.org/content/214/3/349>
- W. Schmidt, Der Wellpropeller, ein Neuer Antrieb fuer Wasser-, Land-, und Luftfahrzeuge. *Zeitschrift fuer Flugwissenschaften* **13**, 427–479 (1965)
- W. Shyy, Y. Lian, J. Tang, D. Viieru, H. Liu, *Aerodynamics of Low Reynolds Number Flyers* (Cambridge University Press, Cambridge/New York, 2008)
- T. Theodorsen, General theory of aerodynamic instability and the mechanism of flutter. NACA report 496 (1935)
- H. Wagner, Ueber die Entstehung des dynamischen Auftriebs an Tragfluegeln. *Zeitschrift fuer angewandte Mathematik und Mechanik (ZAMM)* **5**, 17–35 (1925)
- T. Weis-Fogh, Quick estimates of flight fitness in hovering animals, including novel Mechanisms for lift production. *J. Exp. Biol.* **59**, 169–230 (1973)
- R.J. Wootton, J. Kukalova-Peck, Flight adaptation in palaeozoic Palaeoptera. *Biol. Rev. Camb. Philos. Soc.* **75**(1), 129–167 (2000)
- M. Yamamoto, K. Isogai, Measurement of unsteady fluid dynamic forces for a mechanical dragonfly model. *AIAA J.* **43**(12), 2475–2480 (2005)
- J. Young, J.C.S. Lai, Oscillation frequency and amplitude effects on the wake of a plunging airfoil. *AIAA J.* **42**(10), 2042–2052 (2004)
- J. Young, J.C.S. Lai, Mechanisms influencing the efficiency of oscillating airfoil propulsion. *AIAA J.* **45**(7), 1695–1702 (2007)

Jayant Ratti and George J. Vachtsevanos

Contents

| | | |
|--------|--|------|
| 57.1 | Introduction | 1386 |
| 57.2 | State of the Art | 1387 |
| 57.3 | Quad-Wing MAV Design: Background | 1388 |
| 57.3.1 | MAV Design Conception: QV Design | 1389 |
| 57.3.2 | QV Developmental Plan | 1389 |
| 57.4 | Prototype Design and Construction | 1393 |
| 57.4.1 | Hardware-in-Loop-Simulation (HILS) | 1393 |
| 57.5 | Conclusion | 1395 |
| 57.6 | Energy Reserve Enhancement | 1395 |
| 57.6.1 | Energy Enhancement: Multi-wing MAV | 1395 |
| 57.7 | Fixed Frequency, Variable Amplitude | 1395 |
| 57.7.1 | Non-linear State Space Model for Each Wing | 1396 |
| 57.8 | Aerodynamics | 1398 |
| 57.8.1 | Grid and Wing Mesh | 1398 |
| 57.8.2 | Vortex Formation | 1398 |
| 57.9 | Flapping Hardware and Prototyping | 1399 |
| 57.9.1 | Load Cell and Wind Tunnel Rig Designs | 1399 |
| 57.9.2 | Feathering Mechanism | 1400 |

This work was supported by the Air Force Office of Scientific Research under contract No. FA9550-10-C- 0036. We acknowledge their support.
This article was originally published in the *Journal of Intelligent & Robotic Systems* (January 2012, Volume 65, Issue 1–4, pp 437–455)

J. Ratti (✉)

Robotics & Intelligent Machines, TechJect Inc., Atlanta, GA, USA

e-mail: jayantratti@gatech.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: george.vachtsevanos@ece.gatech.edu

| | | |
|---------|---|------|
| 57.10 | Micro Architecture and Control (MARC) Avionics Platform | 1400 |
| 57.10.1 | Avionics Design | 1402 |
| 57.11 | Real-Time Flight Control | 1402 |
| 57.11.1 | Real-Time Operating System Architecture | 1404 |
| 57.11.2 | Results: Power Drain and Weight | 1404 |
| 57.12 | MAV Designs Based on Different Actuation Schemes | 1404 |
| 57.12.1 | Modular MAV Configurations | 1404 |
| 57.12.2 | 6 DoF Testing of the First MAV Prototype | 1404 |
| 57.13 | Simulation Results | 1406 |
| 57.13.1 | Constant Gain: Manually Selected | 1406 |
| 57.13.2 | GA-Based Gain Tuning and Performance Improvement | 1406 |
| 57.13.3 | Visualization Environment and Simulation | 1410 |
| 57.14 | Conclusions | 1411 |
| | References | 1412 |

Abstract

In recent years, research efforts have focused on the design, development, and deployment of unmanned systems for a variety of applications ranging from intelligence and surveillance to border patrol, rescue operations, etc. Micro aerial vehicles are viewed as potential targets that can provide agility and accurate small area coverage while being costeffective and can be easily launched by a single operator. The small size of MAVs allows such flight operations within confined space but the control effectors must provide sufficient maneuverability, while maintaining stability, with only limited sensing capability onboard the platform. To meet these challenges, researchers have long been attracted by the amazing attributes of biological systems, such as those exhibited by birds and insects. Birds can fly in dense flocks, executing rapid maneuvers with g-loads far in excess of modern fighter aircrafts, and yet never collide with each other, despite the absence of air traffic controllers. This chapter introduces a novel framework for the design and control of a micro air vehicle. The vehicle's conceptual design is based on biologically inspired principles and emulates a dragonfly (Odonata-Anisoptera). A sophisticated multilayered hybrid and linear/non-linear controller to achieve extended flight times and improved agility compared to other rotary and flapping wing MAV designs. The chapter addresses the design and control features of the proposed QV design and gives an overview on the developmental efforts towards the prototyping of the flyer. The potential applications for such a high-endurance vehicle are numerous, including airdeployable mass surveillance in cluster and swarm formations. The disposability of the vehicle would help in battlefield deployment as well, where such a, MAV would be made available to soldiers for proximity sensing and threat level assessment. Other applications would include search and rescue operations and civilian law enforcement.

57.1 Introduction

In recent years, many military and civilian missions in an urban setting require improved surveillance and reconnaissance. To address the need for real-time

intelligence, unmanned aerial vehicles (UAVs) have been used to provide video surveillance. Of particular interest in urban operations are MAVs that require minimum operator workload, are user-friendly, and can be deployed in closed quarters or in dense and cluttered urban areas with their primary mission to detect and report target positions.

Of particular interest in urban operations are MAVs that require minimum operator workload, are user-friendly, and can be deployed in closed quarters or in dense and cluttered urban areas with their primary mission to detect and report target positions. The successful deployment and application of such micro flyers warrants that the MAVs be able to sustain flight for appreciably long durations. The prospect of such an MAV flight has so far eluded researchers. The MAV platforms inspired from flapping wing, biological counterparts have shown very limited flight endurance (Cox et al. 2002; Grasmeyer and Keenon 2001; Kornbluh 2002; Samuel et al. 2000; Pornsin sirirak et al. 2000). The majority of flight-capable flapping-wing MAVs have very limited VTOL/hovering capabilities. Their primary functionality is to mimic fixed-wing flight with the primary difference of producing forward thrust by the aid of flapping air backwards. The requirement is to have MAVs be able to hover and sustain long-flight endurance at the same time. The simultaneous inclusion of these two primary requirements in an MAV calls for a paradigm shift in the development of enabling actuation mechanisms and control techniques.

Missions envisioned for MAVs, such as intelligence, surveillance, and reconnaissance (ISR) in urban environments, require exceptional long flight times and significant sensor payloads. These missions typically involve flight through or in close proximity to buildings, tunnels, and trees and require the MAVs to maneuver using sharp dives and climbs along with small-radius turns. The small size of MAVs allows such dynamic flight operations within confined space but the control effectors must provide sufficient power for both dynamic maneuvers and long flight times. Also MAVs operate in a very sensitive Reynolds number regime, wherein aerodynamic flow physics exhibits strong variance from conventional steady aerodynamic effects seen over the wings of fixed and rotary wing MAVs. Hence, designing vehicles that can efficiently fly in this regime represents an entirely new challenge to design engineers (Table 57.1).







57.2 State of the Art

Table 57.2 (<http://www.ornithopter.org>) showcases some of the state-of-the-art MAVs that have been developed over the years (Cox et al. 2002; Grasmeyer and Keenon 2001; Kornbluh 2002; Samuel et al. 2000; Pornsin sirirak et al. 2000). At present, a uniformity among most of the MAV designs can be seen from their propulsion/actuation, a variation of a simple reciprocating crankshaft mechanism, symmetric flapping through mechanical coupling to a single rotary actuator, restricted control schemes, and low energy efficiency attributed to the use of frequency modulated flapping, as opposed to amplitude modulation, as used in insect and bird flight (Cox et al. 2002; Pines and Bohorquez 2006; Ratti and

Table 57.1 DARPA: MAV definition/design requirements

| Specifications | Requirements | Details |
|----------------|-------------------|------------------------|
| Size | <15.24 cm (6 in.) | Maximum dimension |
| Weight | ~ 100 g | Objective GTOW |
| Range | 1–10 km | Operational range |
| Endurance | 60 min | Loiter time or station |
| Altitude | <150 m | Operational ceiling |
| Speed | 15 m/s | Maximum flight time |
| Payload | 20 g | Mission dependent |
| Cost | \$1,500 | Maximum cost |

Table 57.2 Present state of the art 2000–2002

| | | |
|---|---|---|
| <p>2000</p>  <p>MicroBat</p> | <p>2000s</p>  <p>Univ. of Florida</p> | <p>2002</p>  <p>Mentor</p> |
| <p>2006</p>  <p>Delfly</p> | <p>2007</p>  <p>Nathan Chronister</p> | <p>2009</p>  <p>Petter Muren</p> |

Vachtsevanos 2010; Rayner 1979). Elastic storage and reuse of pectoral muscle energy is not facilitated. Complete 6 DoF dynamics is attained by addition of an airplane-like tail rudder or a tail elevator.

57.3 Quad-Wing MAV Design: Background

Limitations of fixed, rotary wing technologies heighten with size, following the Reynolds number decrease with smaller wing surfaces. (Aerodynamic effects in the low Reynolds number region are still a subject of much research (Prosser et al. 2011).) However, insects and birds are adept flyers at low Reynolds numbers. A lot of research in the unsteady Reynolds number region has also supported the fact (Pines and Bohorquez 2006). Flight comparison of the flying mechanisms between

a bird, hummingbird, butterfly, and dragonfly (Ratti and Vachtsevanos 2010) was made to choose the most suitable MAV design (Fig. 57.2).

57.3.1 MAV Design Conception: QV Design

Flight mechanism comparison between a bird, hummingbird, butterfly, and a dragonfly was made to choose the optimal MAV design; a dragonfly-like design (Ratti et al. 2010; Ratti and Vachtsevanos 2010) chosen as the inspiration for the QV (QV stands for quad-winged MAV. The name has been chosen to abbreviate the most basic appearance of the proposed quad winged MAV design) design was born. Figure 57.3 illustrates the in-flight control of primitive maneuvers by the coordinated control of power distribution to the individual wings.

57.3.2 QV Developmental Plan

The MAV program has been centered around the parallel research and development scheme of various software/hardware elements illustrated in Fig. 57.4 (Prosser et al. 2011; Ratti et al. 2010, 2011a; Ratti and Vachtsevanos 2010, 2011).

An actuation mechanism converting a rotary motion to reciprocating motion by the use of a crank mechanism exhibits this problem. The system amplitude is always constant, and thus the system can accelerate or decelerate based on how fast the wings are flapping. For generating higher thrusts, the wings are simply made to move faster at the fixed amplitude of motion. But for a fixed frequency flight system like in the case of insects and birds, changing flapping frequency disturbs the spring-mass system dynamics, resulting in more energy consumption and poor performance. Hence most of the flyers do away with the spring altogether, thereby losing a lot of flight performance exhibited by birds and insects.

The idea of using a system which maintains constant frequency but varying amplitude better approximates the flight dynamics of insects and birds. Such a system has been attempted but never implemented in a flying prototype (Michelson et al. 1997; Michelson 1988; Keshavan et al. 2007; Raney and Slominski 2004). The mathematical model of the actuation mechanism has been developed by Ratti and Vachtsevanos (2010).

The system model is based on two coupled mechanisms in series, a four-bar linkage mechanism, and an inverse crankshaft mechanism.

As seen in Fig. 57.5, the linear actuator drives the wing up and down by pushing or pulling the wing. However to obtain an appreciable amplitude, the placement of the solenoid needs to be very close to the wing pivot, as shown in the Fig. 57.5. This raises linkage and structural deformation concerns. The four-bar linkage was utilized to overcome the linkage difficulties. The solenoids are attached much further from the wing pivots now, as shown in Figs. 57.5 and 57.6. The wing with the actuator, spring, and four-bar linkages has been shown in Fig. 57.5.

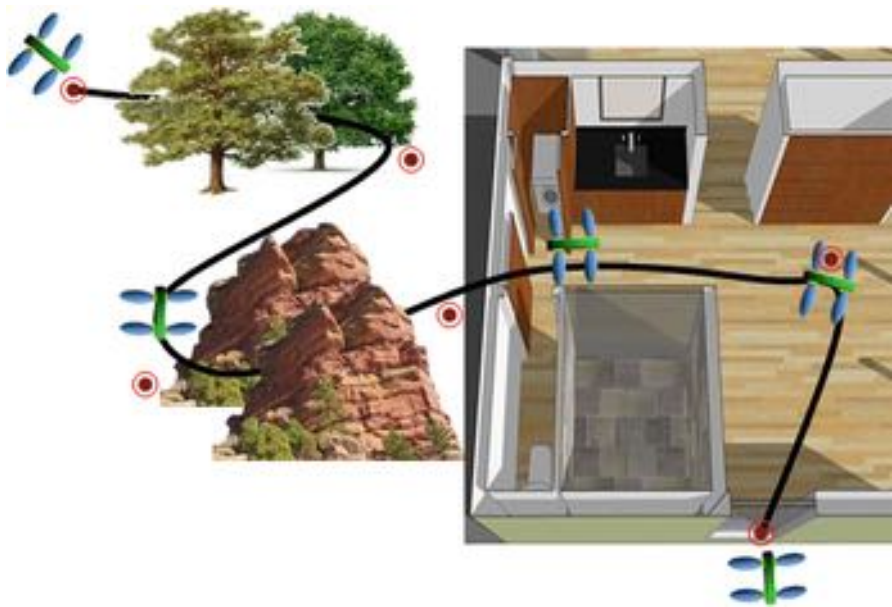


Fig. 57.1 Indoor/outdoor MAV operation

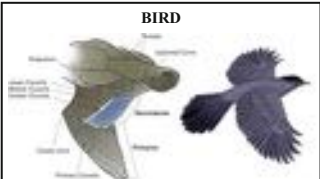



| BIRD | HUMMINGBIRD | BUTTERFLY |
|--|---|--|
|  |  |  |
| <p>A. <i>Complex co-ordination</i>: Many muscles</p> <p>B. <i>Larger wing-span</i> for long flight times</p> <p>C. <i>Not recommended</i> for closed-quarter flight</p> | <p>A. <i>Good Confender</i> for a design</p> <p>B. <i>Not power efficient and short flight time</i></p> <p>C. <i>Complex Wing mechanisms</i> implementation</p> | <p>A. <i>Excellent conlender</i> for a MAV</p> <p>B. <i>Long flight times</i></p> <p>C. <i>Slow dynamics, low agility</i></p> <p>D. <i>Low controllability</i></p> |
| DRAGONFLY - THE DESIGN CHOICE | | |
| <p>A. Four sets of wings provide <i>maximum Lifting power</i></p> <p>B. The Wings resonate synchronously, sustaining <i>super-long flight times</i></p> <p>C. Four wings give it unparalleled <i>agility and maneuverability</i></p> | | |
|  | | |
| <p>D. Only <i>one actuator per wing</i></p> <p>E. <i>Simple controls</i></p> <p>F. <i>Relatively less complex</i> parts-tolerance to damage</p> | | |

Fig. 57.2 Trade study of natural flyers: insects and birds

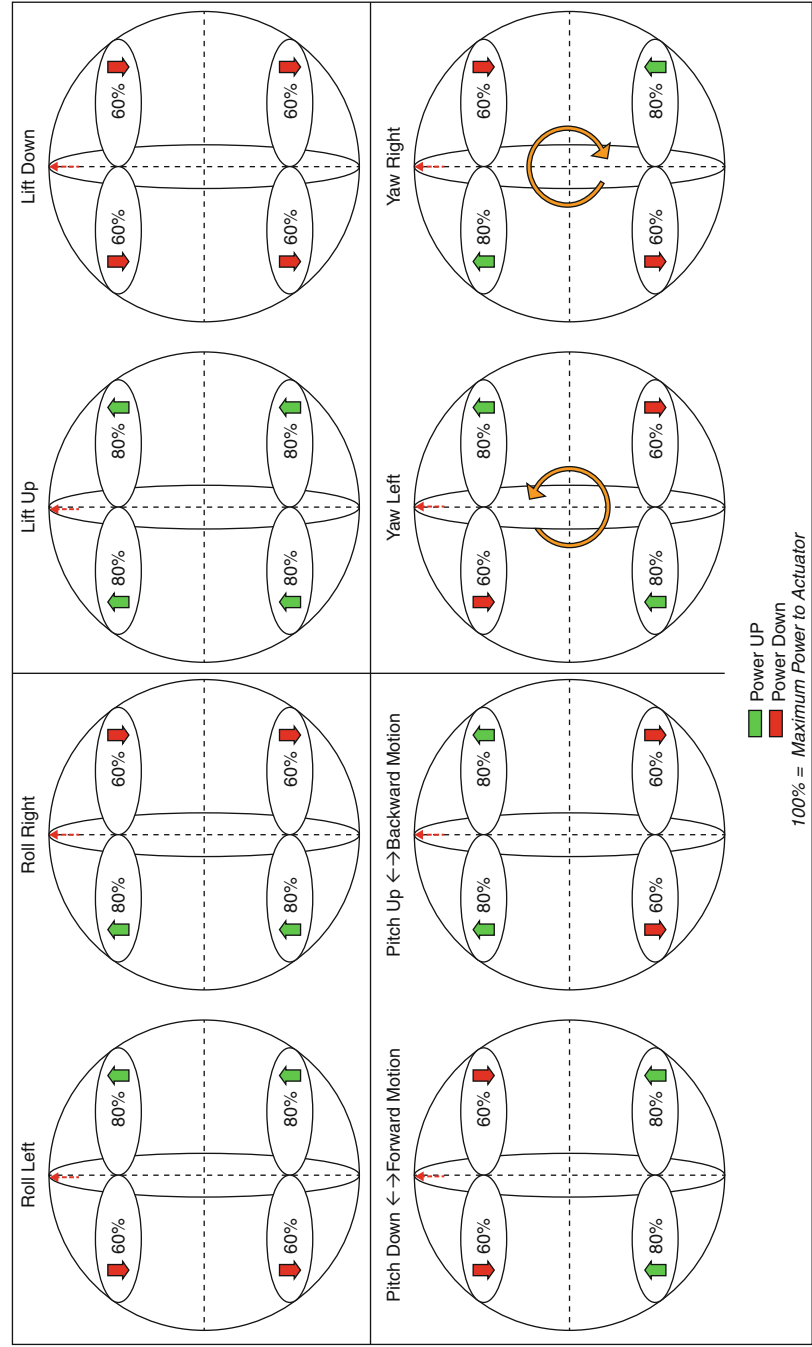


Fig. 57.3 Flight modes for the QV design

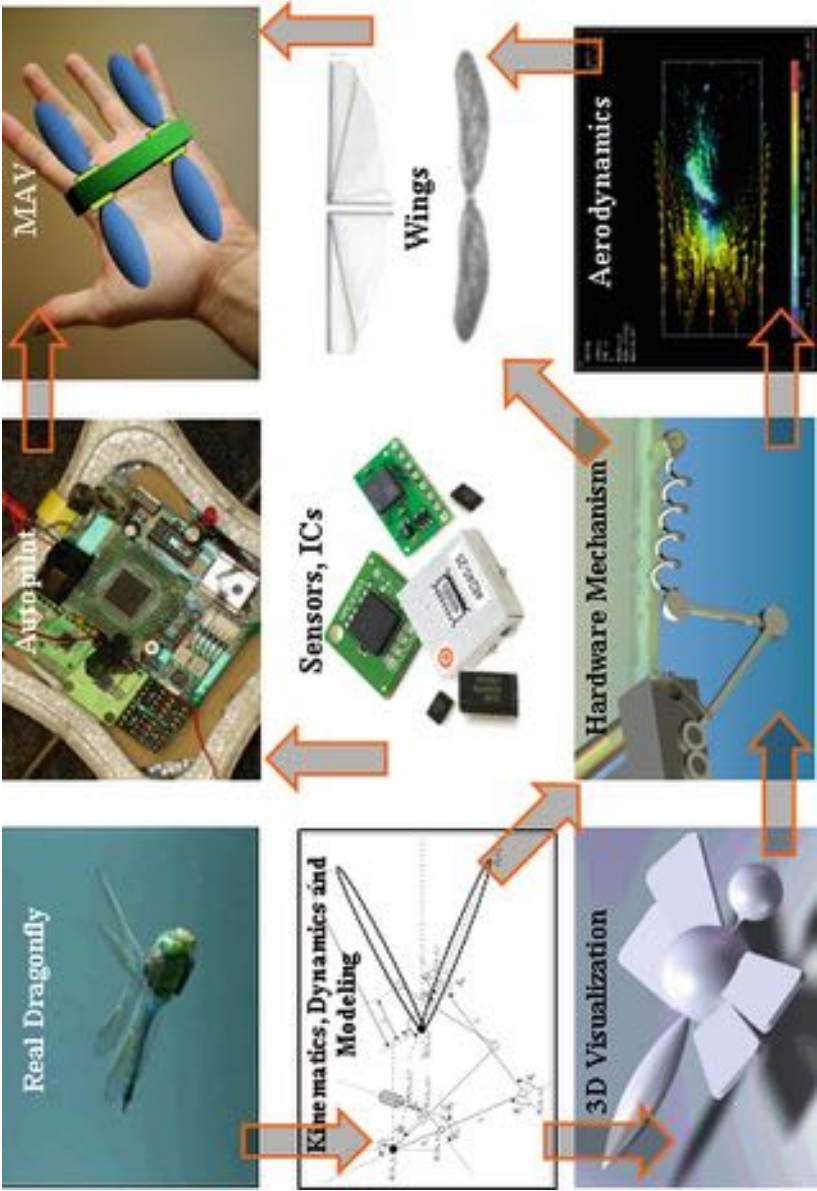


Fig. 57.4 Program objective and implementation flowchart

Fig. 57.5 Actuation mechanism using linear actuators: basic idea

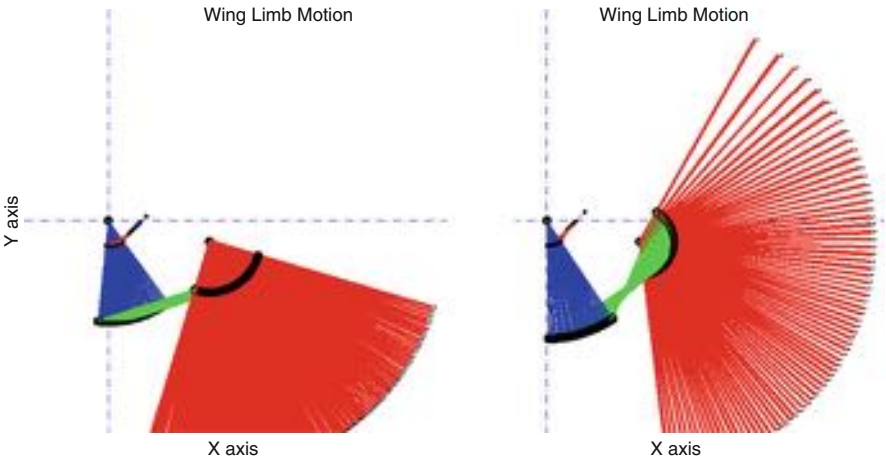
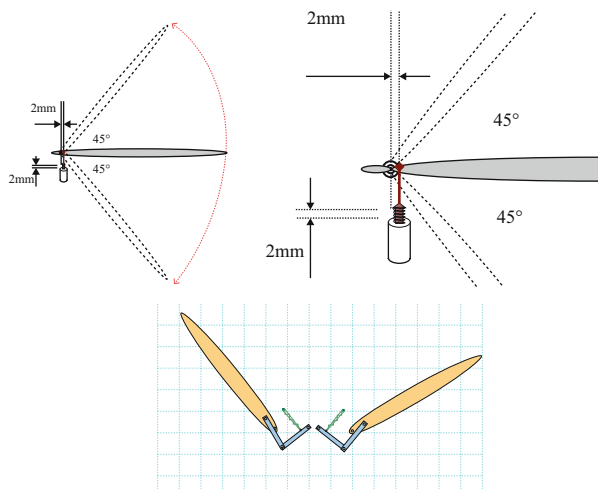


Fig. 57.6 Wing sweep: range of motion

57.4 Prototype Design and Construction

57.4.1 Hardware-in-Loop-Simulation (HILS)

The HILS setup is shown in Fig. 57.7. The flight dynamics have been modeled on a computer and the flight visualization has been linked to the dynamics. The model continuously sends out states of the system to the autopilot. The autopilot

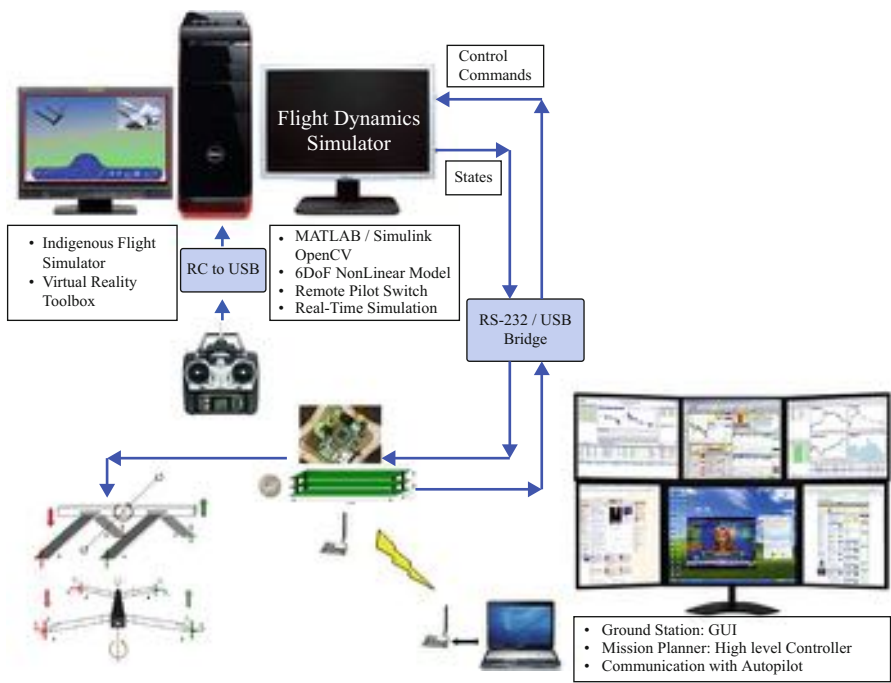


Fig. 57.7 Hardware in the loop simulation setup

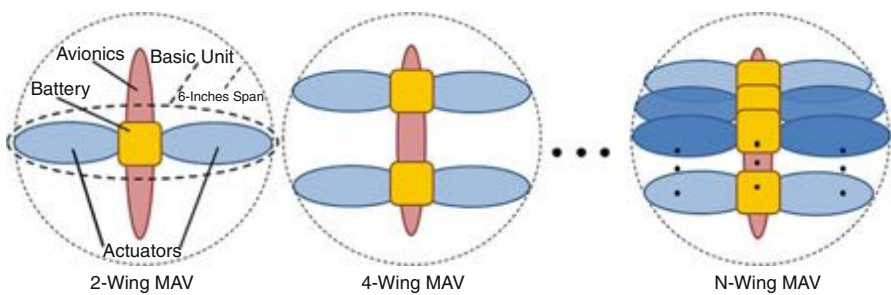


Fig. 57.8 2, 4, N-wing MAVs: repeating the basic unit

processes the control routines and sends back control commands to the computer (simulator/modeler). The ground station constantly keeps a communication lock with the autopilot and receives and sends mission status from time to time. The RC controller is connected to the modeler.

57.5 Conclusion

A new micro unmanned aerial vehicle design was developed and presented. This chapter focused on the review of the MAV conceptual design and the development of the constituent sensing and control algorithms. Simulation results demonstrated the efficacy of the design. The research seeks to achieve stable flying conditions by combining the unique design features of the MAV with novel sensing and control strategies. The objective is to integrate the MAV hardware and software modules into a device that will exhibit attributes of stable and robust flying conditions. The inspiration for the design stems from the study of various flying insects and other entomophthora developed in the past. The design was mathematically modeled and verified in simulation. A hierarchical control paradigm was pursued to capture various control requirements from actuation to trajectory following. A brief overview of the hardware mechanisms, embedded autopilot designs, and hardware-in-the-loop simulation architecture was presented. A laboratory prototype is currently under development (Fig. 57.25).

57.6 Energy Reserve Enhancement

The QV design's inherent advantage lies in providing the MAV with a manyfold increase in lift, thus the ability to carry higher payloads in the form of avionics and battery packs.

57.6.1 Energy Enhancement: Multi-wing MAV

From Fig. 57.9 (Ratti and Vachtsevanos 2011), the energy saving per actuator is demonstrated for multi-wing vehicles with changing avionics power consumption, showing energy savings of the order of 100 % and more in cases. Energy efficiency on the QV design goes beyond that provided by the four-wing configuration over a two-wing configuration alone; the inclusion of elastic/restorative wing flapping further improves the energy efficiency of the MAV. It has been proved (Ratti and Vachtsevanos 2011) that by adding a passive spring at the wing joints of the MAV, considerable torque (and in turn energy) savings is observed, which is greater for lower air-damping on the wings.

57.7 Fixed Frequency, Variable Amplitude

Definition 1 *Fixed Frequency, Variable Amplitude (FiFVA) Control Problem:* An actuation mechanism converting a rotary motion to reciprocating motion by the use of a crank mechanism exhibits a low efficiency state. The system amplitude

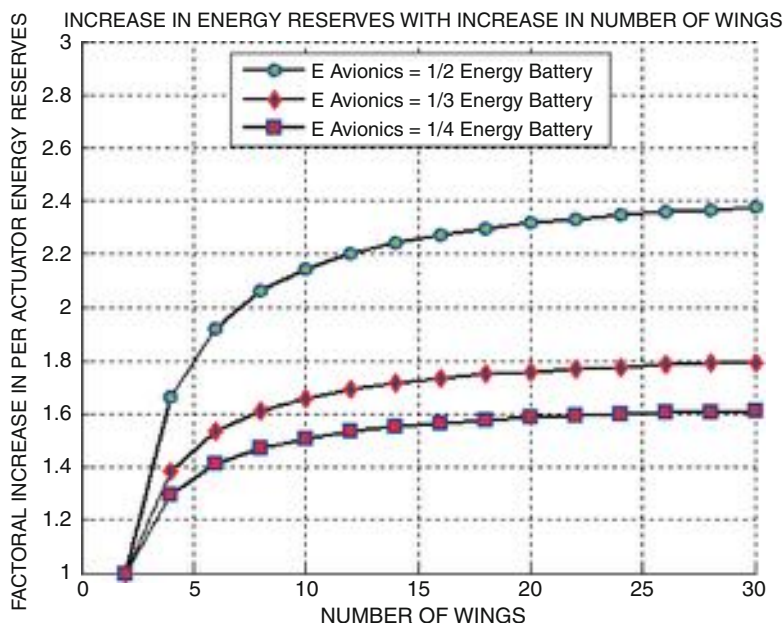


Fig. 57.9 Energy saving/flight time increase curves

is always constant, and thus the system can accelerate or decelerate based on how fast the wings are flapping (frequency modulation). But for a fixed frequency flight system (insects/birds), changing flapping frequency disturbs the spring-mass system dynamics, resulting in more energy consumption and poor performance.

57.7.1 Non-linear State Space Model for Each Wing

The idea of using on the QV, a system which maintains constant frequency but controls lift and maneuvering by amplitude modulation, better exhibits the flight dynamics of insects and birds. Such a system has been attempted but met with limited success (Keshavan et al. 2007; Raney and Slominski 2004). The mathematical model of the actuation mechanism has been defined in Ratti et al. (2010) and Ratti and Vachtsevanos (2010). The model is based on two coupled mechanisms in series, a four-bar linkage mechanism, and an inverse crankshaft mechanism (Fig. 57.10). Our two FiFVA actuators are as follows:

The wing state vector X bundles the dynamic relationships between the four-bar mechanism, the inverse crankshaft mechanism, the solenoidal linear motion, and current intake at an input u . The state space model representation is as shown:

$$\dot{X} = f(X, u), Y = g(X, u) \quad (57.1)$$

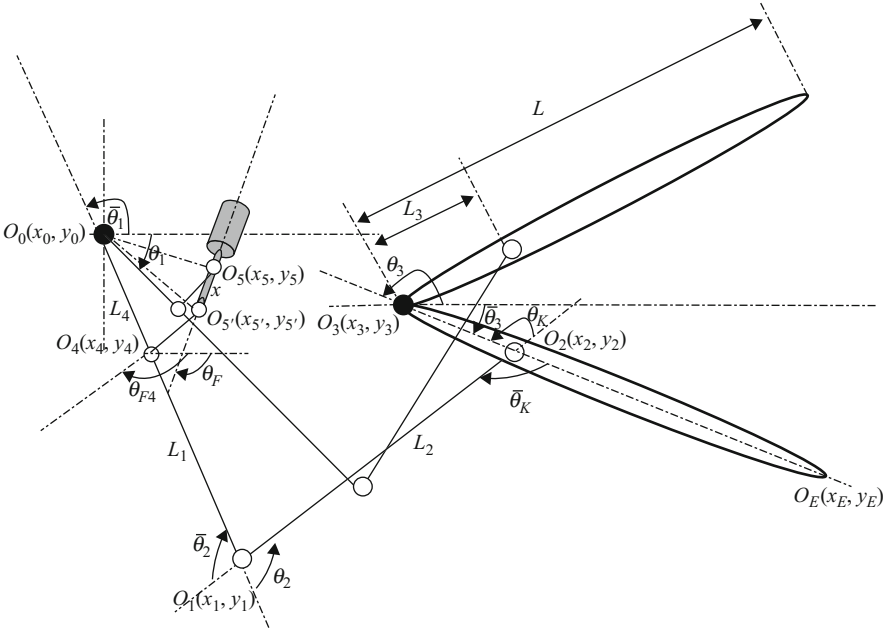


Fig. 57.10 Kinematic illustration of the basic wing mechanism

$$X = \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \theta_2 \\ \theta_k \\ \theta_3 \\ \dot{\theta}_3 \\ i \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}, \dot{X} = \begin{bmatrix} \text{Solenoid Linear Velocity} \\ \text{Solenoid Linear Acceleration} \\ \text{[Joint Angular Velocities]} \\ \text{Joint 3 Angular Acceleration} \\ \text{Solenoid Current Gradient} \end{bmatrix} \tag{57.2}$$

The dynamics of the wing motion depend on various configuration-specific elements and constants of the wing coupling which have been captured using X in Eq. 57.2.

The outputs for the system are the lifting forces produced by the wings of the MAV. The output model is shown below with its dependence on wing configuration parameters like shape and size and also on air drag and density coefficients (Fig. 57.11).

A full nonlinear model and state space derivation has been given in Ratti and Vachtsevanos (2010):

$$Y = K_{\text{constant}} \cdot x_7 |x_7| \begin{bmatrix} -\sin x_6 \\ \cos x_6 \end{bmatrix} \tag{57.3}$$

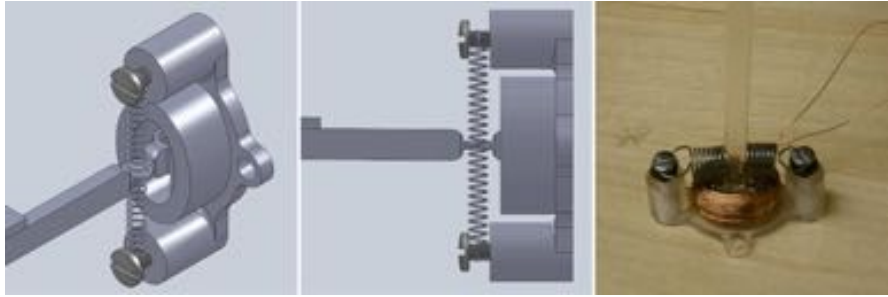


Fig. 57.11 Micro solenoidal actuator: CAD (*left*); working prototype (*right*)

1. *Solenoidal – One-spring configurations:* The solenoidal/magnetic actuation methodology was designed to produce FiFVA actuation and control using just one actuator with only one control/power input. The actuator was also miniaturized to weigh 5 g and produced an output torque of 75 g-cm. At 3.6 V, the coil drew 500 mA and reached a resonant frequency of 20 Hz (Ratti et al. 2011a).
2. *Hypocycloidal Gear Train:* A hypocycloidal gear train was examined since by the use of an additional worm gear arrangement, amplitude modulation is possible, providing FiFVA control over wing flapping (Fig. 57.12) (Ratti et al. 2011a).

57.8 Aerodynamics

57.8.1 Grid and Wing Mesh

The unsteady flapping + feathering motion of the cambered elliptical wing was simulated in Ansys Fluent V12.1. (Through a commercially available CFD software package.) Figure 57.13 shows the unstructured tetrahedral grid generated using *Gambit*. (A commercially available software for grid generation.) The overall grid was divided into two parts: an inner smaller spherical domain bounding the wing motion with a fine mesh to ensure good resolution of the flow around the wing and an outer spherical domain with a coarser mesh.

57.8.2 Vortex Formation

Figure 57.14 represents the formation of leading edge vortex which plays an important role in production of lift for unsteady flapping wings at low Reynolds number. At time $t = 0$, the wing is horizontal. Due to the increased acceleration of the wing from its initial position, a leading edge vortex grows from the base towards

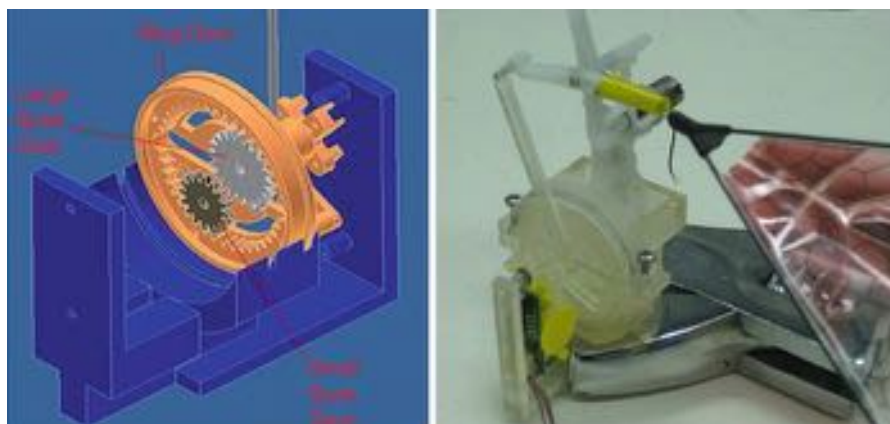


Fig. 57.12 Inter-meshing of the internal gears (*left*); working prototype (*right*)

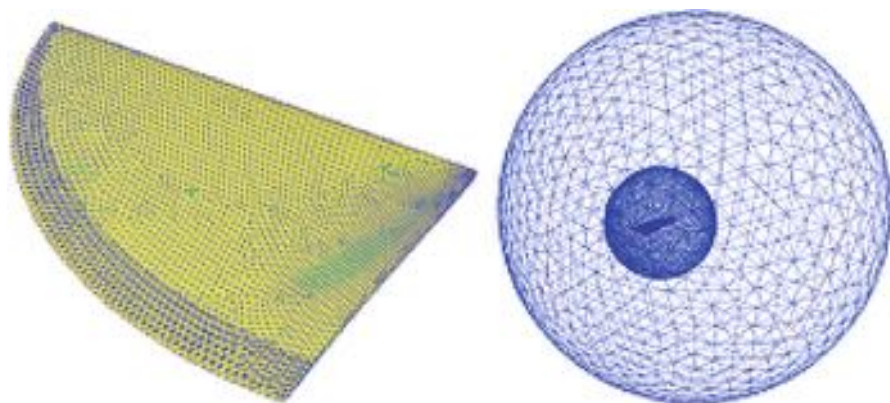


Fig. 57.13 Wing mesh (*left*); mesh grain/boundary (*right*)

the tip. The growth of the vortex is maximum at a point about $3/4$ of wing span. The formation of this vortex is believed to be related to the dynamic stall phenomenon due to which a wing produces high lift even at high angles of attack.

57.9 Flapping Hardware and Prototyping

57.9.1 Load Cell and Wind Tunnel Rig Designs

Figure 57.15 is a scale model of the wing actuation mechanism, connected to a 6-axis load cell.

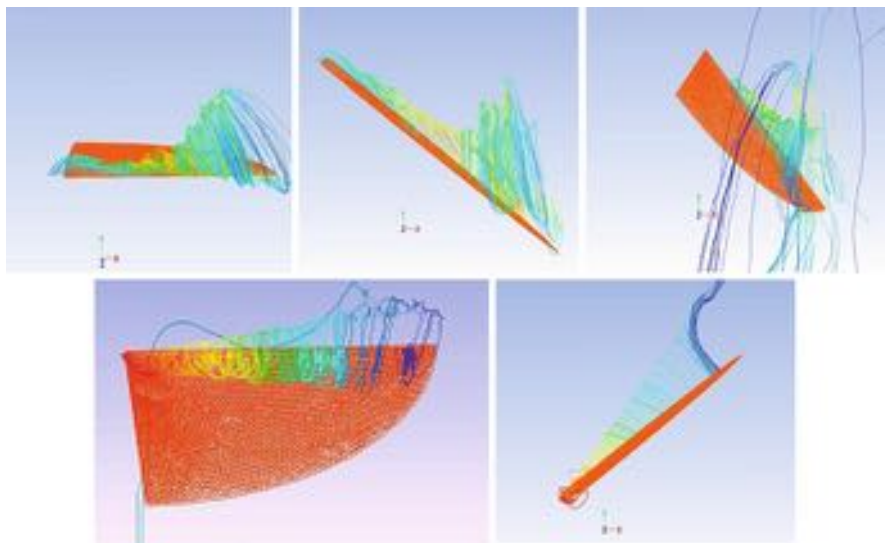


Fig. 57.14 Leading edge vortex dynamics on wing surface (from left): at 0 s, $T/8$ s, $T/4$ s, $T/2$ s, $7T/8$ s

57.9.2 Feathering Mechanism

Addition of an extra actuator proves bulky and difficult to install. The feathering mechanisms we have implemented constitute spring-loaded passive feathering of the wings to conform to fixed frequency, variable amplitude flapping as exhibited by birds and insects.

1. *Passive Feathering* – Designs: Some of the modular passive feathering wing designs tested on the flapping setup shown in Fig. 57.15 are presented in Fig. 57.16.

57.10 Micro Architecture and Control (MARC) Avionics Platform

The conventional mini- and large-scale UAV systems span anywhere from approximately 12 in. to 12 ft, endowing them with larger propulsion systems, batteries/fuel tanks, which in turn provide ample footprint and power reserves for onboard avionics and wireless telemetry. The conventional MAV (Table 57.1) can have a maximum dimension of 6 in. and weighs no more than 100 g. Under these tight constraints, the footprint, weight, and power reserves available to onboard avionics are drastically reduced. The chapter presents the advent of a new line of micro architecture and control (MARC) avionics systems with very lowpower, multisensor, multiprocessor avionics, interconnect architecture designed specifically with the power and payload constraints of MAVs in mind towards matching performance to their larger-scale UAV cousins.

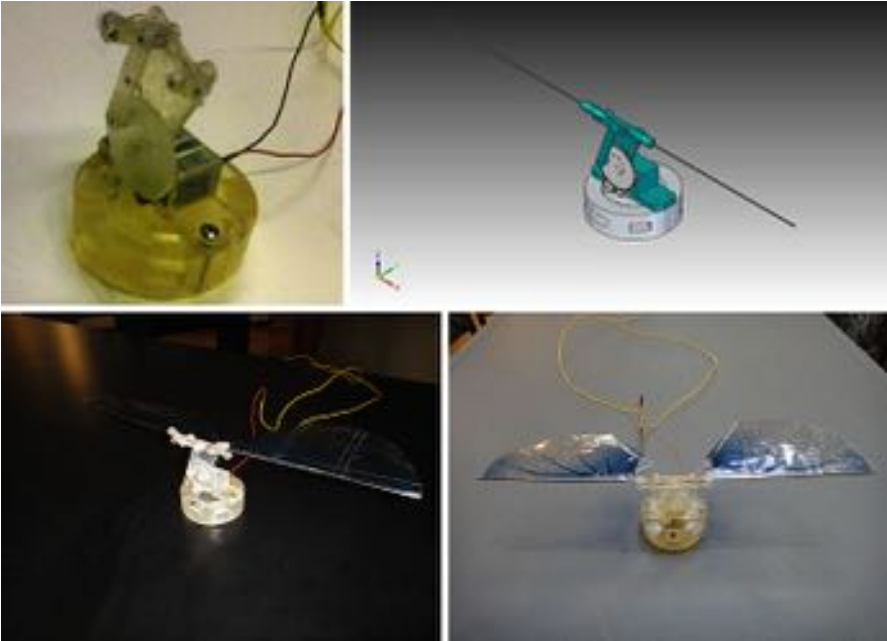


Fig. 57.15 Flapping mechanism

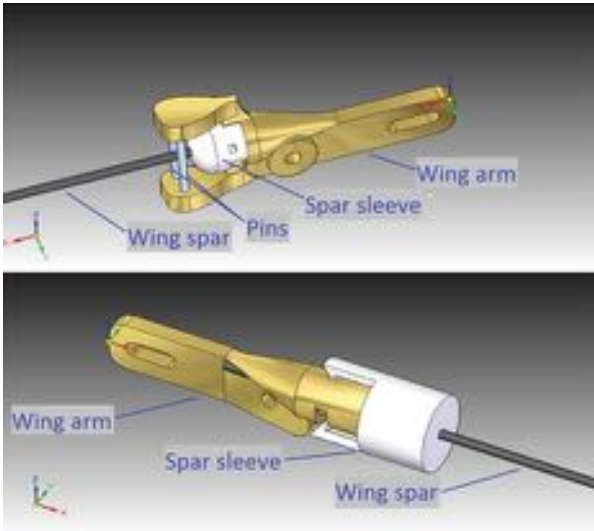


Fig. 57.16
Passive-feathering designs

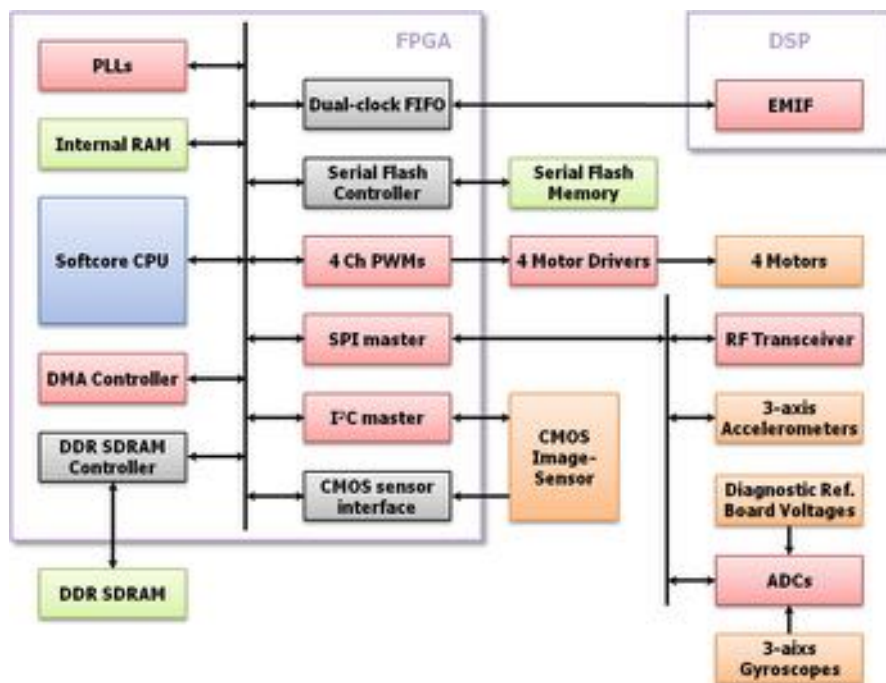


Fig. 57.17 Embedded hardware architecture

57.10.1 Avionics Design

Figure 57.17 shows the outline of the embedded hardware onboard the MAV. The selection of the processors for this embedded device involved many important considerations such as power consumption, size/weight, and performance.

1. *Embedded Vision and Wireless Telemetry*: The ground station tele-operations the MAV and/or uploads waypoints for semiautonomous flight navigation. The MARC collects internal state information and environmental data including aerial video/images during flight and transmits this data via a block data packet to the ground station also through the RF transceiver; Fig. 57.18 briefly illustrates the link.

57.11 Real-Time Flight Control

The MAV control architecture is illustrated through the flow diagram (Jung et al. 2009; Vachtsevanos 1998; Vachtsevanos and Ludington 2006) shown in Fig. 57.21. The *mission* module describes the global mission parameters to be met by the

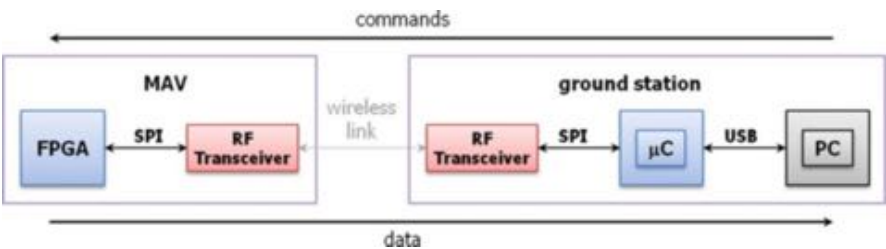


Fig. 57.18 Wireless link between the MAV and the ground station

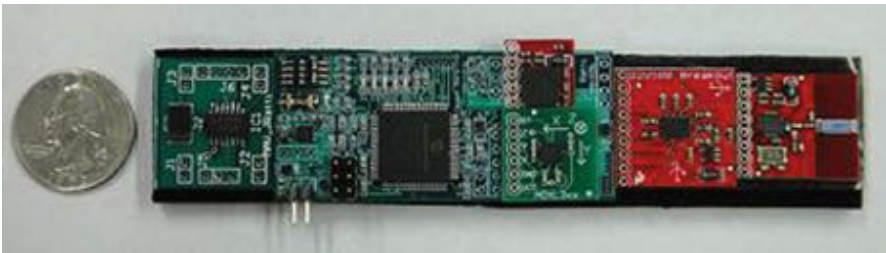


Fig. 57.19 MARC-1 (dsPIC33F) single core

MAV, such as flying from point A to point B to point C and avoiding point D. The *path/trajectory planning* module incorporates the above mission goals and assigns a trajectory to be followed to meet the mission goals. The *target positioning* layer produces reference Euler angles and rates for the MAV to successfully follow the waypoints produced in the previous higher layer. The *flight control* layer is a linear controller, which matches the vehicles dynamic Euler angles and rates to conform to the reference values generated by the immediately upper layer. The *hybrid energy controller* is responsible for sustaining oscillations about the wing-beat frequency. The energy controller controls the thrust produced by each wing by modulating the power output of each wing. There are four actuators in the system, which together control pitch (and forward/backward translation), yaw, roll (and left/right translation) (Figs. 57.19 and 57.20).

The MAV plant has been expanded for better understanding of the internals of the system. The plant can be seen to comprise the following as shown in Fig. 57.21. The system energy is calculated from output – sensor feedback and internal sensor feedback for the system – by taking into account the length and mass, angular position, spring extension, and angular velocity of the wing. The interface of the nonlinear controller can be seen from Fig. 57.21. The energy controller controls the power delivered to the actuators, in effect controlling the three body angular rates of the system. Details on the hybrid energy controller can be found under Ratti et al. (2010) and Ratti and Vachtsevanos (2010) (Fig. 57.21).



Fig. 57.20 MARC-2 (Cyclone III, TI 55xx DSP) dual core

57.11.1 Real-Time Operating System Architecture

The architecture of the hardware-in-the-loop (HILS) setup utilizing the RTOS (Jung et al. 2009) has been shown in Fig. 57.23. The architecture has been designed to keep the code flexible enough to allow adding higher level tasks to mission directives sent by the ground station. All this can be achieved without engrossing into the system-level intricacies of handling and programming a microcontroller/microprocessor. The flexibility of the architecture also makes it extremely efficient to debug faults in low-level, mid-level, or high-level tasks, without having to recode/interfere with the other tasks.

57.11.2 Results: Power Drain and Weight

The goal of the MARC-2 board was to keep all the avionics within a total weight of 15 g and the onboard avionics power requirement within 0.75 W. Towards that end the different processors and peripherals were selected. The total weight of the MARC-2 avionics suite was 15.75 g and the total power consumption was 0.65 W.

57.12 MAV Designs Based on Different Actuation Schemes

57.12.1 Modular MAV Configurations

At present there are four different MAV prototypes being constructed in our laboratory conforming to the QV design specifications; each has four wings to produce lift and 6 DoF control. Some of the 3D prototypes are shown in Fig. 57.24.

57.12.2 6 DoF Testing of the First MAV Prototype

A test-bench simulator has been designed to perform 3 DoF motions and maneuvers. The goal is to test different control algorithms and their performance on a bench-top

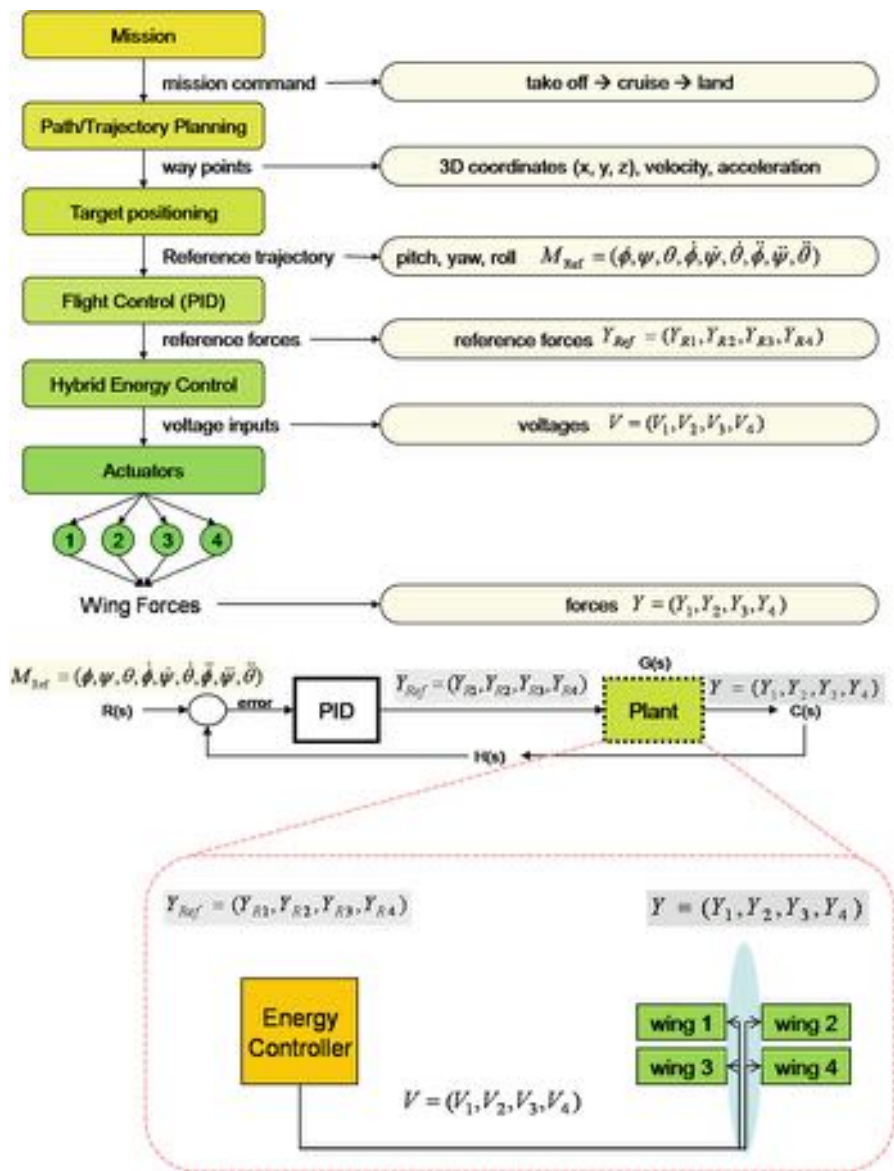
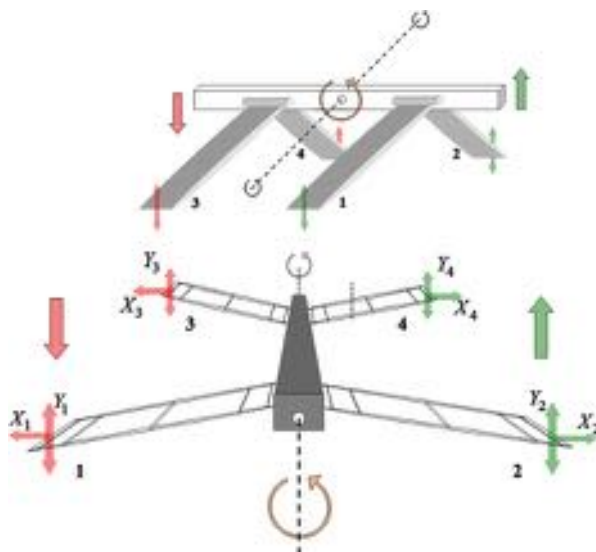


Fig. 57.21 MAV hierarchical control structure

setup without the need to initially subject the prototype to unnecessary wear and tear, crashes, and in-flight failures. The vehicle has to be made 6 DoF capable by the installation of an indigenous, low-weight autopilot (Ratti et al. 2011b) (Fig. 57.19 and 57.20).

Fig. 57.22 Linear control visualization



57.13 Simulation Results

The system step response for a given initial condition of $\phi_{\text{Ref}} = -12^\circ$, $\theta_{\text{Ref}} = 11^\circ$, $\psi_{\text{Ref}} = 11.5^\circ$ is shown in Fig. 57.26. The MAV has also been commanded to maintain a level altitude of 20 cm above its launch position (Fig. 57.27).

57.13.1 Constant Gain: Manually Selected

The PID gains for the mid-level controller have been kept constant at reasonably agreeable values in the first controller and the system response for the above reference parameters is shown in Fig. 57.26. The five graphs show pitch, roll, yaw, altitude, and actuator thrust response versus time.

57.13.2 GA-Based Gain Tuning and Performance Improvement

Typically, the gains in the PI controller are adjusted by observing characteristic factors such as rising time, settling time, and overshoot under a step response until the desired performance metrics are achieved. However, the QV system is highly nonlinear and the driving mechanism is underactuated as well; thus the choice of gains is highly experimental and cannot guarantee satisfactory performance. In order

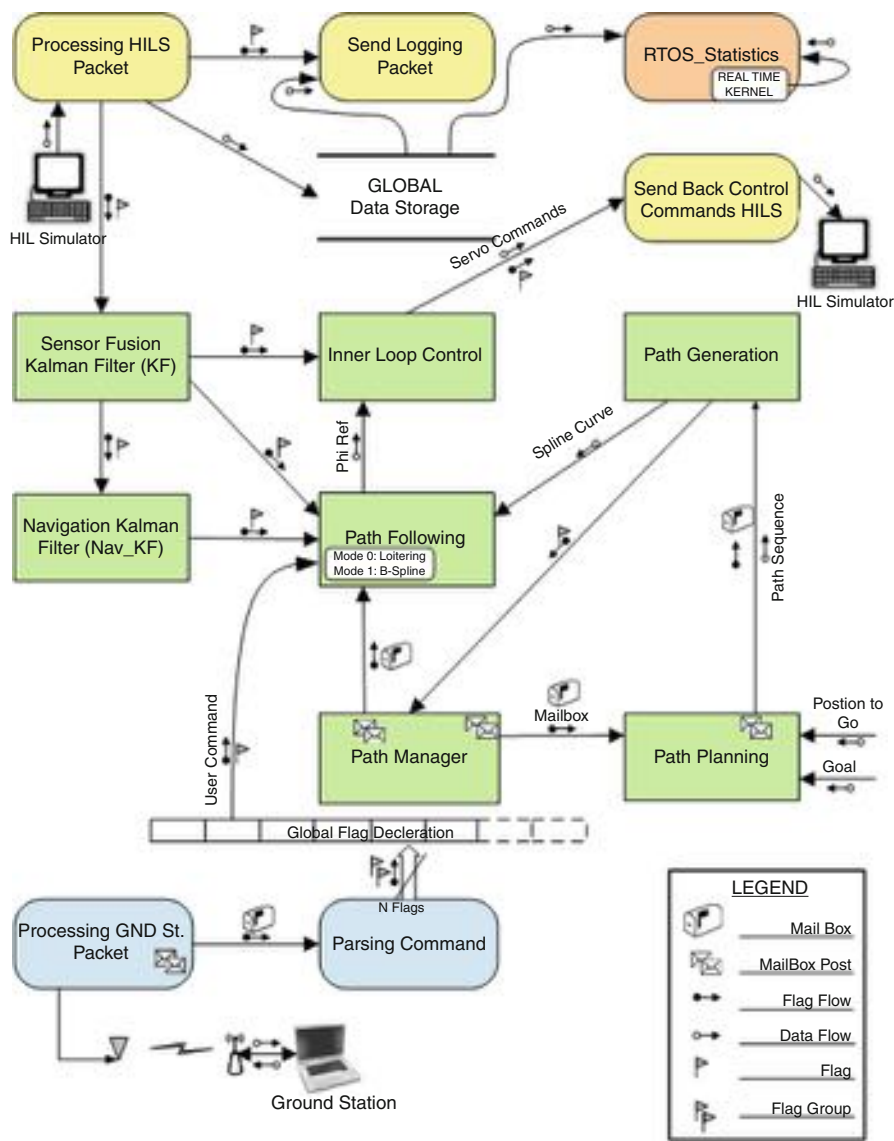


Fig. 57.23 RTOS architecture for flight-control and path planning

to optimize the step response performance of the MAV, a good set of PI controller gains are necessary; the genetic algorithm (GA) is applied towards that goal. The performance of the proposed system can be measured by the system response times to a step input.

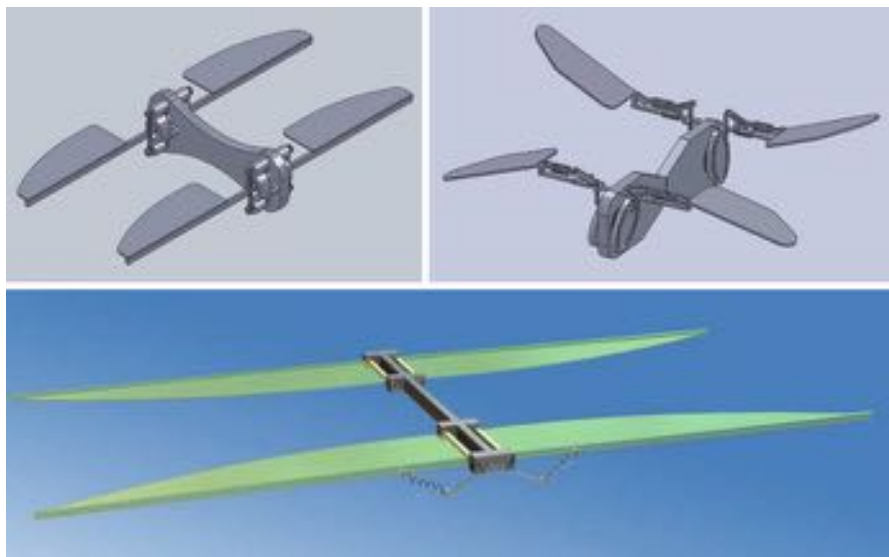


Fig. 57.24 MAV designs based on coil actuators, micro linear actuators, cam-follower drive trains

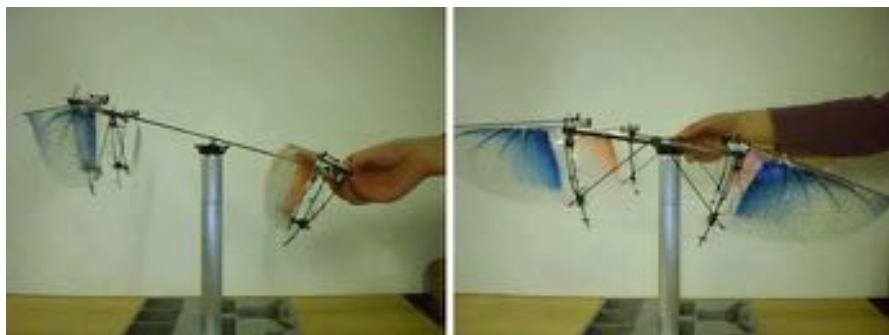


Fig. 57.25 3 DoF benchtop prototype

Cost Metric: The cost function for the GA is defined as the cumulative area under the step response graph until the system settles down. The smaller the area, the faster the response time.

Generally, in order to apply the GA to a selection problem, the gene structure and evaluation method should be prepared; in this research, the set of PI gains is defined as a gene and the fitness function is defined as the cost metric. The crossover rates and the mutation rates are set as 100 and 10 %, respectively. Figure 57.28 highlights the results of gain tuning using the devised GA after six generations.

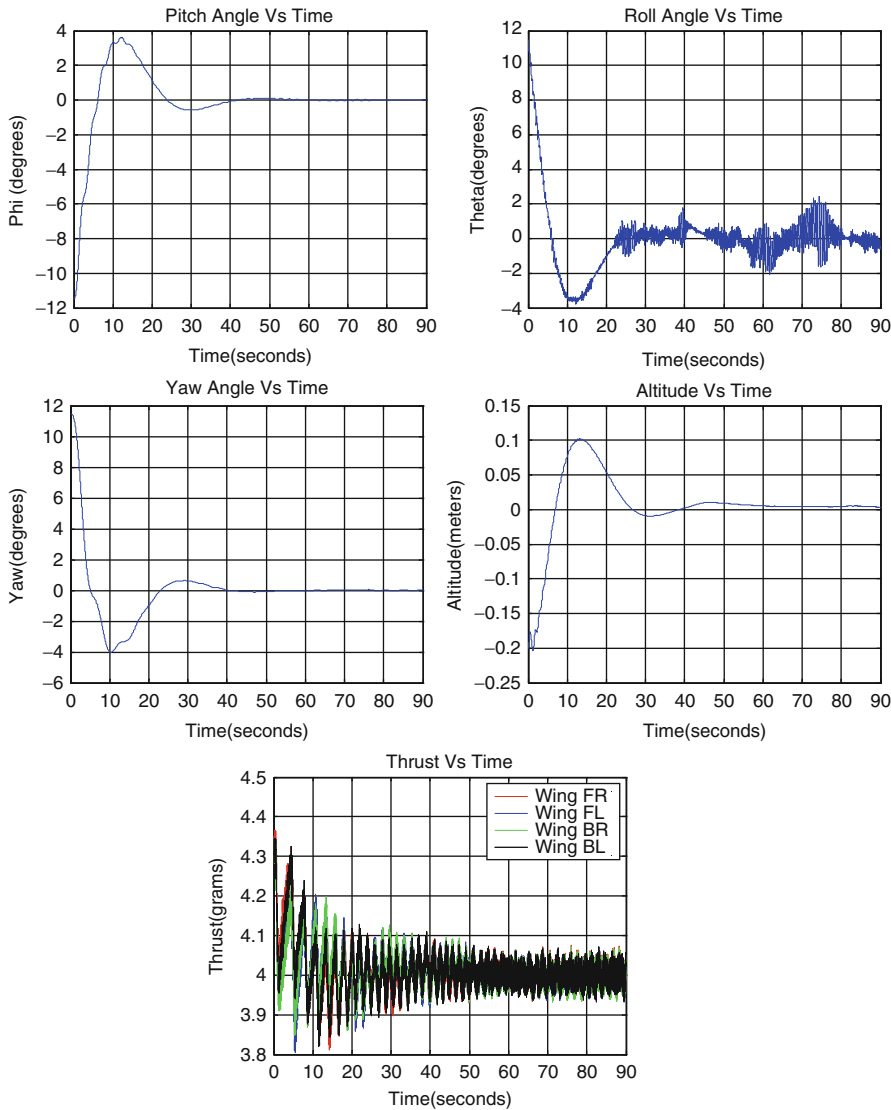


Fig. 57.26 Linear/hybrid energy control step response: manual PID gain selection

There has been a marked performance improvement over manual gain selection. Settling time for manual selection was about 50–60 s to start with, with an overshoot of approximately 4°. The GA-based gain tuning had 0° overshoot and the settling time has been reduced to approximately 5 s. The step response after GA tuning conforms to a critically damped system and has shown very promising results (Table 57.3).

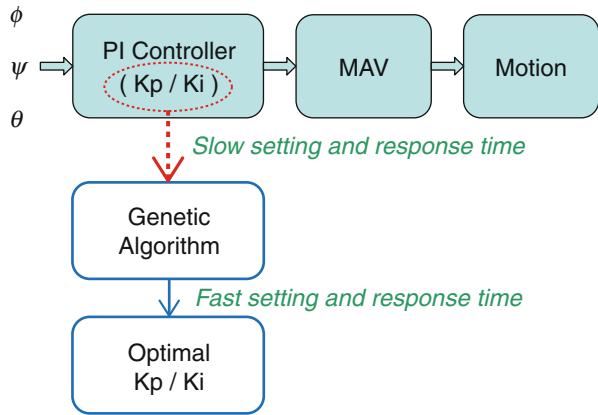


Fig. 57.27 GA tuning block diagram

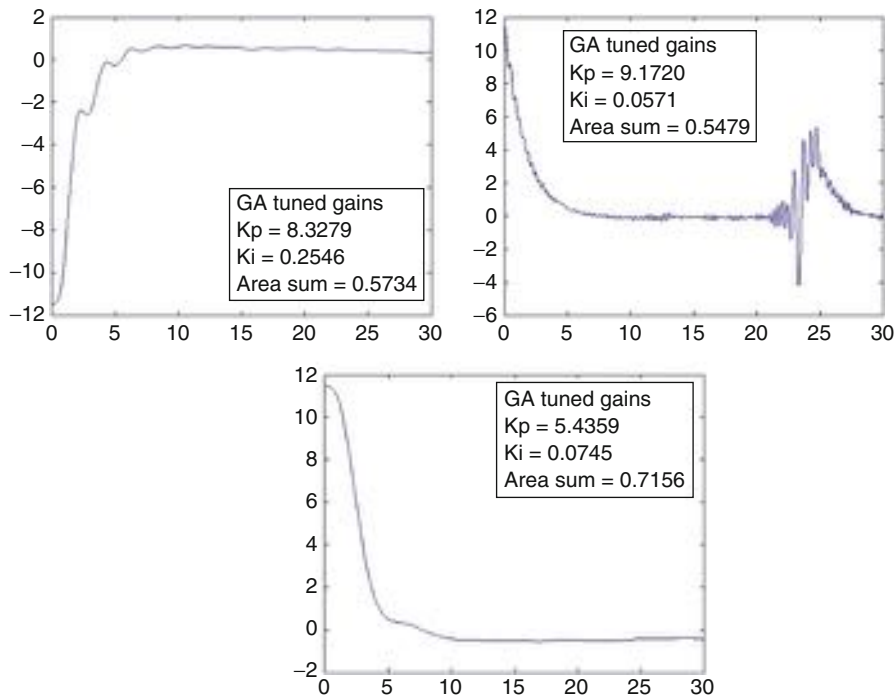


Fig. 57.28 Linear control/hybrid energy control: step response: GA-based PID gain selection

57.13.3 Visualization Environment and Simulation

The indigenous flight simulator/Visualization environment has been shown here:

Table 57.3 Gains from left: manual selection, after GA tuning (6 generations), after 96 generations

| PI con- troller | Kp | Ki | Area sum | PI con- troller | Kp | Ki | Area sum | PI con- troller | Kp | Ki | Area sum |
|--------------------|------|------|-------------|--------------------|--------|--------|-------------|--------------------|--------|--------|----------|
| Pitch | 12.5 | 0.05 | 0.3752 | Pitch | 8.3279 | 0.2546 | 0.5061 | Pitch | 4.3874 | 3.2975 | 1.46E-12 |
| Roll | 14 | 0.02 | 0.3596 | Roll | 9.1720 | 0.0571 | 0.3506 | Roll | 6.3704 | 1.2216 | 1.29E-12 |
| Yaw | 7.3 | 0.05 | 0.5832 | Yaw | 5.4359 | 0.0745 | 0.6380 | Yaw | 2.4128 | 7.1614 | 0.00019 |

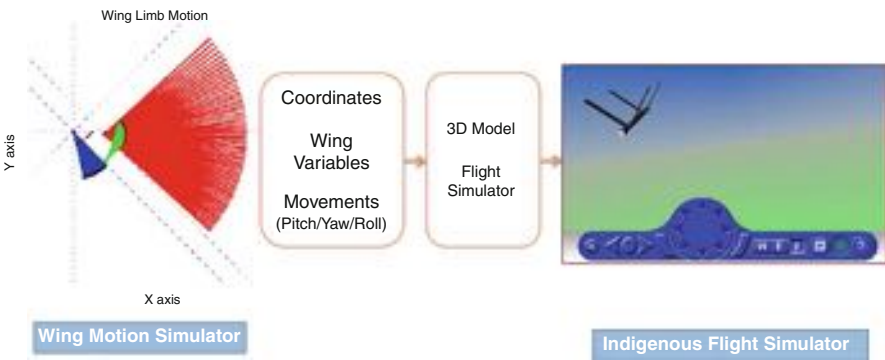


Fig. 57.29 Wing motion simulator and flight simulator

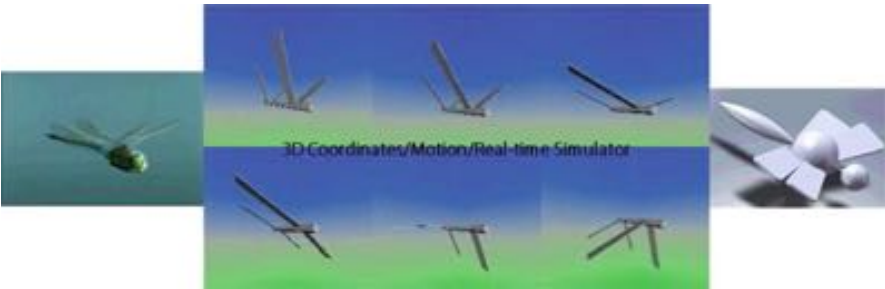


Fig. 57.30 Inspiration to design

57.14 Conclusions

The chapter presented the novel designs and developmental progress made towards the engineering of a high-endurance, micro aerial robot for ISR operations. The advent of a novel flight control and MAV configuration was illustrated in the form of a quad-wing MAV (QV). The discussion was led onto the design and successful development of a new actuation methodology for MAVs based on the dynamics of insect flight. The fixed frequency, variable amplitude (FiFVA) actuation was engineered and used to successfully demonstrate the possibility and effectiveness of replicating insect wing-flapping mechanisms. High-performance, onboard processing capability is essential for any surveillance robot, and thus

the stringent size, weight, and power consumption limitations of micro flyers addressed in the form of the MARC avionics control board. The aerodynamics of the MAV were researched towards the design and development of efficient wings and flapping/feathering schemes. The research aims to improve energy efficiency of MAVs, resulting in higher endurance micro robotic aerial vehicles, with the capacity for both military and civilian applications (Fig. 57.1).

Acknowledgments We thank the Air Force Office of Scientific Research for sponsoring the research and development of the project. We also acknowledge the support of our industrial partner Impact Technologies, LLC. The authors thank the undergraduate student members and research scholars for their contributions: Seong-Joo Kim, Jung-Ho Moon, Emanuel M. Jones, Thomas.

D. Pappas, Andrew Punnoose, Aaron T. May, Maha Hosain, and Joshua A. Sandler. We also acknowledge the facilities and resources provided by Georgia Institute of Technology for the research.

References

- A. Cox, D. Monopoli, D. Cveticanin, M. Goldfarb, E. Garcia, The development of elastodynamic components for piezoelectrically actuated flapping micro-air vehicles. *J. Intell. Mater. Syst. Struct.* **13**(9), 611–615 (2002)
- J. Grasmeyer, M. Keenon, Development of the black widow micro aerial vehicle, in *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, 2001
- D. Jung, J. Ratti, P. Tsiotras, Real-time implementation and validation of a new hierarchical path planning scheme of uavs via hardware in the loop simulation. *J. Intell. Robot. Syst.* **54**(3), 163–181 (2009)
- J. Keshavan, N.M. Wereley, A. Gessow, Design and development of a high frequency biologically-inspired flapping wing mechanism, in *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, 2007
- R. Kornbluh, Project mentor: biologically inspired platform, in *Keynote Presentation at the 8th AIAA/CEAS Aeroacoustics Conference*, June 2002
- R. Michelson, Update on flapping wing micro air vehicle research-ongoing work to develop a flapping wing, crawling entomopter, in *13th Bristol International RPV/UAV Systems Conference Proceedings*, Bristol, 1998, pp. 30.1–30.12
- R. Michelson, D. Helmick, S. Reece, C. Amarena, A reciprocating chemical muscle (rcm) for micro air vehicle ‘entomopter’ flight, in *Proceedings of the Association for Unmanned Vehicle Systems International*, Baltimore, 1997, pp. 429–435
- D.J. Pines, F. Bohorquez, Challenges facing future micro-air-vehicle development. *J. Aircr.* **43**(2), 290–305 (2006)
- T.N. Pornsin sirirak, S.W. Lee, H. Nassef, J. Grasmeyer, Y.C. Tai, C.M. Ho, M. Keenon, Mems wing technology for a battery-powered ornithopter, in *Proceedings of the 13th IEEE Annual International Conference on MEMS*, Miyazaki, 2000
- D. Prosser, T. Basrai, J. Dickert, J. Ratti, A. Crassidis, G. Vachtsevanos, Wing kinematics and aerodynamics of a hovering flapping micro aerial vehicle, in *IEEE Aerospace*, 2011
- D.L. Raney, E.C. Slominski, Mechanization and control concepts for biologically inspired micro air vehicles. *J. Aircr.* **41**(6), 1257–1265 (2004)
- J. Ratti, G. Vachtsevanos, A biologically-inspired micro aerial vehicle: sensing, modeling and control strategies. *J. Intell. Robot. Syst.* **60**(1), 153–178 (2010)
- J. Ratti, G. Vachtsevanos, Towards energy efficiency in micro hovering air vehicles, in *IEEE Aerospace*, 2011
- J. Ratti, R. Goel, S.-J. Kim, J.-H. Moon, T.D. Pappas, G. Vachtsevanos, M. Roemer, Bio-inspired micro air vehicle: design and control issues, in *AIAA Infotech@Aerospace*, 2010

- J. Ratti, E.M. Jones, G. Vachtsevanos, Fixed frequency variable amplitude (fifva) actuation systems for micro aerial vehicles, in *IEEE International Conference on Robotics and Automation@ICRA*, Shanghai, 2011a
- J. Ratti, J.-H. Moon, G. Vachtsevanos, Towards low- power, low-profile avionics architecture and control for micro aerial vehicles, in *IEEE Aerospace*, 2011b
- J.M. Rayner, A new approach to animal flight mechanics. *J. Exp. Biol.* **80**, 17–54 (1979)
- P. Samuel, J. Sirohi, L. Rudd, D. Pines, R. Perel, Design and development of a micro coaxial rotorcraft, in *Proceedings of the AHS Vertical Lift Aircraft Design Conference*, American Helicopter Society, Alexandria, Jan 2000
- G. Vachtsevanos, Hierarchical control, in *Handbook of Fuzzy Computation* (Institute of Physics Publishing, 1998), pp. 42–53
- G. Vachtsevanos, B. Ludington, Unmanned aerial vehicles: challenges and technologies for improved autonomy. *WSEAS Trans. Syst.* **5**, 2164–2171 (2006)

Christian Michelson

Contents

| | | |
|---------|---|------|
| 58.1 | Introduction | 1416 |
| 58.2 | The Major Considerations | 1416 |
| 58.2.1 | Power Considerations | 1417 |
| 58.2.2 | Data and Command Link Considerations | 1417 |
| 58.2.3 | Amplifier Classes | 1418 |
| 58.2.4 | Amplitude Modulation (AM) | 1419 |
| 58.2.5 | Frequency Modulation (FM) | 1422 |
| 58.2.6 | Phase Modulation (PM) | 1422 |
| 58.2.7 | Quadrature Amplitude Modulation (QAM) | 1423 |
| 58.2.8 | Phase-Shift Keying (PSK) | 1424 |
| 58.2.9 | Frequency-Shift Keying (FSK) | 1425 |
| 58.2.10 | Modulation Summary | 1426 |
| 58.2.11 | Multiple Access | 1426 |
| 58.2.12 | Frequency Division Multiple Access (FDMA) | 1427 |
| 58.2.13 | Time Division Multiple Access (TDMA) | 1427 |
| 58.2.14 | Code Division Multiple Access (CDMA) | 1428 |
| 58.3 | Aerodynamic Issues | 1429 |
| 58.3.1 | Antenna Integration and Frequency Issues | 1430 |
| 58.3.2 | Patch Antennas | 1430 |
| 58.3.3 | Whip Monopole (Rubber Ducky) Antennas | 1432 |
| 58.3.4 | Dragged Wire Antennas | 1432 |
| 58.4 | MAV Data Link Power Saving Tricks and Advanced Communications | 1433 |
| 58.4.1 | The Case for Antenna Arrays | 1434 |
| 58.4.2 | Beamforming | 1434 |
| 58.4.3 | Diversity Matters | 1436 |
| 58.4.4 | Additional Beamforming Advantages | 1437 |
| 58.5 | Conclusion | 1437 |
| | References | 1438 |

C. Michelson

Test Engineering Division, Georgia Tech Research Institute, Atlanta, GA, USA

e-mail: christian.michelson@gtri.gatech.edu

Abstract

Micro air vehicles (MAVs) face a unique set of challenges due to their size. Limited payload capacity leads to considerable constraints on power sources, sensors, and communication systems. Power sources are by far the most weight-inefficient components on an MAV. MAV designers are forced to look elsewhere to optimize their designs. The best way to do so in lieu of focusing on improving battery technology is to optimize the systems that draw power, thereby increasing endurance. Motors, onboard processing, and communications transceivers are the largest three power consumers on MAVs today. While motors and embedded processing are important to optimize, the sheer number of available communications options may leave MAV designers unsure how to proceed. By building an MAV around its onboard communications system, designers can increase reliability, endurance, and capability with little or no added cost. Making the right data link and antenna choices at the beginning of the MAV design process greatly eases the constraints later in the project. Specific MAV missions can and should inform the choice of the best communication architecture. Care must be taken to ensure that the end result meets the power, aerodynamic, and electromagnetic requirements for the particular MAV and its particular mission. This chapter seeks to answer many of the communications link questions that MAV designers may have and give a high-level overview of the factors that affect MAV data and control links.

58.1 Introduction

Micro air vehicles (MAVs) face a unique set of challenges due to their size. Limited payload capacity leads to considerable constraints on power sources, sensors, and communication systems. Communications is perhaps one of the largest problems since it has repercussions on so many other onboard systems. If an MAV cannot communicate, then it is of little use as it cannot be teleoperated and it cannot report back data. Many of today's roles for MAVs are indoors where loss due to walls, steel framing, and electromagnetic interference are commonplace. This chapter seeks to explore the various obstacles to MAV communications as well as present several possible solutions to overcoming those obstacles. The majority of this chapter is presented in layman's terms so MAV designers without a heavy RF background can gain the foundational conceptual knowledge to make informed design decisions.

58.2 The Major Considerations

The DARPA definition of an MAV requires that the vehicle be smaller than 15 cm in any dimension (McMichael and Francis 1997). A flying platform with dimensions of merely 15 cm drastically constrains the payload available for designers to use (Zhu et al. 2007). In an average communications system, provisions must be made

for batteries, receivers, transmitters, modulators, digital circuitry, antennas, and interconnecting cabling. Luckily, many of these parts of the transceiver chain are now available in single chip solutions.

58.2.1 Power Considerations

The first major constraint is power. Batteries are typically the heaviest single component (and the least weight efficient) on an MAV. They require a large amount of supporting structure that does not directly provide power. The energy densities available in batteries are an order of magnitude lower than the gasoline used to power the propulsion systems on larger vehicles. For example, the energy densities of common lithium-ion polymer batteries are around 0.5 MJ/kg, whereas gasoline has an energy density of 44.4 MJ/kg (Voelcker 2009). For comparison, a 1-ton car traveling at 100 miles per hour has approximately 1 MJ of kinetic energy (Beidel 2012). A 9-V 1,200 mA-h battery weighs on the order of 35 g. The well-known Black Widow MAV weighs a total of 80 g (National Research Council 2002). This means that the battery accounts for the vast majority of the payload.

Recently, MAVs have started to utilize other power sources such as hydrogen fuel cells. While not technically an MAV, the Hornet, a 6-oz, 15-in. vehicle, successfully demonstrated in 2003 that fuel cells can provide for powered flights of up to 15 min (AeroVironment, Inc. (AV) 2012). Hydrogen and solid-oxide fuel cells provide energy densities much higher than traditional batteries. These technologies will become very viable power sources once miniaturization technology has matured. Many of the recent developments in this field have been driven by cellular phone technology since cellular phones face many of the same constraints that are present in MAVs.

This chapter does not focus heavily on battery or power source technology. Instead, it seeks to present power saving and optimization recommendations for MAV communications subsystems. By optimizing the communications link, designers can increase flight endurance by reducing wasteful power consumption.

58.2.2 Data and Command Link Considerations

Modulation, or the means by which data is encoded on an RF signal, should be tailored to specific MAV roles. Some modulation schemes are designed for spectral efficiency (bandwidth efficiency), some for power efficiency, and others for throughput robustness (Green 2000). There are numerous modulation types, each having unique advantages and disadvantages. The following section describes common analog and digital modulation schemes, amplifier types, and their associated advantages and disadvantages for use in MAVs. Since everything is a trade-off in systems engineering, it is important to know what can and should be sacrificed to meet particular communications requirements. This section should help MAV designers determine which communications trade-offs are worth making.

From a systems engineering perspective, maximizing the efficiency of the MAV's communication links can save power and increase reliability. Higher-efficiency communications can ease the impact of the inherent power constraints since it is comparatively far easier to optimize communication systems than to optimize their power source. Before delving into the particulars of modulation, a primer on amplifier classes is needed in order to appreciate the modulation trade-offs.

58.2.3 Amplifier Classes

Amplifiers are one of the most important components in a transceiver. Radio waves are generally very weak and must be amplified many hundreds or thousands of times to be useful. Amplifiers come in several varieties known as classes. The trade-off between the classes is between output linearity and power consumption. Linearity is a term that describes how accurately an amplifier increases the power level of a signal without distorting the signal. Some might refer to linearity from a qualitative perspective as fidelity (Weisman 2002). This chapter will cover A, B, AB, and C class amplifiers.

For the most accurate and linear amplification of an input signal, a class-A amplifier is desired. Class-A amplifiers are active (or conducting) all of the time. Class-A amplifiers can be thought of as pushing and pulling the signals to reproduce them most accurately. While this is good from a linearity perspective, Class-A amplifiers are very inefficient at converting source power to RF power (Quilter 1993). The power that is not converted to RF is converted to heat. Battery power converted to heat is waste. Class-A amplifiers are required in certain systems where high data rates are required. As a general rule of thumb, class-A amplifiers should be avoided for MAV applications since efficiencies are generally around 25 %.

Class-B amplifiers are conducting approximately half of the time. They can be thought of as only pushing the signals and letting them return to zero on their own. This means that the power required is roughly one-half of a class A, but since the signal is not being pulled to the exact input signal, the output can vary somewhat from the input. This leads to non-linearity and distortion (Quilter 1993). To demodulate signals with high accuracy, linearity is required. If a digital sequence in a signal was supposed to contain 10110 but the values of 1 and 0 were weak, a nonlinear amplifier might skew some of those values to the point that they might get interpreted as 10111. In a digital system, this would be problematic. Such nonlinearity is typically unacceptable in high-bandwidth systems since the error rates would climb to unacceptable levels (Green 2000). Class-B amplifiers have efficiencies on the order of 75 %.

Class-AB amplifiers are a compromise between A and B amplifiers. They operate more than 50 % of the time but less than 100 % of the time. This results in acceptable linearity with lower power requirements than a pure class-A amplifier. These amplifiers can be biased toward class-A operation or class-B operation, thereby allowing the designer to tune where they want the power to linearity trade-off to fall.

These amplifiers are popular because they afford designers a degree of flexibility. Class-AB amplifiers can achieve efficiencies between 50 and 70 % (Quilter 1993).

Class-C amplifiers are much more efficient than A or B since they operate less than 50 % of the time. This results in efficiencies exceeding 80 % (Quilter 1993). Class-C amplifiers are typically not used for advanced modulation since they have low-output linearity (Chi and Das 2008). They can however be used for certain low-bandwidth applications where linearity is not required. This type of amplifier can be used for continuous wave (CW) (such as Morse code transmission) and frequency-shifted modulation schemes where amplitude is not terribly important. These modulation schemes will be discussed later in this chapter.

The input and output waveforms for each of these amplifier classes are shown below for comparison (Fig. 58.1).

When viewed graphically, the effects of only conducting part of the time become apparent. Every section of the input signal that is not reproduced accurately results in distortion of the signal and an increased occurrence of demodulation errors.

For most MAV applications, the bandwidth requirements will be low to medium since the power requirements of high-data-rate sensors are typically too great for MAV platforms. Telemetry, sensor data, and command and control are common MAV data links. Class-AB amplifiers are typically acceptable for use on MAVs with such data links since the data rates are comparatively low. If telemetry and low-speed sensor traffic are the only things sent across the data link, then a class-B amplifier may even suffice. MAV designers must determine the sensor data rates and ascertain how much error is acceptable. Switching from a class-AB to a class-B amplifier could provide several minutes of increased mission endurance. A high-level summary of the amplifier classes is shown in Table 58.1.

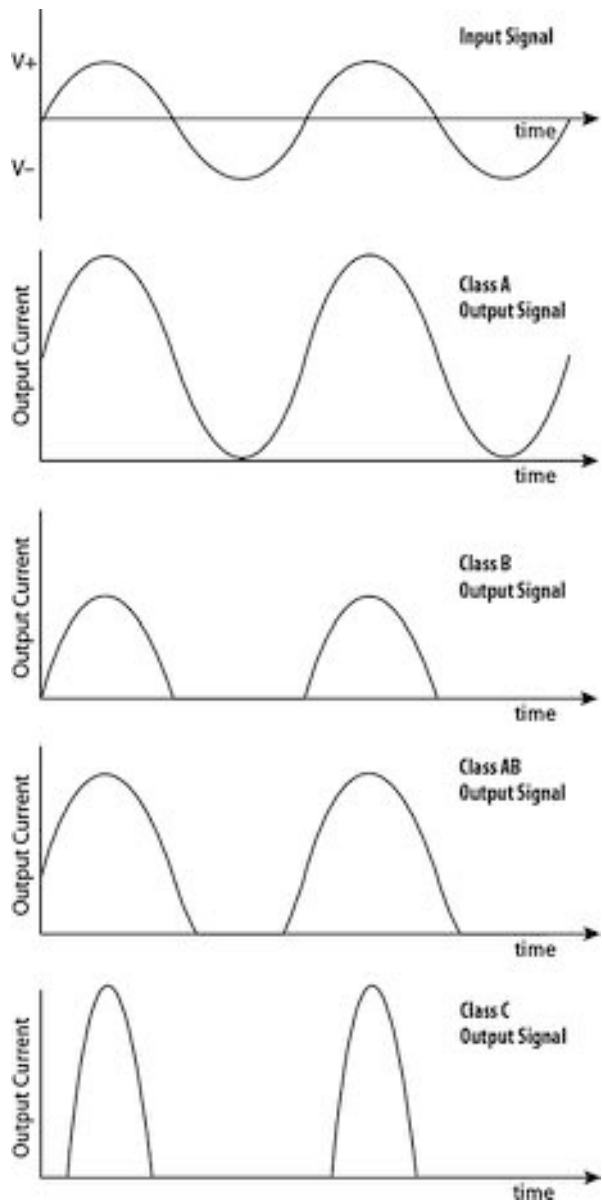
Modulation schemes are the next key element that must be understood in order to make the best decision for an MAV communications architecture. In order to understand the underlying theory behind the most popular digital modulation schemes, a primer on three common analog modulations is required. The waveforms for the three basic types of modulation are shown below (Fig. 58.2).

58.2.4 Amplitude Modulation (AM)

Amplitude modulation (AM) is an analog modulation scheme that has been used for a long time. AM modulation increases and decreases the envelope of the RF carrier to convey information. The term envelope refers to the waveform's shape that is formed by the changing amplitude of the faster moving carrier. Figure 58.2 shows an AM-modulated waveform.

The "value" of the modulated data is determined by the height of the envelope at any point in time. The faster moving waveform inside is the carrier. The modulated data rides on top of the carrier. An AM receiver simply extracts the envelope, so the value of the signal can be determined. The advantage to this modulation scheme is that it is very simple to demodulate. An AM demodulator can be implemented

Fig. 58.1 Comparison of amplifier waveforms

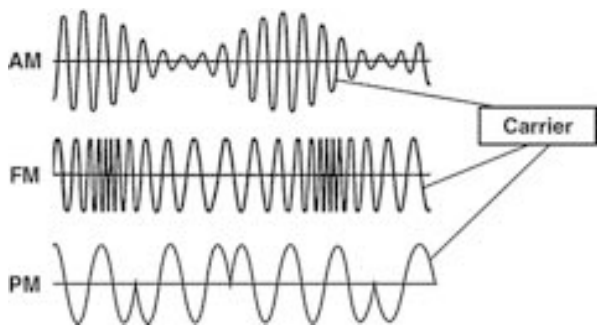


with very little hardware and requires very little power (Carr 1997). The downside to this modulation scheme for use in MAVs is that it is relatively bandwidth limited. Bandwidth is a term that describes how much data can be sent per unit time. If the MAV is required to send back data at a medium or high rate, AM is definitely not the best candidate.

Table 58.1 Pros and cons of various amplifier classes

| Amplifier class | Pros | Cons |
|-----------------|-----------------------------|--------------------------|
| A | Highly linear | Highly power inefficient |
| AB | Reasonable linearity | Low to medium efficiency |
| B | Reasonable power efficiency | Low to medium linearity |
| C | Very power efficient | Low linearity |

Fig. 58.2 Comparison of modulation waveforms



The other primary disadvantage is that the only means of overcoming noise in an AM system is to increase the transmit power. Increasing transmit power drains the batteries at a faster rate. Noise is present in all communications systems. Noise comes from numerous natural and man-made sources. Anything that contains rapidly changing electrical currents generates RF noise. The sun, auroras, car starters, and everything with a temperature above absolute zero all create a degree of RF noise that can make receiving signals difficult. Using filtering or placing receiver electronics in a Faraday cage can reduce noise levels. Faraday cages act as an RF shield, so spurious and unwanted signals stay out of the receiver. There are other modulation schemes that are significantly less sensitive to noise than AM.

One of the main problems with AM modulation is that AM transmitters require high linearity since the value of the data is directly proportional to the instantaneous power level. If the amplifiers were nonlinear, then the true value of the data would have to be scaled by some correction factor to be interpreted correctly. Linear amplifiers are more complex and require more power than nonlinear amplifiers as was previously discussed.

The last, and perhaps most problematic, issue is spectral efficiency. AM wastes a large percentage of the transmitted energy with redundant and unused power. AM modulation creates a carrier and two main sidebands. Each sideband contains a duplicate version of the data, so only one of the sidebands is actually used by most AM receivers. The extra sideband conveys no extra information yet the power in this redundant sideband must be sourced by the battery (Carr 1997). For these reasons, AM is not the most efficient modulation scheme for MAVs to use.

58.2.5 Frequency Modulation (FM)

Frequency modulation (FM) offers many advantages over AM. FM encodes information in frequency offsets from the main carrier frequency (Carr 1997). In an FM system, the shift off of the main carrier frequency (positive or negative) represents a value. Notice in the FM waveform that the output power envelope is constant over time and only the frequency changes (Weisman 2002).

The main advantages of FM are noise resilience, signal strength resilience, linearity invariance, and higher efficiency. FM is relatively impervious to noise because noise superimposes itself on top of the desired signal, thereby randomly affecting the amplitude. Since FM receivers are not demodulating based on amplitude but on frequency, the noise has little effect on the demodulated signal. This extends itself to allow FM signals to be somewhat signal strength invariant as well. Within reason, a frequency shift in a signal with a small amplitude can be demodulated just as easily as a signal with a large amplitude. By comparison, a weak AM signal will become increasingly inundated by noise and may not be discernible.

Linearity does not affect FM signals like it does AM since linearity directly influences amplitude and not frequency. This means that class-B or class-C amplifiers could be used without a significant impact on the reliability of an FM link (Weisman 2002).

FM signals have a much larger bandwidth than AM signals. This allows them to carry more data per unit time. Spectrally, FM signals are much more efficient than AM as well since the data in all of the sidebands is used. This means that there is little wasted energy. An FM signal's sidebands increase in amplitude and number as the modulated signal increases in frequency deviation from the carrier (this is known as a modulation index).

There are two main downsides to FM modulation. The first is that it requires a significantly more complex modulator and demodulator than AM. This not only increases cost but typically results in a larger demodulation circuit. Another negative aspect of FM is that the sidebands extend out infinitely far in frequency (theoretically), so a wide unused RF range must be allocated to keep FM signals from interfering with neighboring signals. To constrain the transmitted frequencies to a particular range, additional filters must be added. This filtering can distort the signal, but this is typically not a large problem. FM modulation is perfectly conducive to use in MAV applications. It is appropriate to use FM for control and telemetry links. Until the advent of 2.4 GHz radios, hobbyist radio control transmitters typically used FM. The only reason that it is not recommended for MAV data links is that there are multiple digital modulation techniques that have much higher efficiency.

58.2.6 Phase Modulation (PM)

Phase modulation sends information via the instantaneous phase of the carrier signal (Carr 1997). This modulation scheme is not terribly common for analog data links because it has some inherent weaknesses. The first weakness is that there is not

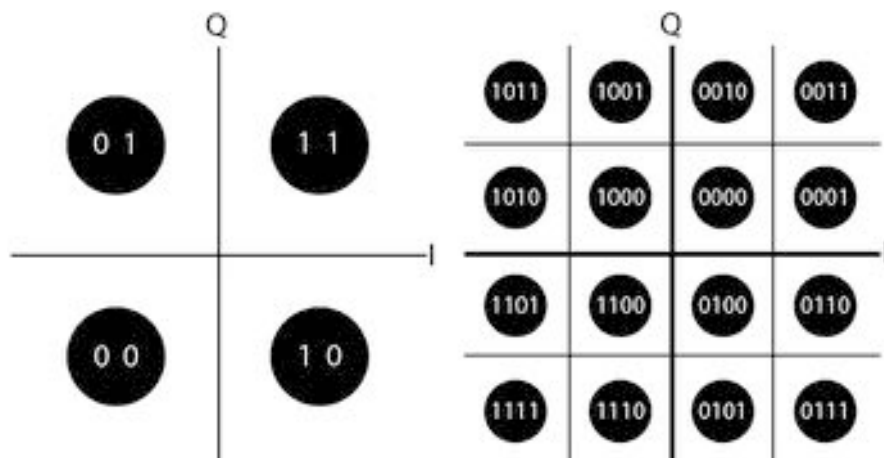


Fig. 58.3 4-QAM and 16-QAM constellations

a good way to determine whether a signal is at -180° or $+180^\circ$ since they look identical. The modulation hardware is also more complex than FM. Spectrally, PM is not terribly efficient either as it doubles the required baseband bandwidth. PM is used far more often in its digital form. An example of a digital PM signal is shown below (Fig. 58.3). Notice the phase transitions. In its digital form, PM is known as PSK or phase-shift keying (which will be explained later).

This modulation scheme is used as a component of many advanced digital modulation schemes. Its usefulness as a stand-alone modulation scheme is questionable for MAV applications.

58.2.7 Quadrature Amplitude Modulation (QAM)

QAM is the one of the most common modulation schemes used for high-bandwidth digital wireless data links. IEEE 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), and numerous cellular standards (HSPA+) are based on QAM. QAM attempts to pack more data per unit time into the signal. This is accomplished by adding additional carriers. By amplitude modulating these two out of phase carriers (one carrier called I and the other “quadrature” carrier 90° out of phase called Q), two independent values can be sent at once. In order to determine what value the combination has, one carrier is assigned to an I value and one is assigned to a Q value on a Cartesian coordinate grid called a constellation (Weisman 2002). Each unique I, Q coordinate can be associated with a “symbol” value on the plane. The symbol mapping is agreed upon ahead of time by the transmitter and receiver, so proper decoding can take place.

Constellations can be implemented with varying degrees of granularity. If only positive and negative values are sent for I and Q, then there are four possible combinations. This would mean that one of only four possible values could be sent

at once. In order to increase the data rate, additional possible values can be added by breaking those quadrants up into smaller quadrants. The constellations for 4-QAM (4 regions or a 2×2 constellation) and 16-QAM (16 regions or a 4×4 constellation) are shown below (Fig. 58.3).

Note that 16-QAM can send 4 bits in each symbol whereas 4-QAM can only send 2 per symbol. That means that the theoretical maximum data rate of 16-QAM is twice that of 4-QAM. In reality, 16-QAM, 64-QAM, and 256-QAM are the most used for wireless data links. As the density of the constellation increases, the required demodulation accuracy increases and the amount of acceptable error decreases. If a value is not demodulated correctly due to noise, then the wrong constellation value will be read resulting in data corruption. Notice how much error is allowed in 4-QAM before a boundary is crossed compared to 16-QAM. For this reason, QAM density cannot be extended indefinitely. Only in high-precision optical links can extremely high-density constellations such as 4096-QAM be used (64×64 constellation) (Weisman 2002).

The complexity of QAM transceivers is much higher as well. The additional out-of-phase demodulator requires additional mixers, filters, and amplifiers, which increase cost and device footprint size. The most significant problem, however, is that QAM is an advanced form of AM modulation. As such, it requires that the amplifiers be linear. This means that QAM transceivers are power inefficient.

Noise susceptibility is a large problem. Noise causes improper constellation decoding, so if the MAV is expected to communicate in an urban environment with a high noise floor, range will be limited. The large number of noise sources present indoors may also be a problem for MAVs with indoor missions. QAM is therefore not recommended for command and control data links where high reliability is required. Nonessential, error-tolerant high-data-rate sensor traffic such as video are more suited to QAM modulation.

58.2.8 Phase-Shift Keying (PSK)

PSK uses the same constellation principle as QAM. The difference is that instead of modulating the main and quadrature carriers' amplitudes, it instead modulates their phase. Each unique phase combination is mapped to a set of symbols (Weisman 2002). Instead of implementing the constellation in rectangular I and Q terms as in QAM, PSK instead implements the constellation on a circle. Constellation density increases as points are added to the circle. A standard quadrature PSK (QPSK) and a higher-density 8-PSK constellation are shown below for comparison (Fig. 58.4). Although viewed from a different perspective, a QPSK constellation is technically equivalent to the 4-QAM constellation, and the signals are indistinguishable.

PSK is less susceptible to noise than QAM since noise directly affects amplitude and not phase. For this reason, PSK is more suited to noisy urban environments. PSK is also bandwidth efficient. The negative aspect to PSK is that phase is relative,

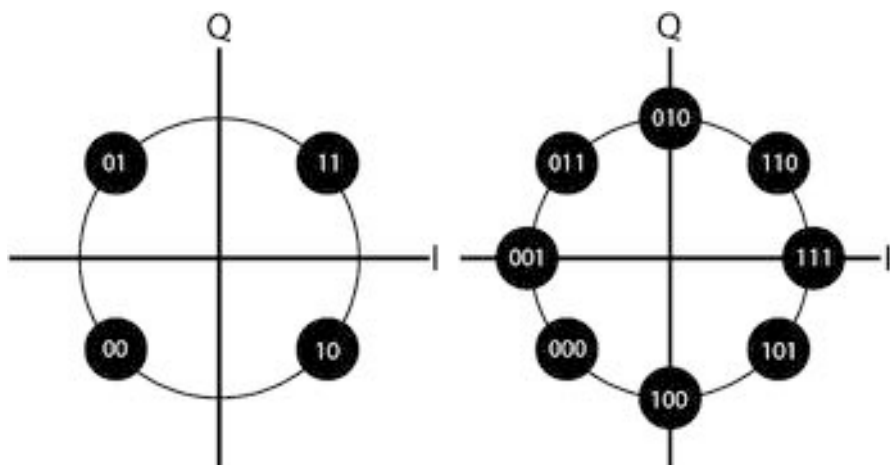


Fig. 58.4 QPSK and 8-PSK constellations

so the receiver must be phase and frequency locked with the transmitter in order to determine the phase offsets.

Binary phase-shift keying (BPSK) is a variant of PSK that can only encode a 1 or 0. The constellation contains only two points. A sine wave is interpreted as a 0, and an inverted sine wave is interpreted as a 1. This is not spectrally efficient but is very robust as a large amount of error can be tolerated before data is corrupted. For long range high-reliability data links with relatively low data rates (such as control links), BPSK is optimal.

58.2.9 Frequency-Shift Keying (FSK)

Frequency-shift-keying is the digital form of FM modulation where the frequency shift is discretized. The simplest form is called binary FSK (BFSK) where frequencies lower than the carrier are interpreted as zeroes and frequencies higher than the carrier are interpreted as ones (Weisman 2002). This concept can be extended by assigning different symbol values to the various frequency offsets. This can be thought of as a linear constellation in one dimension.

FSK, like FM, requires a greater bandwidth since the carrier can move up and down in frequency. FSK has two major advantages over PSK. First, there is no need in FSK to phase and frequency lock with the transmitter. Second, since FSK is a form of FM, it does not require highly linear amplifiers. For this reason, if an MAV design does not have much overhead left on the power budget, FSK would be better than PSK or QAM. The downside to FSK is that at a given power level, the error rate is higher than PSK.

Table 58.2 Summary of modulation schemes

| Modulation | Environment | Power required | Data rate | Link reliability | Bandwidth |
|------------|---|----------------|-------------|------------------|-------------|
| AM | Low noise, rural, long distance | High | Low | Low | Low |
| FM | High noise, urban or rural | Low | Medium | High | Medium |
| PM | High noise, urban or rural | Low | Medium | Medium | Medium |
| QAM | Low noise, outdoor, rural | High | Very high | Low | Medium-high |
| PSK | High noise, urban or rural, indoor or outdoor | Medium | Medium-high | Medium-high | High |
| FSK | High noise, urban or rural, indoor or outdoor | Low | High | Medium-high | High |

58.2.10 Modulation Summary

MAV designers must prioritize their needs based on MAV roles. Power, robustness, and data rates are all constraints at odds with one another. For maximum range and minimum power consumption, FSK is likely the best choice. A high-power amplifier is required for long-distance data links, so it is best to design for FSK modulation that can use amplifiers that are efficient at the expense of linearity. For high-reliability operation in noisy environments, BPSK is best suited since it can tolerate a large degree of noise and nonlinearity. For the highest data rates, a high-order QAM modulation is best. Multiple data links may be used concurrently for the various onboard systems so long as care is taken to avoid self-interference. If multiple data links are used, designers should ensure that the transceivers support a sleep mode to reduce the power requirements. There are other modulation types, but many of them are variations or combinations of the six mentioned here. Table 58.2 sums up the features of the six modulation schemes discussed in this chapter.

58.2.11 Multiple Access

Modulation alone does not solve all MAV communication issues since there is no provision for communicating with multiple MAVs concurrently. Swarm operations are a very real possibility in the future, so having a multiple access architecture in place is advisable. Communicating with multiple receivers concurrently is known as multiplexing or “multiple access.” Of course, the trade-off here is that this capability requires yet more transceiver complexity and reduces the error tolerance of the data link.

There are three primary methods for communicating with multiple recipients: time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA). Do not get confused; these are not modulation schemes. These operate in a layer above modulation that requires additional processing and therefore additional power consumption. These multiple access schemes are required if multiple MAVs must communicate in the same frequency space at the same time.

58.2.12 Frequency Division Multiple Access (FDMA)

The first multiple access scheme breaks up the users into frequency slots. This is called frequency division multiple access (FDMA). Each user is assigned a particular frequency (called a channel) by a controller. Each user then can use that one frequency as long as is needed. This would be like people talking at the same time but in different pitches, so the individual speakers could be distinguished concurrently.

The controller knows the frequency allocations and can quickly switch to the various frequencies to communicate with multiple MAVs. Each user can transmit and receive constantly since each has a dedicated channel. This allows for data rates that are equivalent to non-multiple access scenarios. FDMA requires a much larger available bandwidth since whole channels must be allocated to each user. The downside to this is that users that are not actively utilizing their channel are still assigned a channel. This leads to wasted time and bandwidth. If a limited bandwidth is available (as is almost always the case), only so many channels can be carved out before transmitters start leaking over into adjacent channels. For this reason, FDMA cannot support large numbers of users. If a large MAV swarm must communicate, pure FDMA is not the best choice. FDMA is not a good choice for mesh communications either. If each MAV must communicate with other MAVs, FDMA requires that each MAV either switch frequencies or send information via a controller acting as an intermediate relay (Green 2000).

58.2.13 Time Division Multiple Access (TDMA)

The second method is similar to a group of people talking in a round-robin fashion where one person talks and then the next person talks and so on until all of the people have talked. At this point, the first person can again talk. This is known as time division multiple access (TDMA). The difference between people talking and TDMA is that the times during which each person is allowed to talk are assigned. Time slots are assigned to and reserved for each transmitter, so two are never transmitting at the same time. This allows a scheduled rotation to take place, so multiple MAVs can share the same frequency. Each complete rotation is called a frame. Each frame is broken into time slots assigned to each user. A diagram of a TDMA rotation for four users is shown below (Fig. 58.5).

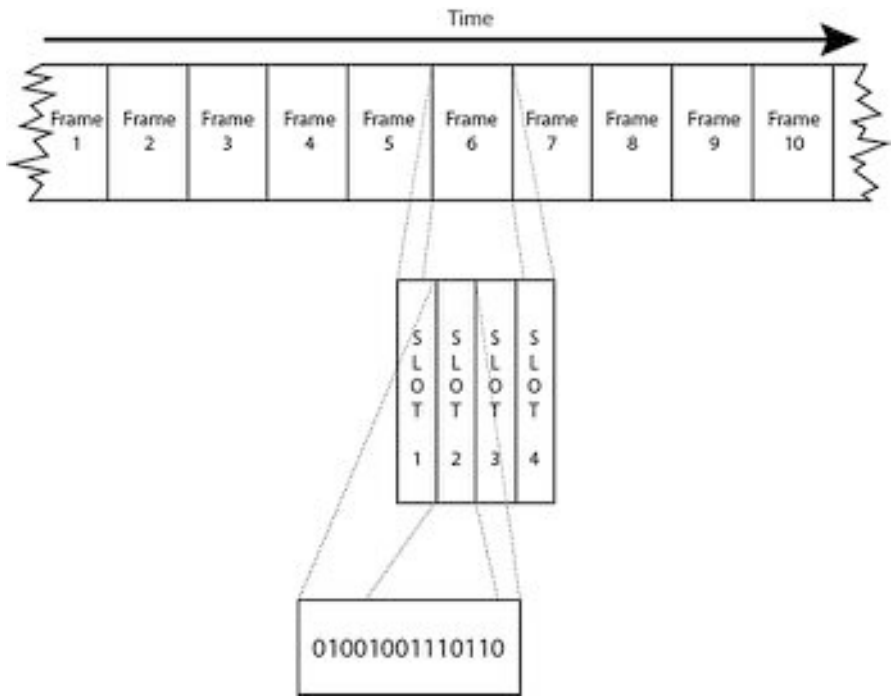


Fig. 58.5 Visualization of time-domain multiple access

This access scheme requires there to be a controller that is in charge of coordinating and assigning the time slots. As additional transceivers are added, the controller must re-allocate the frames and inform all of the transceivers of the new rotation parameters. The downside to this scheme is that the data throughput for any one user drops as more users are added. Furthermore, all users must be time synchronized, so every user knows when their frame slot starts and ends. This requires a more complex controller. For large MAV swarms where high data rates are required, TDMA may not be the best choice. For small swarms (10–20 MAVs), TDMA is a reasonable choice. An advantage TDMA has over FDMA is that MAVs can communicate with one another since each MAV is listening and transmitting on the same frequency.

FDMA and TDMA can coexist to gain the best of both worlds and increase the number of supported users by using TDMA on several FDMA channels. This primarily serves to increase the maximum possible number of concurrent users.

58.2.14 Code Division Multiple Access (CDMA)

The latest multiple access scheme to see widespread use is called code division multiple access (CDMA). CDMA is equivalent to a group of people speaking in multiple languages. Anyone speaking a different language is ignored just like all

of the other environmental noise. CDMA assigns each user a unique code that can be used to modulate that user's data. The receiver then tries to correlate all of the received signals with the unique code to interpret the data. A CDMA receiver can pull the coded data out of noise since noise is random and the individual signals are not random. The nonrandom nature of the desired signal allows it to be pulled out of the noise by a digital receiver that knows the characteristics of the desired signal.

CDMA is a spread-spectrum multiple access scheme where each user's bandwidth is spread evenly across the entirety of the used bandwidth. This scheme requires a more complex and power-hungry transceiver than FDMA or TDMA since relatively complex mathematical processing must be done in real time to correlate the various codes with the signals.

If MAVs must operate with a very limited bandwidth, CDMA is not a good choice since it requires much more bandwidth than FDMA or TDMA. This, however, can be good and bad. Since the signal is spread over a large frequency range, it is more resistant to interference from narrowband noise sources. CDMA is much better in noisy urban environments than FDMA or TDMA. If a noise source is present at or near the frequency used for FDMA or TDMA, the signal can quickly become difficult to demodulate whereas a CDMA data link would still be able to operate (Weisman 2002).

Another advantage of CDMA is that it can support a large number of users at the same time on the same frequency. Since a unique code is used for each receiver, neighboring transmitters can transmit at the same time and still be received without difficulty. For this reason, CDMA is an excellent choice for swarm MAV applications.

If the MAV application requires a solid jam-resistant data link, then CDMA is the best multiple access scheme. While TDMA signals could be confusing to interpret, jamming them is trivial since all of the signals are at one frequency. FDMA is slightly more difficult to jam since multiple frequencies are involved; however, the frequencies are all relatively close to one another. A high-power jammer would have enough bleed over into adjacent channels that an FDMA system could be jammed quite easily (Chi and Das 2008).

CDMA is hard to jam because the energy is spread across a large frequency range. Jammers typically operate at a fixed frequency at any given time. A traditional jammer would only affect a small portion of the CDMA bandwidth since there is only so much power available with which to jam. To a CDMA receiver, a jammer would merely raise the noise floor several dB. This is not terribly problematic for CDMA since CDMA can operate below the noise floor anyway (Kashyap et al. 2004).

58.3 Aerodynamic Issues

The second large constraint is not based on power but on aerodynamics. The lifting capability of a 15 cm airfoil is relatively low. The addition of communications systems is not only heavy but may adversely affect the aerodynamic properties

of the platform. To overcome this, designers must integrate the communications systems into the airframe as much as possible while utilizing the lightest available materials.

58.3.1 Antenna Integration and Frequency Issues

Data link frequency is a major concern. The tradeoff with respect to frequencies is threefold. Low-frequency signals are much more able to penetrate walls and travel large distances than are high-frequency signals at the same power. On the other hand, low-frequency signals inherently have less available bandwidth (Green 2000). If high-data-rate digital traffic must be relayed, then low frequencies will not be able to support the necessary bandwidth. This is a key constraint, but MAVs are more constrained by size than by bandwidth.

Low-frequency signals are received and transmitted much better by antennas that are on the same order of magnitude as the wavelength of the signal (Green 2000). Wavelength is the physical distance required for a signal to go through one cycle. Low frequencies have high wavelengths, and high frequencies have low wavelengths. With the limited space available on an MAV, designers are forced to move toward higher-frequency communications simply to accommodate the antennas. Large antennas weigh more and have more aerodynamic drag.

There are three antenna classes that are particularly conducive to use on MAV platforms. The first is the patch antenna, the second is the whip monopole style antenna, and the third is the dragged wire-style antenna.

This section will describe the advantages of each antenna type and will demonstrate how best to integrate them with MAV platforms from an aerodynamic and RF standpoint.

58.3.2 Patch Antennas

Patch antennas use thin microstrips laid down on a flat surface that is separated from a “patch” of metal that acts as a ground plane. Ground planes effectively act as an RF mirror to extend a short antenna to twice its electrical length (Weisman 2002). A well-designed ground plane can greatly increase the efficiency of an antenna. The microstrips are designed to be $\frac{1}{2}$ wavelength (or $\frac{1}{4}$ in some cases). An example of a simple dual patch antenna is shown below (Fig. 58.6).

Gain is a term referring to the degree that an antenna concentrates its energy in a particular direction. Do not think of gain as energy that an antenna creates. Antennas cannot create additional power since they are passive devices. Gain is a measure of how much more power a receiver in the direction of the gain lobe would see compared to an antenna that radiated perfectly evenly in all directions (the so-called “isotropic radiator”). This quantity is measured in the logarithmic unit of decibels relative to an isotropic radiator, or dBi. The term “lobe” refers to the direction of the

Fig. 58.6 Example of a patch antenna

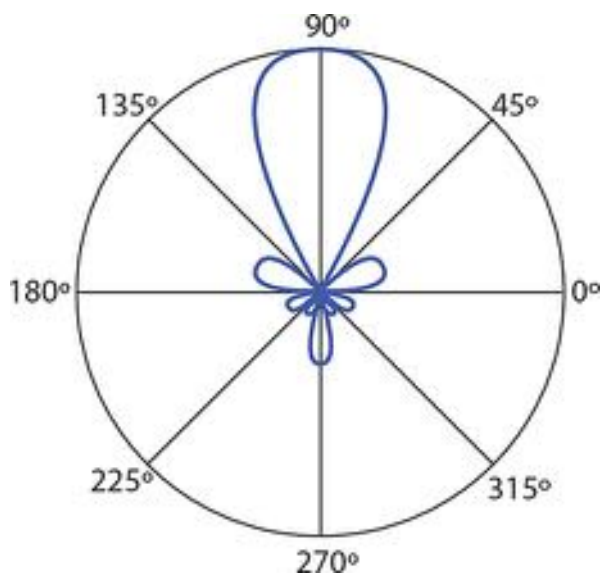
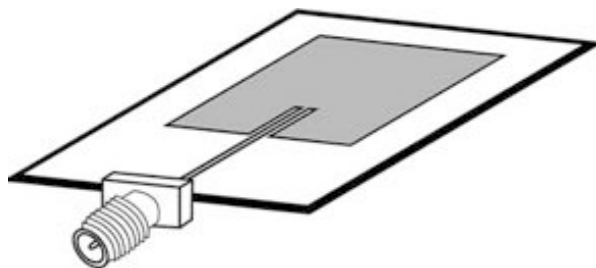


Fig. 58.7 Example of a patch antenna radiation pattern

gain (Carr 1997; Schwarz 1990; Green 2000). A diagram of an example radiation pattern for a patch antenna with a large gain (main lobe) in the 90° direction is shown above (Fig. 58.7).

Gain is good and bad depending on the design application. If the design application requires that there be omnidirectional (e.g., all horizontal directions) coverage, then a low-gain antenna is desired. If the data link is fixed in a particular direction, then it is good to have very high gain in that direction. The higher the gain, the more narrow the main lobe. The smaller secondary lobes are always present and result from nonideal characteristics that cannot be totally overcome in physical antennas (Durney and Johnson 1969).

The gain of a patch antenna can be tuned by an RF designer to meet the particular design specifications of MAV designers. Patch antennas are small and easy to integrate into the airfoils of wings or vertical stabilizers. For this reason, they are particularly conducive to use on an MAV from an aerodynamic perspective. Patch antennas can easily be integrated into wing surfaces or vertical stabilizers.

Fig. 58.8 Example of a whip monopole antenna



58.3.3 Whip Monopole (Rubber Ducky) Antennas

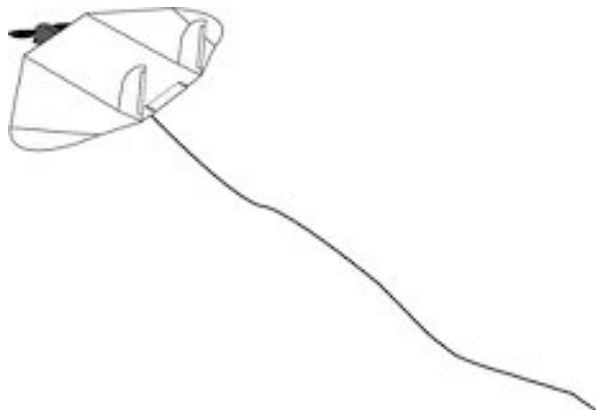
Whip antennas are common in consumer electronics. The external antennas on wireless routers are examples of whip antennas. Older cell phones often used whip antennas. An examples of a removable whip monopole antenna is shown below (Fig. 58.8).

This antenna style has a relatively low gain, so it is useful for omnidirectional applications. The main reason that this antenna type is covered in this chapter is that they are what many people think about when they start to research antennas. As such, MAV designers may flock to them as an easy solution. This is not advisable for MAV designs. The main reason is that whip monopoles are aerodynamically troublesome. Dragging a vertical pole through the air creates turbulence in the form of shed vortices that detrimentally affect a MAV's drag characteristics. Unless the antenna can be integrated into the vertical stabilizers in such a way that laminar flow is maintained on the downwind side of the antenna, avoid their use.

58.3.4 Dragged Wire Antennas

Dragged wire antennas have been used for a long time for both MAVs and larger platforms. This type of antenna has the advantage that it can be cut to any length to match the wavelength of the desired frequency (Schwarz 1990). Adding or removing wire length has little effect on the aerodynamic drag of the MAV (Fig. 58.9).

Fig. 58.9 Example of a dragged wire antenna



Low-frequency patch and monopole antennas become too large to integrate into the vehicle. If the MAV application requires operation at significant distance, then the dragged antenna is the best choice. As was stated before, low frequencies are better for long-range communications. Aerodynamically, the dragged wire antenna is not as favorable as an integrated patch antenna and also introduces additional risks to the platform such as snagging on objects or becoming tangled in propulsion systems (especially on rotary wing platforms). Another consideration is that these antennas will be roughly horizontally polarized when in forward flight, yet they will be vertically polarized when hovering. Polarization refers to the spatial orientation of the electromagnetic fields emitted by and best received by an antenna. Having matching polarization on both the transmitting and receiving antenna is desirable, so designers need to take this into consideration when designing the MAV as well as the ground station antennas (Weisman 2002). Vertical antennas will tend to stay mostly vertical in flight, whereas horizontal antennas change their relative orientation as the platform turns.

58.4 MAV Data Link Power Saving Tricks and Advanced Communications

In lieu of developing lighter and more efficient batteries, reducing the power consumption of onboard systems is the logical path forward. One of the two power-hungry systems onboard is the data link transceiver (the other being the electric motor). Lower power consumption results in increased endurance and range.

The implementation of this trade-off seems initially paradoxical. If transmitter power is reduced, one might conclude that the data link's range will be reduced. This is not necessarily the case. By employing some interesting technologies, MAVs can reduce transmitter power while increasing range.

58.4.1 The Case for Antenna Arrays

Omnidirectional antennas emit energy in all horizontal directions relatively equally. Omnidirectional antennas are easy to mount since they do not require aiming, but they have very low gain. The energy radiated from an omnidirectional antenna is not utilized very efficiently (Weisman 2002). Consider an antenna radiating in a 360° horizontal pattern. The receiving antenna is only in one direction so all of the energy radiated in other directions is wasted. This seems like a waste of precious limited battery power. The obvious initial solution is to use directional antennas with high gains. Unfortunately, it is not that simple to implement a traditional directional antenna on an MAV. Yagi and dish antennas are far too large and would require a servo-driven steering apparatus, which is not feasible on an MAV. Regular high-gain patch antennas are lighter and smaller and can achieve gains of 8–12 dBi, but they still would require servo-driven steering. One possible solution is employing antenna arrays which are a number of antennas working together (Carr 1997).

58.4.2 Beamforming

Phased array transmission was discovered in 1905 by Nobel Laureate Karl Ferdinand Braun. He was able to electronically steer radio waves using phase-shifted signals and multiple antennas. This concept grew into the large phased array radars that are used today. On a smaller scale, phased arrays have made their way into consumer devices such as 802.11n wireless routers and in some 4G cell phones.

The corollary that better fits application to MAVs is the wireless router. Modern 802.11n routers utilize a technology called Multiple-In, Multiple-Out, or MIMO. MIMO uses multiple receive and transmit antennas to provide some unique and useful characteristics that could be of particular use to MAVs.

Beamforming generally refers to the adjustment of the power and phase of the signals at each transmitting antenna (or element). The resulting constructive and destructive interference can null the signal in certain directions and increase the signal in others (effectively creating gain lobes and nulls). The lobes created by a beamforming technique effectively increase the signal gain in a direction, thereby increasing the received signal strength at the receiver (Schwarz 1990). This is good for MAVs since it allows for very fast beam steering without any moving parts. Lower power transmitters at higher frequencies can achieve beamforming gains that overcome the otherwise troublesome problems associated with transmitting a signal in free space (Sung 2004; Gollakota et al. 2009).

One of the most important things MIMO can do is to counteract free-space path loss by using MIMO precoding to “steer” a beam toward a receiver by using onboard intelligence instead of raw power (Biglieri et al. 2007). This is done using digital signal processing, the details of which will not be discussed here. Beamforming and precoding together can direct a data link directly at an MAV without interfering with the neighboring MAVs, thereby effectively giving a dedicated RF channel to each MAV (Bakr 2010).

There are a few caveats to this however. Beam steering requires some knowledge about the direction of the receiver (Sung 2004). If the MAV is launched from a known location, periodic location updates would be required to keep the beam aimed at the MAV. This is not an insurmountable problem due to their relatively slow speed. If the initial location is not known, the MAV will have to report its position back using a beaconing technique. An antenna array can be temporarily omnidirectional if properly driven, thereby allowing for occasional beaconing signals. Once the beam is aimed at the MAV, periodic updates can keep it aimed for the remainder of the flight. This same idea works in reverse and allows the MAV to aim its beam at the ground station.

This concept can be expanded to allow for communication with multiple MAVs using one ground station. By utilizing TDMA on top of precoding, time slots are assigned to each MAV for data transfer. During each time slot, the beam is steered toward a particular MAV. This does however reduce the data rate to any one MAV. If location information is shared in an MAV swarm, MAVs can even send communications between one another by aiming their beams at each other during specific TDMA slots. The key factors that must be maintained by each MAV and the ground station are relative location and time.

When beamforming is utilized, the antenna pattern is always symmetrical. This means that at best 50 % of the emitted energy is headed toward the intended receiver. While not optimal, this is much better than the fraction of a percent of the energy heading toward the receiver as was the case with the omnidirectional antennas.

As an example, consider a system using the 5.8-GHz band. If small $1/2$ wavelength patch antenna arrays were used, each would only need to be 1" long. Theoretically, MIMO antennas should be spaced $1/2$ wavelength apart for optimal gain. When $1/2$ wavelength spacing is used, the antennas are said to be critically spaced, resulting in only one pair of main lobes being emitted. At other spacings, additional wasteful side lobes form in other directions. A 4-antenna array (4×4 MIMO, meaning 4 receiving and 4 transmitting antennas) would only need 4 in. in any dimension to mount a 5.8-GHz antenna array (Jiang 2004). Many MAVs can accommodate this as the definition of an MAV is less than 6 in. in any dimension. Black Widow, for example, could accommodate a 4-antenna patch array with very little weight or aerodynamic drag. If an airframe cannot accommodate a 4-antenna array, a 2-antenna array could be used (with the obvious caveat that your beamforming gain will drop significantly). The larger the array of antennas, the narrower the beam that can be achieved (Gollakota et al. 2009).

4×4 MIMO systems have been commercially shown to have +12 to +25dB beamforming gains (<http://www.quantenna.com/beamforming.html>). These types of gains are just not possible using conventional steerable small antennas that would fit on a MAV. Extending the previous 5.8 GHz example, assume a standard 250-mW (24-dBm) data link using a single $1/4$ wave antenna with 2 dBi of gain is used. The effective output power relative to an isotropic radiator (EIRP) is now 26 dBm (24 dBm + 2 dBi). Assume the MAV and the ground station are 1 km apart. The logarithmic link budget equation is the standard method for determining signal strength at a receiver given a particular transmitter and antenna gain (Fig. 58.10) (Laskar et al. 2009; Green 2000).

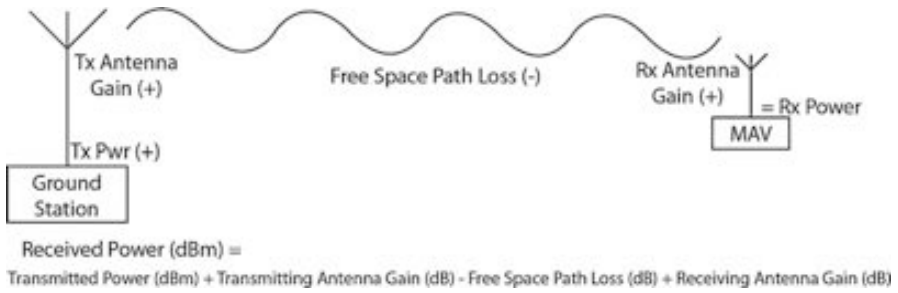


Fig. 58.10 Graphical representation of the logarithmic link budget equation

$$\begin{aligned}
 \text{ReceivedPower(dBm)} &= 24\text{dBm} + 2\text{dBi} + 2\text{dBi} - (32.45\text{dB} \\
 &\quad + 20 * \log[5, 800\text{Hz}] + 20 * \log[1\text{km}]) \\
 &= 28 - 107.7\text{dB} = -79.7\text{dBm}
 \end{aligned}$$

With a 4-element MIMO array having 18 dBi of gain at the MAV and ground station, received power = $60 - 107.7\text{ dB} = -47.7\text{ dBm}$.

This means that if you used MIMO array antenna arrays on both ends that you could lower your MAV transmitter power by 32 dB and achieve similar performance. That allows your 250 mW radio to become 200 μW . The effect on battery life is tremendous.

There is a slight penalty that must be paid when using array antennas and MIMO. MIMO processing is intensive, and therefore the precoding processor will consume some additional power. However, this power consumption (using current technology and system on a chip solutions) will not outweigh the savings gained by using beamforming and MIMO.

58.4.3 Diversity Matters

TDMA, as mentioned before, is a form of time diversity that allows multiple data streams to occupy the same channel. There is a second form of diversity called spatial diversity or antenna diversity (Laskar et al. 2009). Spatial diversity is achieved by utilizing antennas that are placed sufficiently far apart that the channel fading between the antennas vary and are therefore independent. Fading is the amount of loss along a given path between the transmitter and receiver. This antenna independence effectively creates multiple signal paths that can be used (Green 2000).

In most cases, multipath and scattering are considered to be negative in the world of communications. Multipath is the receiving of the same signal along multiple paths. For example, if a signal was received directly and a short time later a version of that same signal that bounced off a mountain and then to the receiver was received, then there would be an RF echo called multipath (Laskar et al. 2009).

Scattering is the deviation in the trajectory of a reflected signal due to variations in the medium that they are traversing (Paris and Hurd 1969). Both of these phenomena create problems when demodulating the signal.

However, in MIMO, multipath is actually helpful. This is a good thing since much of the MAV's role is in urban and indoor environments where signals are scattered and bounce off multiple surfaces readily. The scattering separates the signals in both space and time. This phenomenon is known as Rayleigh fading and has the effect of decorrelating the signals or making multiple independent versions of the signal with slightly different characteristics (Jiang 2004). The slight variation in characteristics can be detected and separated by a digital processor. Diversity gain takes advantage of the spatial diversity of the receive and transmit antennas to receive a higher-quality signal that would otherwise be possible (Gollakota et al. 2009).

This Rayleigh diversity, if properly exploited, can be used to achieve a diversity gain. With multiple transmit and receive antennas, signals can be differentiated despite the fact that they are close together. By using averaging of the signals received at each antenna, Gaussian white noise can also be cancelled, thereby increasing the signal-to-noise ratio (SNR). Linear matrix algebra can be used to correlate all of the channels and sum them to reconstruct the original data stream.

Between spatial diversity, temporal diversity, and beamforming gain, signals can be sufficiently strong to support communications even at reasonably long distances and inside buildings. For these reasons, MIMO is particularly useful for MAVs.

58.4.4 Additional Beamforming Advantages

Another advantage of beamforming is communications covertness. A beam is much harder for an eavesdropper to detect. Omnidirectional antennas make eavesdropping easy since the signal is being sent in all directions. With MIMO beamforming, an eavesdropper would have to be in the beam's path to detect the signal. For covert operations, having a minimal signature both acoustically and electromagnetically is preferred. With a plethora of low-cost frequency scanners on the market, it is important to have direct narrow-beam communications to avoid detection.

58.5 Conclusion

When considering the best communications architecture for MAVs, designers must remember that everything is a trade-off. As such, designers should use a three-step process to arrive at the optimal solution. First, determine the characteristics of the operating environment. This should consider urban vs. rural, indoors vs. outdoors, and short vs. long distance. This determination will dictate which frequencies, modulation schemes, and multiple access schemes are appropriate. Second, designers should determine the data rates involved. High data rates require the sacrifice of other desirable aspects. The proper balance must be determined, and consideration of data rates will help pair down the options resulting from the first

consideration. Finally, the designer should decide what level of data link reliability is required. This again is in direct conflict with the other two considerations. Once the link parameters have been considered in light of these three steps, iterate once more through them to arrive at the optimal solution for the particular MAV application. Talking with an RF engineer early in the design process is advisable since many of the optimal solutions require integration with the air vehicle in a way that is not conducive to implementation late in the project.

As technology continues to miniaturize and advance, many of the weight and power constraints will ease. Unfortunately, RF characteristics are largely bounded by the universal constraints of nature and are therefore not controllable by the designer. RF links and power will continue to be problematic for MAVs for many years to come, particularly as MAVs become smaller. The best that designers can hope for is an optimal communications solution given the current state of technology. Everything else must then be built around that solution.

References

- AeroVironment, Inc., AeroVironment, Inc., (AV): UAS Advanced Development Center: Hornet (UAV): UAS Advanced Development Center: AeroVironment, Inc. (AV) (2012), <http://www.avinc.com/uas/adc/hornet/>. Accessed 11 Feb 2012
- O.M. Bakr, A Scalable and Cost Effective Architecture for High Gain Beamforming Antennas, Thesis, University of California, Berkley, 2010, p. 35
- E. Beidel, Navy's Electric Gun Could Hit Targets More Than 100 Miles Away, NDIA National Defense Magazine (2012), <http://www.nationaldefensemagazine.org/archive/2012/April/Pages/Navy'sElectricGunCouldHitTargetsMoreThan100MilesAway.aspx>. Accessed 8 Feb 2012
- E. Biglieri, et al., *MIMO Wireless Communications* (Cambridge University Press, Cambridge/New York, 2007), pp. 88–133
- J.J. Carr, *Microwave and Wireless Communications Technology* (Newnes, Boston, 1997), pp. 175–191, 410–412
- D.W. Chi, P. Das, *Effects of Jammer and Nonlinear Amplifiers in MIMO-OFDM with Application to 802.11n WLAN* (Department of Electrical and Computer Engineering, University of California, San Diego, 2008), pp. 1–8
- C.H. Durney, C.C. Johnson, *Introduction to Modern Electromagnetics* (McGraw-Hill, New York, 1969), pp. 250–258
- S. Gollakota, et al., *Interference Alignment and Cancellation*. SIGCOMM '09, Barcelona, Spain, 2009, pp. 1–6
- J.H. Green, *The Irwin Handbook of Telecommunications* (McGraw-Hill, New York, 2000), pp. 321–338
- J. Jiang, Measurement, Modeling, and Performance of Indoor MIMO Channels, Thesis, Georgia Institute of Technology, 2004, pp. 160–164
- A. Kashyap, et al., *Correlated jamming on MIMO Gaussian fading channels*. IEEE Transactions on Information Theory (IEEE Digital Library, 2004), pp. 3–5
- J. Laskar, S. Chakraborty, M. Tentzeris, F. Bien, A. Pham, *Advanced Integrated Communication Microsystems*, ed. K. Chang (Wiley, Hoboken, 2009), pp. 83–87
- J.M. McMichael, M.S. Francis, *Micro Air Vehicles – Toward a New Dimension in Flight*, Federation of American Scientists (1997), http://www.fas.org/irp/program/collect/docs/mav_auvsi.htm. Accessed 11 Jan 2012
- National Research Council, *Implications of Emerging Micro and Nanotechnology* (The National Academies Press, Washington DC, 2002), p. 213

- D.T. Paris, F.K. Hurd, *Basic Electromagnetic Theory* (McGraw-Hill, New York, 1969), pp. 377–378
- P. Quilter, Amplifier anatomy. *Sound and Video Contractor Magazine*, 1993, pp. 6–7
- S.E. Schwarz, *Electromagnetics for Engineers* (Oxford University Press, New York, 1990), pp. 350–369
- J. H. Sung, Transmitter Strategies for Closed-Loop MIMO-OFDM, Thesis, Georgia Institute of Technology, 2004, pp. 24–65
- J. Voelcker, *Market Watch: Electric Cars Move Slowly* (MIT Technology Review, 2009), <http://www.technologyreview.com/energy/23723/>. Accessed 22 Jan 2012
- C.J. Weisman, *The Essential Guide to RF and Wireless* (Prentice-Hall PTR, Upper Saddle River, 2002), pp. 38–47, 52–54, 119–127, 145–149, 188–190, 201–205, 209–218
- R. Zhu, et al., *Integrated Design of Trajectory Planning and Control for Micro Air Vehicles*. *Mechatronics: The Science of Intelligent Machines* (International Federation of Automatic Control, Elsevier, 2007), p. 1

Section XII

UAV Mission and Path Planning

***Camille Alain Rabbath and
David Hyunchul Shim***

Kimón P. Valavanis and George J. Vachtsevanos

UAV Mission and Path Planning moves beyond fundamental UAV technologies to higher level functionalities such as mission and path planning for single and multiple UAVs, as UAVs are deployed to execute a variety of “missions” in both military and civilian environments. Planning and executing a mission in terms of prescribing the path or trajectory generation and other related topics are challenging problems that require careful consideration. The chapters in this section present a plethora of methods and techniques, along with specific examples from the utility of mission and path planning technologies, suitable to assist the planner to complete an assigned mission.

► *Cooperative Mission Planning for Multi-UAV Teams* by Ponda, Johnson, Geramifard, and How provides an overview of three of the most common planning frameworks: integer programming, Markov decision processes, and game theory. It also considers various architectural decisions that must be addressed when implementing online planning systems for multi-agent teams, providing insights on when centralized, distributed, and decentralized architectures might be good choices for a given application and how to organize the communication and computation to achieve desired mission performance. Algorithms that can be utilized within the various architectures are identified and discussed, and future directions for research are suggested.

► *Multi-team Consensus Bundle Algorithm* by Argyle, Beard, and Casbeer focuses on how the consensus-based bundle algorithm (CBBA) is incorporated into

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

a hierarchical concept of operation, where agents are divided into teams and each team plans for its agents to service a set of tasks. This team planning is carried out in a distributed manner using the traditional CBBA. An “outer-loop” team CBBA strategy coordinates teams’ plans. The hierarchical structure of the team CBBA facilitates a manageable architecture for large numbers of unmanned agents through human-centered operations. This is because each team is managed by a human operator with the team CBBA aiding coordination between teams.

► *Cooperative Mission and Path Planning for a Team of UAVs* by Oh, Shin, Kim, Tsourdos, and White addresses the cooperative mission and path planning problem of multiple UAVs in the context of the vehicle routing problem. Since the conventional vehicle routing algorithms approximate their path to straight lines to reduce computational load, the physical constraints imposed on the vehicle are not to be taken into account. In order to mitigate this issue, a framework is described allowing integrated mission and path planning for coordinating UAVs using the Dubins theory based on the differential geometry concepts which can consider non-straight path segments. The main advantage of this approach is that the number of design parameters can be significantly reduced while providing the shortest, safe, and feasible path, which leads to a fast design process and more lightweight algorithms. In order to validate the integrated framework, cooperative mission and path planning algorithms for two missions are developed: (1) road-network search route planning patrolling every road segment of interest efficiently based on the optimization and approximation algorithm using nearest insertion and auction negotiation and (2) communication relay route planning between a ground control station and the friendly fleet satisfying the constraints on the UAV speed and the avoidance of nonflying zones.

► *Cooperative Task Assignment and Path Planning for Multiple UAVs* by Moon, Shim, and Oh presents a hierarchical framework for path planning and task assignment for multiple UAVs in a dynamic environment. The discussed path planning algorithm is based on the visibility and shortest path principles in Euclidean space, with the A* algorithm adopted to find an admissible path in a “best-first” approach during the search process. The path planner is augmented with a potential field-based trajectory planner, which solves for a detouring trajectory around other agents or pop-up obstacles. Task assignment is achieved by a negotiation-based algorithm, which assigns a task with the lowest cost to each agent after comparing all task costs of all participating agents. These algorithms are implemented on MATLAB/Simulink, which can run with simulated vehicle models or actual UAVs through a communication network. In the simulations, the algorithms are validated to perform task assignment and path planning flawlessly. In actual flight tests, the proposed algorithms were tested with a number of fixed-wing UAVs in a fully realistic situation under various reality factors such as communication loss or tracking errors. The flight test shows, even in the presence of such uncertainties and logistic factors, the algorithms were able to perform all of the given tasks without any collision with other agents or obstacles.

► *On the Decentralized Cooperative Control of Multiple Autonomous Vehicles* by Hirsch and Schroeder is concerned with dynamically determining appropriate

flight patterns for a set of autonomous UAVs in an urban environment, with multiple mission goals. The UAVs are tasked with searching the urban region for targets of interest and tracking those targets that have been detected. It is assumed that there are limited communication capabilities between the UAVs and that there exists possible line of sight constraints between the UAVs and the targets. Each UAV operates its own dynamic feedback loop, in a receding-horizon framework, incorporating local information as well as remote information to determine the task to perform and the optimal flight path of UAV over the planning horizon. This results in a decentralized and more realistic model of the real-world situation. As the coupled task assignment and flight route optimization formulation is NP-hard, a hybrid heuristic for continuous global optimization is developed to solve for the flight plan and tasking over the planning horizon. Experiments are considered as communication range between UAVs varies.

► *Innovative Collaborative Task Allocation for UAVs* by Karaman, Koyuncu, and Inalhan is devoted to a large-scale distributed task/target assignment problem for a fleet of autonomous UAVs, deriving an algorithm that uses the delayed column generation approach on a non-convex supply-demand formulation for the problem. The algorithm exploits a computationally tractable distributed coordination structure, i.e., a market for tasks and targets created by the UAV fleet. The resulting structure is solved via a fleet-optimal dual simplex ascent in which each UAV updates its respective flight plan costs with a linear update of waypoint task values as evaluated by the market. Synchronized and asynchronous distributed implementations of this approximation algorithm are demonstrated in a thorough experimental study involving dynamically changing scenarios with random pop-up targets.

► *Control of Communication Networks for Teams of UAVs* by Kopeikin, Ponda, and How studies the communication network that is a fundamental component of a multi-UAV system, as it enables exchanges in command and control messages and allows for remotely sensed mission data to be sent to processing centers. Network control is critical for the system to function properly. Challenges in network communication control are presented, followed by a background discussion in wireless networking including system architecture, wireless channel performance, topology models, and information routing, as well as bounds and limitations in cooperative control and decision making because of network limitations. Several network control strategies for multi-UAV systems are discussed, including motion planning methods to control the topology and relay deployment techniques to extend the performance of the network.

► *Information-Theoretic Exploration of Unmanned Aerial Vehicle in Unknown Cluttered Environment* by Yang, Gan, and Sukkarieh proposes an information-theoretic path planning method for exploration mission in unknown environments. Instead of using traditional grid-based occupancy maps, the Gaussian process (GP) is used to build an environmental map. The GP map performs inference directly on the collected sensor datasets, which allows it to infer the probability of collision for any query point in continuous 3-D space, removing the need for maintaining a full discrete map. This GP map is fused with the rapidly exploring random tree (RRT) path planner to plan a safe path and acquire information about the

unknown environments. Using mutual information as an information measure, the most informative path is chosen as the path for exploration. Simulation results show that GP map combined with RRT planner achieves exploration task successfully in unknown complex environment and present their potential implementation in UAV missions.

► *Implementing Dubins Airplane Paths on Fixed-Wing UAVs* by Beard and McLain presents a 3-D Dubins airplane model, deriving a complete architecture for implementing Dubins airplane paths on small fixed-wing UAVs. A vector field method is then used to design a guidance law that causes the Dubins airplane model to follow straight-line and helical paths. Dubins airplane paths are more complicated than Dubins car paths because of the altitude component. Based on the difference between the altitude of the start and end configurations, Dubins airplane paths are classified as low-, medium-, or high-altitude gain. While for medium- and high-altitude gains, there are many different Dubins airplane paths, it is proposed to select the path that maximizes the average altitude throughout the maneuver. The proposed architecture is implemented on a six-degrees-of-freedom MATLAB/Simulink simulation of an Aerosonde UAV, and results from this simulation demonstrate the effectiveness of the technique.

► *Health Monitoring of a Drone Formation Affected by a Corrupted Control System* by Lechevin, Rabbath, and Maupin proposes a dynamic feature that is instrumental in achieving vulnerability assessment of a network of UAVs, whose control system is possibly affected by the diffusion of malware. The feature consists of the characterization of the transition from stability to instability with probability one. The stability of the networked UAVs can be indirectly affected by malicious attacks targeting the communication units or the control systems. The network is modeled as a discrete-time, jump, linear system whose state space variables represent the probabilities that each node receives a malware and is infected by it. The stability analysis is obtained by means of a stochastic Lyapunov function argument and yields a sufficient condition expressed as a linear matrix inequality (LMI). This LMI involves the networked asset state space matrices and the probability that each UAV's control system is infected. An approximation to the sufficient condition is proposed so that convergence of the system trajectories could be monitored online.

Sameera S. Ponda, Luke B. Johnson, Alborz Geramifard, and
Jonathan P. How

Contents

| | | |
|--------|---|------|
| 60.1 | Introduction | 1448 |
| 60.2 | Approaches | 1449 |
| 60.2.1 | Integer Programming | 1449 |
| 60.2.2 | Markov Decision Processes | 1456 |
| 60.2.3 | Game Theory | 1464 |
| 60.3 | Planning Architectures | 1468 |
| 60.3.1 | Centralized | 1469 |
| 60.3.2 | Distributed | 1470 |
| 60.3.3 | Decentralized | 1471 |
| 60.3.4 | Synchronous Versus Asynchronous | 1471 |
| 60.3.5 | Performance Metrics | 1472 |
| 60.3.6 | Coordination Techniques | 1473 |
| 60.3.7 | Consensus | 1475 |
| 60.4 | Decentralized Algorithms | 1477 |
| 60.4.1 | Market-Based Approaches | 1477 |
| 60.4.2 | Decentralized Markov Decision Processes | 1480 |
| 60.5 | Conclusions and Future Directions | 1483 |
| | References | 1484 |

Abstract

The use of robotic agents, such as unmanned aerial vehicles (UAVs) or unmanned ground vehicles (UGVs), has motivated the development of numerous autonomous cooperative task allocation and planning methods for heterogeneous networked teams. Typically agents within the team have different roles and responsibilities, and ensuring proper coordination between them is critical for

S.S. Ponda (✉) • L.B. Johnson • A. Geramifard • J.P. How

Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts
Institute of Technology, Cambridge, MA, USA

e-mail: sponda@alum.mit.edu; lbj16@mit.edu; agf@csail.mit.edu; jhow@mit.edu

efficient mission execution. However, as the number of agents, system components, and mission tasks increase, planning for such teams becomes increasingly complex, motivating the development of algorithms that can operate in real-time dynamic environments.

Given the complexity of the cooperative missions considered, there have been numerous solution approaches developed in recent years. This chapter provides an overview of three of the most common planning frameworks: integer programming, Markov decision processes, and game theory. The chapter also considers various architectural decisions that must be addressed when implementing online planning systems for multi-agent teams, providing insights on when centralized, distributed, and decentralized architectures might be good choices for a given application, and how to organize the communication and computation to achieve desired mission performance. Algorithms that can be utilized within the various architectures are identified and discussed, and future directions for research are suggested.

60.1 Introduction

The use of robotic agents, such as unmanned aerial vehicles (UAVs) or unmanned ground vehicles (UGVs), has motivated the development of numerous autonomous cooperative task allocation and planning methods. Applications of these planning algorithms include utilizing teams of heterogeneous networked agents to accomplish autonomous missions (e.g., intelligence, surveillance, and reconnaissance (ISR) operations (U.S. Army UAS Center of Excellence 2010; U.S. Air Force Chief Scientist AF/ST; Office of the Secretary of Defense 2007)), as shown in Fig. 60.1. Other promising applications of autonomous vehicles include environmental disaster relief (Maza et al. 2011); chemical, biological, radiological, and nuclear (CBRN) search and rescue in the aftermath of an attack in an urban area (Welch and Edmonds 2003); precision agriculture (Schmale et al. 2008); fighting forest fires (Sujit et al. 2007); and weather forecasting (Frew and Argrow 2010). Typically agents within the team have different roles and responsibilities, and ensuring proper coordination between them is critical for efficient mission execution. This process requires ensuring spatial and temporal synchronization of the team, while considering mission costs, available resources, and network constraints.

Given the complexity of the cooperative missions considered, there have been numerous solution approaches developed in recent years. The following sections briefly describe three common formulations of the problem: integer programming, Markov decision processes, and game theory. The chapter also considers various architectural decisions that must be addressed when implementing online planning systems, providing insights on when centralized, distributed, and decentralized architectures might be good choices for a given application, and how to organize the communication and computation to achieve desired mission performance.

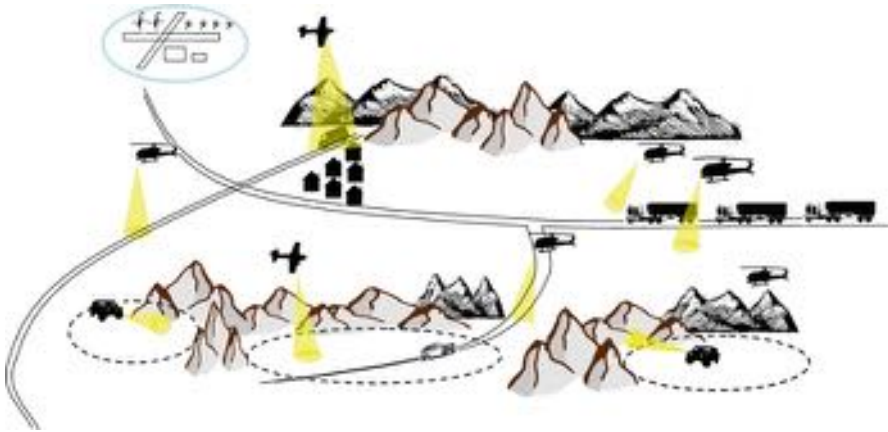


Fig. 60.1 Illustration of an example intelligence, surveillance, and reconnaissance (ISR) mission scenario involving numerous UGV and UAV assets that must perform a variety of search, track, and escort tasks

Algorithms that can be used within each of these architectures are identified, technical challenges and limitations are discussed, and directions for future research are suggested.

60.2 Approaches

This section introduces three of the most common planning frameworks for cooperative multi-agent decision-making problems: integer programming, Markov decision processes, and game theory.

60.2.1 Integer Programming

60.2.1.1 Problem Statement

Given a set of N_t tasks and a set of N_a agents, the goal of a task allocation algorithm is to find an allocation that maximizes a global reward function while enforcing all of the problem- and domain-specific constraints. This generalized task assignment problem can be written as

$$\begin{aligned} \underset{\mathbf{s}, \mathbf{x}, \boldsymbol{\tau}}{\operatorname{argmax}} \quad & \sum_{j=1}^{N_t} R_j(\mathbf{s}, \mathbf{x}, \boldsymbol{\tau}) \\ \text{subject to:} \quad & \mathbf{s} \in \mathcal{S}(\mathbf{x}, \boldsymbol{\tau}), \quad \mathbf{x} \in \mathcal{X}, \quad \boldsymbol{\tau} \in \mathcal{T}. \end{aligned} \quad (60.1)$$

In Eq. 60.1, $R_j(\mathbf{s}, \mathbf{x}, \boldsymbol{\tau})$ represents the score achieved by the team for task j . This score is a function of \mathbf{s} , representing the state trajectories for all agents; of \mathbf{x} ,

a matrix of decision variables (specifically where x_{ij} is a binary decision variable equal to 1 if task j is assigned to agent i and 0 otherwise); and τ_{ij} is a variable representing the time at which agent i services task j if $x_{ij} = 1$ and is undefined otherwise. The index set that iterates over agents i is defined as $\mathcal{I} \triangleq \{1, \dots, N_a\}$, and the index set that iterates over tasks j is defined as $\mathcal{J} \triangleq \{1, \dots, N_t\}$. $\mathcal{S}(\mathbf{x}, \boldsymbol{\tau})$ defines the set of possible trajectories that satisfy both vehicle and environmental constraints as a function of \mathbf{x} and $\boldsymbol{\tau}$; \mathcal{X} defines the set of all feasible task allocations taking into account agent capability constraints and constraints between tasks; and \mathcal{T} defines the set of all feasible task servicing times, taking into account possibly complicated temporal constraints between these tasks. In general, the full constraint space of the task allocation environment is defined by the triple of $(\mathcal{S} \times \mathcal{X} \times \mathcal{T})$. This state space can be used to specify most task allocation and motion planning problems. Because of this extreme generality, the cardinality of the state space is uncountably infinite and thus is a very difficult space to search, even approximately. The following sections will outline some simplifying assumptions that are typically added to the above formulation to make solution approaches tractable.

Task Servicing Uniqueness

A standard assumption made in the task allocation literature is that every task may be assigned to at most one agent. Roughly speaking, this means that two agents are not allowed to service the same task. While often true for most missions, this assumption also has the effect of dramatically reducing the cardinality of \mathcal{X} . This task servicing uniqueness constraint is formalized as

$$\sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J},$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{I} \times \mathcal{J}.$$

Even though this constraint explicitly prohibits multiple agents from performing the same task, it still allows for cooperation given some creative task construction. For example, some approaches like posting multiple tasks at the same location and time (implying cooperation will take place) are a brute force way to establish cooperation at a task location. Other approaches may encode this cooperation explicitly as a more complicated constraint, rather than requiring uniqueness of assignment.

Task Independence

Another simplifying assumption is that the score achieved for each task is independent of the completion of all other tasks and locations of all other agents. This enables the reward function to be decoupled so that the only two things that affect the score of the task are the capabilities of the agent servicing the task (including the availability of the agent due to commitments of servicing other tasks) and what time the task is actually serviced. The form of the reward function in Eq. 60.1 then simplifies to

$$\sum_{j=1}^{N_t} R_j(\mathbf{s}, \mathbf{x}, \boldsymbol{\tau}) \rightarrow \sum_{j=1}^{N_t} R_j(\mathbf{x}_j, \boldsymbol{\tau}_j), \quad (60.2)$$

where $\mathbf{x}_j \triangleq \{x_{1j}, \dots, x_{N_a j}\}$ is a vector indicating which agent i (if any) will be servicing task j and $\boldsymbol{\tau}_j \triangleq \{\tau_{1j}, \dots, \tau_{N_a j}\}$ is the corresponding vector that indicates the service time of task j (this vector will have at most 1 defined component corresponding to the assigned agent). It is worth noting that Whitten (Whitten et al. 2011) also addressed the problem of introducing more complicated task dependencies consistent with Eq. 60.2 above. The solution he proposed was to add constraints that make the task selection infeasible if certain constraints were not met. This introduces coupling in the task constraints instead of in the score function. Explicit coupling in the score function between tasks can often violate a property called *submodularity*. Submodularity is an important property in greedy algorithms because it allows assignments to perform provably well with respect to optimal. Because of this, introducing the coupling as a constraint is often a more tractable approach to searching the assignment space.

Decoupling of Path Planning Solution

Another standard approximation of the original problem formulation is to avoid explicitly solving for \mathbf{s} , the actual dynamic paths that every agent must follow to service its assigned tasks (Richards et al. 2002; Bellingham et al. 2002). Instead, the approach is to guarantee that, given a feasible task allocation \mathbf{x} and a set of service times $\boldsymbol{\tau}$, there *exists* some $\mathbf{s} \in \mathcal{S}(\mathbf{x}, \boldsymbol{\tau})$. The problem of actually optimizing \mathbf{s} is deferred to path planning optimization. Typically conservative estimates of the maneuvering capabilities of the agent are used, leading to straight-line (or possibly Dubin's car) path estimates of the route and timing (Rathinam et al. 2007; Savla et al. 2005; Bellingham et al. 2002). While this decoupling greatly reduces the complexity of the full task allocation optimization, the accuracy of the timing and cost estimates used in the planner can be a concern, possibly leading to poor performance if the optimized paths vary greatly from those assumed in the task allocation process.

60.2.1.2 Full Problem Description

With these standard assumptions, the full problem of interest can be written as

$$\underset{\mathbf{x}, \boldsymbol{\tau}}{\operatorname{argmax}} \quad \sum_{j=1}^{N_t} R_j(\mathbf{x}_j, \boldsymbol{\tau}_j), \quad (60.3)$$

$$\text{subject to: } \sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J},$$

$$\mathbf{s} \in \mathcal{S}(\mathbf{x}, \boldsymbol{\tau}), \quad \mathbf{x} \in \mathcal{X}, \quad \boldsymbol{\tau} \in \mathcal{T},$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{I} \times \mathcal{J}.$$

This formulation, although mathematically complete, is not structured in a convenient form to explain the algorithmic tools used to solve it. The following sections introduce further notation to highlight where the problem can be decoupled and what pieces can be solved separately and introduce some of the data structures used to keep track of the relevant information:

1. For each agent, an ordered data structure called a *path* is defined, $\mathbf{p}_i \triangleq \{p_{i1}, \dots, p_{i|\mathbf{p}_i|}\}$, whose elements are defined by $p_{in} \in \mathcal{J}$ for $n = \{1, \dots, |\mathbf{p}_i|\}$. The path contains the information representing what tasks are assigned to agent i . The order of the tasks in the path represents the relative order in which they will be serviced.
2. A score function matrix $\mathbf{c}(t)$ is defined, which is composed of elements c_{ij} that represent the score that agent i would receive by servicing task j at time t .
3. For each agent i , a maximum path length L_i is defined, representing the maximum number of tasks that the agent may be assigned. This is an artificial constraint placed on the problem to help with computation and communication efficiency. This constraint is often paired with an implicit receding time horizon to guarantee that an agent does not commit to tasks too far in the future and exhaust its path length.

Using these new data structures, the problem definition can be augmented as follows:

$$\begin{aligned} \underset{\mathbf{x}, \boldsymbol{\tau}}{\operatorname{argmax}} \quad & \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x}_i))) x_{ij} \right), \quad (60.4) \\ \text{subject to:} \quad & \sum_{j=1}^{N_t} x_{ij} \leq L_i, \quad \forall i \in \mathcal{I}, \\ & \sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J}, \\ & \mathbf{s} \in \mathcal{S}(\mathbf{x}, \boldsymbol{\tau}), \quad \mathbf{x} \in \mathcal{X}, \quad \boldsymbol{\tau} \in \mathcal{T}, \\ & x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{I} \times \mathcal{J}. \end{aligned}$$

Given the formulation specified in Eq. 60.4, it may become apparent that the computation can be distributed across multiple agents. At this point, given \mathbf{x} , the matrix $\boldsymbol{\tau}$ can be determined by each agent independently. Furthermore, with \mathbf{x} , each individual agent can find its own contribution to the full matrix $\boldsymbol{\tau}$ by solving

$$\underset{\boldsymbol{\tau}}{\operatorname{argmax}} \quad \sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x}_i))) x_{ij}. \quad (60.5)$$

Using Eq. 60.5, each agent i will first have to optimize its own path \mathbf{p}_i , given a list of tasks it has a responsibility to service, \mathbf{x}_i , and subject to the path planning feasibility constraint $\mathbf{s} \in \mathcal{S}(\mathbf{x}, \boldsymbol{\tau})$ and the feasibility of arrival times constraint $\boldsymbol{\tau} \in \mathcal{T}$. During the path optimization, optimal values for $\boldsymbol{\tau}$ will also be determined. Furthermore, from this optimization, the agent's individual scores for each assigned task will be fully determined. The main result of Eq. 60.5 is that the primary source of distributed coupling in the path assignment problem is restricted to the choice of the assignment vector \mathbf{x} . Given this, once all agents agree on a value for \mathbf{x} , the distributed task assignment problem has been solved.

60.2.1.3 Planning in Uncertain Domains

An important issue associated with cooperative planning for networked teams is that planning algorithms rely on underlying system models, which are usually approximations of the real systems involving many limiting assumptions and complex dynamics. Discrepancies between these planner models and the actual system dynamics cause degradations in mission performance. Furthermore, the impact of these discrepancies on the overall quality of the plan is typically hard to quantify in advance due to nonlinear effects, coupling between tasks and agents, and interdependencies between system constraints (e.g., longer-than-expected service times impact the arrival times of subsequent tasks). However, if uncertainty models of planning parameters are available, they can be leveraged to create robust plans that explicitly hedge against the inherent uncertainty given allowable risk thresholds.

For example, for the problem definition introduced in Eq. 60.4, if the score function matrix $\mathbf{c}(t)$ depends on uncertain planning parameters with associated probability distributions, stochastic planning algorithms can be used to create plans that explicitly account for the *variability* in $\mathbf{c}(t)$. There are several metrics that can be used to account for uncertainty in the planning formulation. Perhaps the most common approach is to maximize the expected mission performance (Bertsekas 2007), where the objective function from Eq. 60.4 becomes

$$\operatorname{argmax}_{\mathbf{x}, \boldsymbol{\tau}} \quad \mathbb{E} \left\{ \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x}_i))) x_{ij} \right) \right\}. \quad (60.6)$$

Note that optimizing Eq. 60.6 is *not* the same as planning using the mean values of uncertain planning parameters, which can often lead to poor planning performance since the problem formulation fails to capture the nontrivial coupling of uncertainty in scores, dynamics, and constraints. This is especially problematic when scores are coupled and can lead to biased predictions that drastically misrepresent the actual expected performance.

While optimizing Eq. 60.6 provides a plan that maximizes the expected performance of the system, an actual single run execution of this best expected plan is still subject to the uncertainty in the environment and may result in a relatively poor plan (worse than expected) with some nonzero probability. If the current

mission tolerance to failure is very low, a more *robust* planning objective involves maximizing the worst-case scenario:

$$\operatorname{argmax}_{\mathbf{x}, \tau} \min \left\{ \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} c_{ij} (\tau_{ij} (\mathbf{p}_i (\mathbf{x}_i))) x_{ij} \right) \right\}. \quad (60.7)$$

Optimizing Eq. 60.7 guarantees that the plan execution will result in a score *no worse* than that predicted by the algorithm; however, sometimes this is too conservative (especially for parameter distributions with infinite support).

Several robust optimization and stochastic optimization methods, that find the best solution within some predefined risk threshold, have been developed to mitigate this issue of conservatism (Nemirovski and Shapiro 2007; Bertsimas et al. 2011). One such approach involves optimizing a risk-adjusted expected performance, where a risk function $R(c_{ij})$ biases the original cost function c_{ij} towards more conservative solutions to account for the acceptable level of risk. Another approach is to bound the domain of the uncertainty set to be within certain ranges (e.g., ellipsoids for specified confidence intervals (Ben-Tal and Nemirovski 1998)) or to take on a set of discrete representative values (Chen et al. 2010; Bertsimas and Brown 2009), thus limiting the support of the uncertain parameters. Classical robust convex optimization techniques can then be used to solve the resulting approximate problem. Although these methods allow control over the planner risk, there are a few issues which make their practical implementation difficult. Firstly, it is not usually obvious how to design the risk functions $R(c_{ij})$ or the bounded uncertainty sets, and the selection of these is typically problem specific, time consuming, and ad hoc. A more serious issue, however, is that when a global risk threshold is available, the metric of interest is the *cumulative* risk of the total solution, not the individual parameter or task risks, and it is difficult to quantify how these individual parameter bounds will affect the global risk. Nonlinearities in the cost functions, complex variable coupling and interdependencies, and discrete optimization effects often affect the solution in unpredictable ways, and it is therefore hard to ensure that the total mission outcome is within the desired risk threshold.

An alternative approach that guarantees that the global mission performance will be within a certain risk threshold is the chance-constrained formulation (Delage and Mannor 2010; Nemirovski and Shapiro 2007; Blackmore and Ono 2009),

$$\begin{aligned} & \operatorname{argmax}_{\mathbf{x}, \tau} y \\ & \text{s.t. } \mathbb{P} \left\{ \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} c_{ij} (\tau_{ij} (\mathbf{p}_i (\mathbf{x}_i))) x_{ij} \right) > y \right\} \geq 1 - \epsilon. \end{aligned} \quad (60.8)$$

The goal of Eq. 60.8 is to maximize the worst case score within the allowable risk threshold specified by ϵ (can be interpreted as guaranteeing that the solution will be at least as good as y with probability greater than or equal to $1 - \epsilon$). When $\epsilon = 0$,

the score is guaranteed to be at least y with probability 1 (absolute worst-case), and the chance-constrained formulation reduces to the robust formulation of Eq. 60.7. The main drawback of the chance-constrained formulation is that it is difficult to solve, especially given the extensive coupling between agents and tasks (double sum over distributions). Previous work has mainly considered linear or quadratic optimization with continuous variables (Delage and Mannor 2010; Nemirovski and Shapiro 2007; Blackmore and Ono 2009), where, under special circumstances, optimal solutions and bounds can be found analytically. However, task allocation is a mixed-integer program, and these techniques cannot be easily extended to discrete optimization, especially given nonlinear and heterogeneous score functions with coupled distributions. Furthermore, these solution strategies are centralized and cannot be trivially extended to distributed environments. Recent work has considered implementing distributed approximations for chance-constrained multi-agent programs (Ponda et al. 2012); however, this remains an open area of research.

60.2.1.4 Solution Algorithms

Numerous researchers have examined many aspects of the UAV assignment problem (Chandler et al. 2002; Schumacher et al. 2002; Bellingham et al. 2002; Castanon and Wohletz 2009; McLain and Beard 2005; Beard et al. 2002; Passino et al. 2002; Saber et al. 2003; Ren et al. 2007; Shima and Rasmussen 2009; Leary et al. 2011; Kim et al. 2007). This includes traditional methods for vehicle routing problems (VRPs) from the operations research (OR) and artificial intelligence (AI) communities (Toth and Vigo 2001; Cormen 2001). Exact optimization methods such as branch and bound, branch and cut, constraint satisfaction problems (CSPs), and dynamic programming (DP) have been used to solve the problem to optimality, using standard software solvers (ILOG 2006). While guaranteed to yield optimal results, these methods are computationally intensive, and this complexity becomes an important issue when the problem has *hard* side constraints (Floudas 1995). These complexity issues typically make optimal solution techniques intractable for many problems of interest, and, as a result, numerous approximation methods have been proposed. Classical heuristic methods, such as constructive and two phase methods, have been used to solve large VRP problems relatively quickly (Laporte and Semet 2002), but these methods can generate solutions that are far from optimal. Different heuristic methods such as tabu search (Glover and Marti 2006), cross-entropy (De Boer et al. 2005; Undurti and How 2010; Manh et al. 2010), particle swarm optimization (Salman et al. 2002; Cruz Jr et al. 2004), genetic algorithms (Chockalingam and Arunkumar 1992; Shima et al. 2006; Eun and Bang 2010), and evolutionary algorithms (Nikolos et al. 2007; Pongpunwattana et al. 2003) have also been proposed in recent years to solve these complex optimization problems. These approximations help to reduce the computation time as compared to exact methods, but most of these algorithms are still computationally intractable for real-time replanning environments with complex constraints.

Several strategies have been considered in the literature to further reduce the computation time required to solve these problems. One approach is to reduce the

problem size by limiting the duration (in time or plan length) using a receding horizon formulation (King et al. 2004; Alighanbari and How 2008a; Castanon and Wohletz 2009). A key challenge here, however, is to develop an effective and computationally efficient approximation of the “cost-to-go” from the terminal point of the plan, to avoid having an overly short-sighted planner. Other authors have identified several efficient mixed-integer linear programming formulations that dramatically reduce the number of variables and constraints required. Although problem specific and applicable only in certain environments, these formulations cover a variety of UAV task assignment problems of practical interest (Alidaee et al. 2009, 2011). These reductions significantly alleviate the computational effort required and should always be considered if the particular problem of interest fits into the appropriate formulation. Other approaches have considered fast constant-factor approximate algorithms for multi-UAV assignment problems, which involve making mild assumptions to approximate the vehicle dynamics (e.g., constraints on turning radius) (Rathinam et al. 2007). Finally, recent work has considered employing *learning* techniques to guide the MILP solution process (Banerjee et al. 2011). The approach involves using machine learning to efficiently estimate the objective function values of the LP relaxations to within an error bound and to use that information to perform fast inference at run time for new but similar relaxation problems that occur when the system dynamics and/or the environment changes slightly.

The algorithms presented in this section are all centralized solution strategies, meaning that all of the planning relevant computation occurs in a single location. Section 60.4 breaks this assumption and describes how certain algorithms are able to handle this.

60.2.2 Markov Decision Processes

Another well-studied framework to enable cooperative planning strategies for multi-agent teams is the Markov decision process (MDP), which constitute a formal architecture for representing stochastic sequential decision-making problems (Sutton and Barto 1998; Buşoniu et al. 2010). Cooperative planners based on MDPs have shown tremendous versatility in modeling multi-agent systems (Guestrin et al. 2001; Stone et al. 2005; Valenti et al. 2007; Bethke et al. 2008), and the associated solution algorithms, such as dynamic programming, have enabled near-optimal solutions in many application domains. A well-known issue however, as described in the previous section, is that MDPs and related dynamic programming techniques suffer from bad scalability (Bellman 2003) and quickly become intractable as the number of participating agents increases. As a result, many approximation algorithms have been developed to mitigate this computational challenge (Sutton 1996; Bertsekas 2007; Mahadevan et al. 2006; Buşoniu et al. 2010; Geramifard et al. 2011). While many of these approaches have been successfully used to solve large multi-agent problems which would otherwise remain intractable (Stone et al. 2005; Bethke 2010), most have fundamental limitations such as difficulty in

selecting appropriate approximation architectures or absence of proper guidance in tuning algorithm parameters. This section describes the MDP architecture and its variants and discusses popular solution approaches, challenges, and limitations. Section 60.4.2 extends the frameworks presented in this section to address the additional requirements associated with decentralized planning.

60.2.2.1 Problem Statement

A finite horizon Markov decision process can be described by the following tuple $\langle \mathcal{S}, \mathcal{A}, P, R, T \rangle$, where the elements are described as follows:

- \mathcal{S} : Defines the state space which is composed of a discrete set of states $s \in \mathcal{S}$
- \mathcal{A} : Defines the action space with a discrete set of actions $a \in \mathcal{A}$
- P : Defines a transition function of the form $P(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ specifying the probability of transitioning to state $s' \in \mathcal{S}$ from state $s \in \mathcal{S}$ having taken action $a \in \mathcal{A}$
- R : Defines a reward function $R(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ specifying the rewards and costs associated with applying action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and transitioning to state $s' \in \mathcal{S}$
- T : Represents the discrete finite horizon (number of time steps)

A *trajectory* is a sequence $\{s_0, a_0, r_0, s_1, a_1, r_1, s_2, \dots\}$, where the action a_t is chosen given a *policy* $\pi : \mathcal{S} \rightarrow \mathcal{A}$ mapping states to actions. Given a policy π , a value $Q_k^\pi(s, a)$ can be defined for each state-action pair with k remaining time steps, which represents the expected sum of discounted rewards for an agent following π for k steps, starting at s with action a :

$$Q_k^\pi(s, a) = E_\pi \left[\sum_{t=0}^k r_t \mid s_0 = s, a_0 = a \right]. \quad (60.9)$$

Accordingly, the value of each state for the fixed policy π with k remaining steps is

$$V_k^\pi(s) = \max_a Q_k^\pi(s, a). \quad (60.10)$$

The objective of the MDP problem is to obtain an optimal policy π^* that maximizes the expected cumulative reward for each state and time step:

$$\pi_k^*(s) = \operatorname{argmax}_{\pi} Q_k^\pi(s, \pi_k(s)). \quad (60.11)$$

For some domains, specifying the exact planning horizon is not feasible. Such problems are modeled using MDPs with infinite horizons represented by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. The definitions of all terms except γ remain the same as for finite horizon MDPs. $\gamma \in [0, 1)$ represents the discount factor associated with obtaining rewards in the future instead of now. The Q function corresponding to policy π for infinite horizon MDPs is

$$Q^\pi(s, a) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a \right]. \quad (60.12)$$

Notice that the subscript k was dropped from the definition as compared to Eq. 60.9. Similarly, the V function and the optimal policy π^* are changed to

$$V^\pi(s) = \max_a Q^\pi(s, a), \quad (60.13)$$

$$\pi^*(s) = \operatorname{argmax}_\pi Q^\pi(s, \pi(s)). \quad (60.14)$$

Although the distinction between finite and infinite horizon MDPs seems minor, the choice of problem formulation impacts the potential solution algorithms that can be used (e.g., policy trees cannot be expanded forever in the infinite horizon case, thus requiring an alternate representation). Further discussion of solution algorithms is provided later in Sect. 60.2.2.2.

In the presence of uncertainty, it is often useful to consider partially observable domains, where the state is not directly observable yet it can be inferred from observations of the environment. The partially observable MDP (POMDP) extends the previously described MDP framework to such domains where the global state cannot be necessarily uniquely determined given the information provided by the observations. The POMDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \Omega, \mathcal{O}, T \rangle$ or $\langle \mathcal{S}, \mathcal{A}, P, R, \Omega, \mathcal{O}, \gamma \rangle$ for the finite and infinite horizon cases, respectively, where the additional elements are:

- Ω : Defines a finite set of observations $o \in \Omega$, where o_t represents the observation at time t
- \mathcal{O} : Defines an observation function of the form $O(a, s', o) : \mathcal{A} \times \mathcal{S} \times \Omega \rightarrow [0, 1]$ specifying the probability of receiving observation o after executing action a and transitioning to state s'

In a POMDP, at step t , a *belief* vector $b_t \in \mathbf{R}^{|\mathcal{S}|}$ is maintained using inference which defines a probability distribution over the whole state space. Solving POMDPs is much more challenging as compared to their MDP counterparts because the optimal policy is now a mapping from the belief space to the action space (i.e., $\mathbf{R}^{|\mathcal{S}|} \rightarrow \mathcal{A}$) as opposed to $\mathcal{S} \rightarrow \mathcal{A}$. This increases the dimensionality of the problem leading to higher algorithm complexity for POMDP solvers.

Direct applications of the MDP and POMDP frameworks to cooperative multi-agent scenarios are referred to as multi-agent Markov decision processes (MMDP and MPOMDP). MMDPs are represented by essentially the same tuples as described above but with the addition of n to indicate the number of participating agents. As \mathcal{S} now represents the *joint* state space of all agents, \mathcal{A} the *joint* action space, and Ω the *joint* observation space, it follows that $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $o \in \Omega$ now denote *joint* states, actions, and observations which collectively represent the team's decisions. Formally, the actions and observations can be decomposed as:

- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$, where \mathcal{A}_i is the action space for agent i and each $a \in \mathcal{A}$ is a vector describing the joint action, $a = (a_1, \dots, a_n)$.

- $\Omega = \Omega_1 \times \dots \times \Omega_n$, where Ω_i is the observation space for agent i and $o \in \Omega$ describes a joint observation of the form $o = (o_1, \dots, o_n)$.

The joint state can involve complex coupling depending on the specific application, but, in some instances, it can be factored into individual agent components (see Sect. 60.4.2). Solving the MMDP involves computing an optimal joint policy $\pi^* = (\pi_1^*, \dots, \pi_n^*)$ where each π_i^* describes the individual policy for agent i . It is well known that the computational complexity of MMDPs scales exponentially in the number of agents making this a difficult problem to solve for large teams (Redding et al. 2012). Furthermore, a very limiting assumption of both MMDPs and MPOMDPs is that all agents must have identical knowledge of the global state. In the MMDP case, this is referred to as *individual full observability*, where each agent can observe the full joint state, and, similarly in multi-agent POMDPs, each agent is required to have identical partial knowledge of the joint state. This is usually an unrealistic assumption when distributed teams of cooperating agents are involved, where each agent is likely to have different local information, motivating decentralized frameworks which can handle varying situational awareness among agents. The next section describes solution algorithms for the centralized MDP and its variants, and more details on decentralized techniques are provided in Sect. 60.4.2.

60.2.2.2 Solution Algorithms

The primary goal for solving an MDP involves finding an optimal policy π^* as defined in Eqs. 60.11 and 60.14. In the finite horizon case, the most common approach is to use dynamic programming, which involves solving a backward recursion using a recursive form of the expected reward function, given by

$$\begin{aligned} Q_k^*(s, a) &= \sum_{s' \in \mathcal{S}} P(s, a, s') (R(s, a, s') + V_{k-1}^*(s')), \\ V_k^*(s) &= \max_a Q_k^*(s, a), \\ \pi_k^*(s) &= \operatorname{argmax}_a Q_k^*(s, a), \end{aligned}$$

where the terminal condition for the final time step is given by $V_0^*(s) = 0$. As a reminder, $V_k^*(s)$ specifies the value of the state given k remaining time steps; therefore, at time step $t = 0$, there are T remaining steps to go and the state values are given by $V_T^*(s)$. Likewise at the T th time step, there are 0 steps to go which is why the terminal condition is given by $V_0^*(s)$. Thus, k can be thought of as a backwards index in the recursion. The key insight here is that, at every step k , the expected cumulative return for step $k - 1$ can be lumped into one quantity; thus, by solving backward, one obtains the optimal policy for the desired time step. Notice that the optimal policy in the case of finite horizon MDPs is time dependent; thus, different time horizons result in different policies. This fact should be taken into consideration when selecting an appropriate time horizon for planning. The main benefit of dynamic programming versus forward search is that suboptimal branches

Algorithm 1: Policy Iteration**Input:** R, P, γ **Output:** π $\pi(s) \leftarrow \text{Random}(\mathcal{A})$ for $s \in \mathcal{S}$ changed $\leftarrow \text{True}$ **while** changed **do** $V^\pi \leftarrow (I - \gamma P^\pi)^{-1} P^\pi R^\pi$ **for** $s \in \mathcal{S}$ **do** $\pi^+(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s, a, s') [R(s, a, s') + \gamma V^\pi(s')]$ changed $\leftarrow (\pi^+ \neq \pi)$ $\pi \leftarrow \pi^+$ **return** π

of the policy tree are pruned early on, and thus less computation is required to find the optimal solution. For the infinite horizon case, an alternate formulation is required since there is no final step. In this case, the optimal policy can be found by iterating over two steps: (1) *policy evaluation* and (2) *policy improvement*. The policy evaluation step finds the value function for a given fixed policy. In particular, Eq. 60.13 can be written as

$$V^\pi(s) = \sum_{s' \in \mathcal{S}} P(s, \pi(s), s') (R(s, \pi(s), s') + \gamma V^\pi(s')), \quad (60.15)$$

$$\triangleq T^\pi(V^\pi(s)),$$

where T^π is the Bellman operator. Writing the above equation for all states in matrix form results in $V^\pi = T^\pi V^\pi = P^\pi R^\pi + \gamma P^\pi V^\pi$, where $P^\pi(i, j) = P(s_i, \pi(s_i), s_j)$ and $R^\pi(i, j) = R(s_i, \pi(s_i), s_j)$. The above equation can be solved in closed form as

$$V^\pi = (I - \gamma P^\pi)^{-1} P^\pi R^\pi, \quad (60.16)$$

where I is the identity matrix. Given the value function, the policy can be improved by a one-step lookahead update:

$$\pi(s) = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s, a, s') [R(s, a, s') + \gamma V^\pi(s')]. \quad (60.17)$$

Putting these two phases together (i.e., Eqs. 60.16 and 60.17 gives the policy iteration method shown in Algorithm 1. The output of the algorithm is guaranteed to be the optimal policy for the given MDP (Sutton and Barto 1998). Note that the algorithm requires $\Theta(|\mathcal{S}|^3)$ calculations per iteration to evaluate the policy (line 4) and $\Theta(|\mathcal{A}||\mathcal{S}|^2)$ to calculate the new policy (lines 5, 6). However, as long as the value function is getting closer to the exact solution, the policy evaluation/improvement loop still converges to the optimal solution (Sutton and Barto 1998). This idea

Algorithm 2: Value Iteration

Input: R, P, γ
Output: π
 $V(s) \leftarrow \text{Random}()$ **for** $s \in \mathcal{S}$
 changed \leftarrow **False**
repeat
 for $s \in \mathcal{S}$ **do**
 $v \leftarrow V(s)$
 $V(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s, a, s') [R(s, a, s') + \gamma V(s')]$
 $\pi(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s, a, s') [R(s, a, s') + \gamma V(s')]$
 changed \leftarrow changed **or** $v \neq V(s)$
until not changed
return π

leads to a process that saves substantial computation for finding a good policy because, as long as the ranking of actions in each state is correct, the result of policy improvement yields optimal action selections. To accommodate this idea, line 4 can be changed to the following update rule, known as the Bellman update:

$$V^\pi(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s, a, s') [R(s, a, s') + \gamma V^\pi(s')], \quad \forall s \in \mathcal{S}, \quad (60.18)$$

where the value of each state is improved by a one-step lookahead. This change produces the value iteration method shown in Algorithm 2. The resulting policy for this algorithm is also guaranteed to be optimal. Coping with computer precision limitations, practitioners often replace line 8 of Algorithm 2 with a checkpoint verifying that the maximum change applied to the value function is less than a small positive number (Sutton and Barto 1998). The value iteration algorithm eliminates the matrix inversion step that required $\Theta(|\mathcal{A}||\mathcal{S}|^2)$ computations per iteration. Note that the inner loop requires a loop over all states. Both policy iteration and value iteration are instances of dynamic programming methods.

For large state spaces, representing the value function using a lookup table is not feasible due to memory and computation restrictions. The next sections describe approximate DP methods applied to such domains by trading off solution quality for lower memory and computation requirements.

60.2.2.3 Linear Function Approximation

Holding a tabular representation of the Q function (i.e., storing a unique value for each state-action pair) is impractical for large state spaces. A common approach to coping with large state spaces is to use a linear approximation of the form $Q^\pi(s, a) = \theta^T \phi(s, a)$. The feature function $\phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^n$ maps each state-action pair to a size n vector of scalar values. Each element of the feature function $\phi(s, a)$ is called a *feature*; $\phi_f(s, a) = c \in \mathbb{R}$ denotes that feature f has scalar value c for state-action pair (s, a) , where $f \in \mathcal{X} = \{1, \dots, n\}$. \mathcal{X} represents the

set of features (e.g., being close to a wall and having low fuel or other properties of interest); the vector $\theta \in \mathbb{R}^n$ holds weights representing how important each feature is considered to be.

60.2.2.4 Approximate Dynamic Programming

Thus far, for the policy evaluation phase, V represented the value function for a fixed policy as a point in an $|\mathcal{S}|$ -dimensional space using $|\mathcal{S}|$ parameters (elements of the V vector). The idea of approximate DP (ADP) methods is to approximate the value function by representing it using a lower-dimensional space with $n \ll |\mathcal{S}|$ parameters. In particular, for the linear family of approximators explained in Sect. 60.2.2.3, the value function can be represented as $V \approx \tilde{V} \triangleq \Phi_{|\mathcal{S}| \times n} \theta_{n \times 1}$. When the value function is approximated using linear function approximation, the true value function might not lie in the space spanned by the feature function (i.e., column space of Φ). Hence, a metric is required to define the best approximate value function in the span of Φ . There are two major metrics used in the literature to define the best approximated value function (Bradtke and Barto 1996; Lagoudakis and Parr 2003; Farahmand et al. 2008; Sutton et al. 2009; Scherrer 2010), namely, (1) Bellman residual minimization (BRM) and (2) projected Bellman residual minimization, also known as least-squares temporal difference (LSTD). When the value of each state is being approximated, a weight vector is also required to highlight concerns with respect to the resulting error in each state. Intuitively, states that are visited more often should have higher weights, penalizing the error correspondingly. This intuition is captured through the *steady-state probability distribution* for a fixed policy, defined as a vector $\mathbf{d}_{1 \times n}$, where

$$\begin{aligned} \mathbf{d}P &= \mathbf{d}, \\ \text{s.t. } \|\mathbf{d}\|_{\ell_1} &= 1, \\ \forall i \in \{1, \dots, |\mathcal{S}|\}, \quad \mathbf{d}_i &\geq 0, \end{aligned} \tag{60.19}$$

where \mathbf{d}_i is the i th element of \mathbf{d} . Essentially, the steady-state probability distribution is the eigenvector of P^\top , representing a probability distribution over states that is invariant to the transition model (notice that the transition model P already includes the fixed policy π). For mathematical purposes, $\mathbf{D}_{|\mathcal{S}| \times |\mathcal{S}|}$ is defined as a matrix, with \mathbf{d} on its diagonal ($\mathbf{D} = \text{diag}(\mathbf{d})$).

Figure 60.2 depicts a geometric view of the optimal value function V and its projection into the span of Φ , using the orthogonal projection operator Π . Unfortunately, V is not known, but the value function which has zero Bellman error for all states (i.e., $\mathbf{T}(\tilde{V}) - \tilde{V} = \mathbf{0}$) is guaranteed to be optimal. Hence, both the BRM and LSTD approaches minimize a term defined with respect to the Bellman error with the desired goal that the resulting solution be close to ΠV . In particular, BRM minimizes the norm of the Bellman error shown as the solid green vector, while LSTD minimizes the projected Bellman error shown as the dashed blue line. The closed-form solution for BRM is

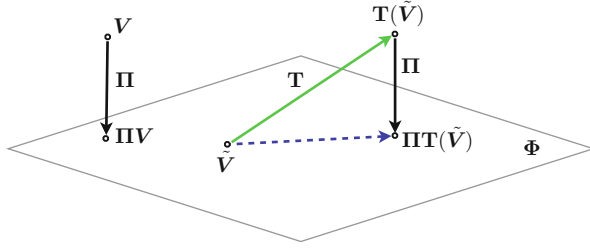


Fig. 60.2 A geometric interpretation of what Bellman residual minimization (*solid green*) and projected Bellman residual minimization (*dashed blue*) minimize, as described in Lagoudakis and Parr (2003). Π is the projection operator, mapping every point to its orthogonal projection on the span of Φ , shown as the 2-D plane. T is the Bellman operator

$$\begin{aligned} \theta &= \left[\underbrace{(\Phi - \gamma P \Phi)^\top D (\Phi - \gamma P \Phi)}_{A_{\text{BRM}}} \right]^{-1} \underbrace{(\Phi - \gamma P \Phi)^\top D R}_{b_{\text{BRM}}} \quad (60.20) \\ &= A_{\text{BRM}}^{-1} b_{\text{BRM}}. \end{aligned}$$

Note that the inverse of A_{BRM} always exists (Scherrer 2010). The closed-form solution for LSTD is given by

$$\begin{aligned} \theta &= \left[\underbrace{\Phi^\top D (\Phi - \gamma P \Phi)}_{A_{\text{LSTD}}} \right]^{-1} \underbrace{\Phi^\top D R}_{b_{\text{LSTD}}} \quad (60.21) \\ &= A_{\text{LSTD}}^{-1} b_{\text{LSTD}}. \end{aligned}$$

The LSTD solutions can be approximated using L_1 sampled transitions in the form of (s_i, a_i, r_i, s'_i) , where $s'_i = s_{i+1}$:

$$\tilde{\Phi} = \begin{bmatrix} \text{---} \phi^\top(s_1) \text{---} \\ \text{---} \phi^\top(s_2) \text{---} \\ \vdots \\ \text{---} \phi^\top(s_{L_1}) \text{---} \end{bmatrix}, \tilde{P}\tilde{\Phi} = \begin{bmatrix} \text{---} \phi^\top(s'_1) \text{---} \\ \text{---} \phi^\top(s'_2) \text{---} \\ \vdots \\ \text{---} \phi^\top(s'_{L_1}) \text{---} \end{bmatrix}, \tilde{R} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{L_1} \end{bmatrix}, \quad (60.22)$$

$$\tilde{A}_{\text{LSTD}} = \frac{1}{L_1} \tilde{\Phi}^\top (\tilde{\Phi} - \gamma \tilde{P}\tilde{\Phi}), \quad (60.23)$$

$$\tilde{b}_{\text{LSTD}} = \frac{1}{L_1} \tilde{\Phi}^\top \tilde{R}, \quad (60.24)$$

$$\theta = \tilde{A}_{\text{LSTD}}^{-1} \tilde{b}_{\text{LSTD}}. \quad (60.25)$$

In the limit of infinite samples, approximations become exact provided that the sampling policy is ergodic (Lagoudakis and Parr 2003). However, extending the BRM solution to the sample-based case is problematic. In particular, for calculating an unbiased estimate of A_{BRM} , for each pair of s_i and a_i , two next states should be sampled independently. This observation makes the BRM approach hard for domains with limited available samples. Once the new value function is calculated, a new policy can be obtained using Eq. 60.17.

Least-squares policy iteration (LSPI) (Lagoudakis and Parr 2003) is an algorithm based on LSTD evaluation and policy improvement. A key property of the LSPI algorithm is that it does not require access to the model of the MDP in order to solve it. Moreover, samples are gathered through a trajectory of experience. Such a framework is known as reinforcement learning (RL) and is widely used throughout the literature. A simple RL solver algorithm is SARSA (state, action, reward, state, action) (Rummery and Niranjan 1994), in which, given a tuple of experience at time $t + 1$ in the form of $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$, the weights are updated as

$$\theta = \theta + \alpha [r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \phi(s_t, a_t), \quad (60.26)$$

where α is the learning rate. The policy at each step is calculated as

$$\pi(s) = \begin{cases} \operatorname{argmax}_a Q(s, a) & \text{with probability } 1 - \epsilon \\ \text{Random}(\mathcal{A}) & \text{with probability } \epsilon \end{cases} \quad (60.27)$$

to satisfy ergodicity (i.e., all states will be visited infinitely often in the limit). It has been shown that, for a fixed policy, right step size parameters, and any linear function approximation, applying Eq. 60.26 will converge to the LSTD solution asymptotically (Tsitsiklis and Roy 1997). There are a vast set of advanced DP and RL methods in the literature (interested readers are referred to Sutton and Barto (1998), Buşoniu et al. (2010), and Bellman (2003)). As expected, solving POMDPs is much more difficult as compared to solving MDPs. As a result, the area for solving real-world POMDP problems has been barely touched upon (Spaan and Vlassis 2005; Zhou and Hansen 2001; Paquet et al. 2005).

60.2.3 Game Theory

The field of game theory presents an alternate way of addressing the multi-agent planning problem by treating the interaction between agents as a game. The basic idea behind game theory is that agents are individual decision-making entities that perform actions to maximize their own local utility based on knowledge of other agents and the environment. As such, game-theoretic frameworks lend themselves naturally to solving autonomous task allocation problems in a decentralized fashion,

motivating significant work in this area (Arslan et al. 2007; Marden et al. 2009; Chapman et al. 2010; Marden and Wierman 2009). Since in game theory agents make individual decisions about their own actions, these frameworks are useful for modeling noncooperative environments; however, enforcing cooperation is difficult because it involves ensuring that individual agent utility functions and incentives are aligned with the global mission goals. Therefore, the main challenge associated with game-theoretic cooperative planning strategies involves designing proper utility functions and negotiation strategies to ensure appropriate levels of coordination and collaboration between agents in order to maximize global mission performance. This section describes current game-theoretic approaches to cooperative multi-agent planning problems, including the design and selection of utility functions and negotiation strategies, and highlights the major associated challenges and limitations.

60.2.3.1 Problem Statement

In the game-theoretic literature, a commonly studied cooperative multi-agent problem is the *vehicle-target assignment problem* (Murphey 1999), which is very similar in nature to the task allocation problem described in Sect. 60.2.1. The vehicle-target assignment problem consists of assigning a set of N_a vehicles to a set of N_t targets with the object of maximizing the global utility. The vehicles may have varying characteristics and the targets different reward values. Using a game-theoretic formulation leads to a decentralized implementation framework where the vehicles are considered self-interested decision-making agents intent on maximizing their own local utility (Arslan et al. 2007). It is desirable to find an “agreeable assignment” (Arslan et al. 2007) where no agent has an incentive to unilaterally deviate (pure-strategy Nash equilibrium (Fudenberg and Tirole 1991)) and to design the agent utilities such that maximizing the local utilities corresponds to maximizing the global utility.

The problem formulation consists of selecting assignments for a set of vehicles $\mathcal{I} \triangleq \{1, \dots, N_a\}$ given a set of targets $\mathcal{J} \triangleq \{1, \dots, N_t\}$. A vehicle’s assignment or action, $a_i \in A_i$, consists of choosing a target from an allowable subset of targets, $A_i \subset \{\mathcal{J} \cup \emptyset\}$, where the action \emptyset is a “null” target or an empty assignment and the total action space for all vehicles is $A \triangleq A_1 \times \dots \times A_{N_a}$. A joint assignment profile $a \in A$ consists of a vector describing the actions of all agents, $a = (a_1, \dots, a_{N_a})$, and is sometimes written as $a = (a_i, a_{-i})$ to highlight agent i ’s action given the actions of the other agents, a_{-i} . Each agent wishes to maximize its own local utility such that for vehicle i ,

$$a_i^* = \operatorname{argmax}_{a_i \in A_i} U_i(a_i, a_{-i}), \quad (60.28)$$

where a_i is vehicle i ’s assignment and a_{-i} is the assignment profile of the other vehicles. Solving these for each agent gives a joint assignment profile $a \in A$ corresponding to a global mission utility $U_g(a)$. If every target has a reward value $R_j(a)$ as a function of the joint assignment profile, then the global utility is the total reward over all targets and is written as

$$U_g(a) = \sum_{j=1}^{N_t} R_j(a). \quad (60.29)$$

In game theory, a framework where maximizing local agent utilities corresponds to maximizing the global utility is referred to as a *potential game* (Monderer and Shapley 1996), and several recent approaches have considered fitting the vehicle-target assignment problem within this architecture (Arslan et al. 2007; Marden et al. 2009; Chapman et al. 2010; Marden and Wierman 2009). A potential game consists of a game where there exists a potential function $\phi(a): A \rightarrow \mathbb{R}$ such that for every vehicle $i \in \mathcal{I}$, for every $a_{-i} \in A_{-i}$, and for every $a'_i, a''_i \in A_i$, the local utility functions satisfy

$$U_i(a'_i, a_{-i}) - U_i(a''_i, a_{-i}) = \phi(a'_i, a_{-i}) - \phi(a''_i, a_{-i}), \quad (60.30)$$

where the potential function is representative of the global utility.

Potential games have several useful properties that can be exploited to obtain near-optimal solutions to the global problem (the reader is referred to Monderer and Shapley (1996)). The most important of these is that potential games are guaranteed to have at least one pure-strategy Nash equilibrium (PSNE), which, in the vehicle-target assignment problem, corresponds to a valid assignment profile where no vehicle has any incentive to deviate from its strategy given the other vehicles' strategies. Several multiplayer learning algorithms that are guaranteed to converge to a PSNE have been recently developed (e.g., fictitious play, spatial adaptive play, and regret matching) (Arslan et al. 2007). However, the main issue with the potential game framework is that for most multi-agent problems, there usually exist many PSNE, and while several current algorithms guarantee convergence to a PSNE, they typically do not find the optimal PSNE (and in practice can produce arbitrarily bad assignment profiles). To mitigate this problem, it is necessary to carefully design the vehicle utilities and the negotiation strategies such that the resulting algorithm converges to a near-optimal PSNE, a challenging task that has generated much interest in the research community (Arslan et al. 2007; Monderer and Shapley 1996).

Designing local utility functions to align agent utility with global utility is difficult, since consideration must be given to what information agents have access to and how much they need to communicate with each other in order to properly represent the mission goals. An obvious choice to align agent utilities with the global utility is the *identical interest utility*, where each vehicle's utility is exactly equal to the global mission utility. Unfortunately, this requires that each vehicle have complete knowledge about the global state, including other vehicles' actions, all target reward functions including targets that are not of interest to the specific agents, and all environmental parameters that may affect any agent. Given that autonomous networked teams typically have limited bandwidth and communication constraints, it is desirable to find a utility function structure that keeps the amount of global information an agent requires to a minimum while still aligning the agent's local utility with the global utility. Better choices for localized utility functions that

have been considered are the *range-restricted utility*, the *equally shared utility*, and the *wonderful life utility* (see Arslan et al. (2007), Monderer and Shapley (1996), and Tumer and Wolpert (2004) for further details). The most popular of these is the wonderful life utility (WLU) due to its computational simplicity and favorable properties. The WLU is defined as

$$U_i(a_i, a_{-i}) = R_j(a_i, a_{-i}) - R_j(\emptyset, a_{-i}) \quad (60.31)$$

for $a_i = j$ and represents the marginal utility obtained by an agent for doing task j given the assignments of the other vehicles. The WLU has the useful feature that the local vehicle utility is equal to the marginal contribution made to the global utility, leading to a potential game with potential function equal to the global utility ($\phi(a) = U_g(a)$). The next section discusses solution algorithms for potential games that ensure convergence to near-optimal equilibria.

60.2.3.2 Solution Algorithms

As mentioned before, depending on the negotiation strategy, stable outcomes of most potential game algorithms might result in arbitrarily bad pure-strategy Nash equilibria (PSNE), motivating the design of negotiation strategies that increase the probability of convergence to a near-optimal PSNE. Furthermore, it is desirable for vehicles to negotiate with each other at every time step given all agents' assignments only, without requiring internal knowledge about other vehicles' utility functions or internal state. Several current negotiation strategies are reviewed in Arslan et al. (2007), including algorithms such as action-based fictitious play, utility-based fictitious play, regret matching (and variants), and spatial adaptive play, and the benefits and drawbacks of each are described.

One negotiation strategy that ensures convergence to a near-optimal PSNE is the spatial adaptive play (SAP) algorithm. The SAP algorithm is described as follows: at stage k , a vehicle is randomly chosen from the set of vehicles (e.g., vehicle i). Vehicle i then proposes a target according to probability distribution $p_i(k)$ that maximizes

$$p_i(k) \left(\begin{bmatrix} U_i(1, a_{-i}(k-1)) \\ \vdots \\ U_i(N_t, a_{-i}(k-1)) \end{bmatrix} \right) + \tau \mathcal{H}[p_i(k)], \quad (60.32)$$

where $\mathcal{H}(\cdot)$ is the entropy function which represents the level of randomization and the parameter τ sets the reward for randomization. The closed-form solution for the maximizing probability distribution $p_i(k)$ is then

$$p_i(k) = \sigma \left(\frac{1}{\tau} \begin{bmatrix} U_i(1, a_{-i}(k-1)) \\ \vdots \\ U_i(N_t, a_{-i}(k-1)) \end{bmatrix} \right), \quad \sigma(\mathbf{x})_i = \frac{e^{x_i}}{e^{x_1} + \dots + e^{x_n}}. \quad (60.33)$$

where $\sigma(\cdot)$ is the soft-max function. The SAP algorithm converges to a near-optimal PSNE with arbitrarily high probability which is tunable by the user through the parameter τ (Arslan et al. 2007). Furthermore, the computational overhead required to run SAP is low, making real-time implementations feasible.

Several extensions to the baseline game-theoretic task assignment formulation have been proposed in the literature. In Marden et al. (2009), weakly acyclic games are introduced to account for time-varying objective functions and action sets, as well as a restricted-action set variant of SAP termed binary restricted SAP. In Marden and Wierman (2009), it has been shown that noncooperative control formulations make it impossible to design budget-balanced agent utilities that guarantee that the optimal control is a PSNE. The proposed solution to address this issue consists of designing a modified agent utility that is conditioned on additional state information and recasting the problem to a stochastic state-based game framework. Finally, to address the problem of dynamic task allocation with task deadlines, Chapman et al. (2010) present a stochastic game formulation approximated by a series of overlapping potential games. The resulting overlapping potential game approximation (OPGA) is shown to be solvable in a distributed manner and is robust to communication restrictions.

60.3 Planning Architectures

There are many important aspects to consider when deciding what type of planning algorithm is most appropriate for a given application. An important observation is to note that, for most problems of interest, the planners being discussed in this chapter will only be able to approximately solve the problem. This is because for most applications, (1) optimality is computationally infeasible, even in the most efficient computation structures (Bertsimas and Weismantel 2005), and (2) the optimal allocation will only be accurate with respect to the objective function, which is often necessarily approximate.

This section introduces the idea that communication between algorithmic modules is a fundamental aspect of partitioning classes of algorithms and will formalize definitions for three communication architectures: centralized, distributed, and decentralized. It is then shown how each of these architectures place constraints on whether the algorithm can run synchronously or not, which may affect overall algorithmic performance. Finally, this section introduces different methods for handling coordination and cooperation in the assignment space, discusses what types of information should be shared among agents, and explains how these choices affect the performance of different algorithms in their respective environments.

The partition introduced in this section is not new, and similar insights have been presented in the literature (Shima and Rasmussen 2009; Smith and Davis 1981). In (Shima and Rasmussen 2009), the authors partition the design space into classes of algorithms according to three factors: message cost, distribution of computation, and severity of task coupling. Each of these three issues is roughly subsumed into the categories presented in this section. Earlier work described in

Smith and Davis (1981) stressed the difference between distributed problem solving (where decisions are made in a distributed fashion) and distributed processing (where data is processed in a distributed way, but decisions are still made in a relatively centralized way), and the term decentralized is used to describe systems that utilize both distributed problem solving and distributed processing. This section redefines these terms to more heavily rely on the communication aspects of the problem; however, the classes of algorithms contained in Smith and Davis (1981) remain consistent with the definitions proposed in this section. Furthermore, Smith and Davis (1981) also introduce a distinction in the level of cooperation required between what is referred to as task-sharing (where the module sub-problems are assumed to be relatively separable) and results-sharing (which requires more intense communication), and this distinction is also mirrored in the discussion on handling cooperation in the assignment space presented at the end of this section.

60.3.1 Centralized

A centralized computation model refers to an algorithm running on a single machine, and where information passing between modules of the algorithm is done through shared memory. Fast centralized algorithms are often implemented using parallelized computation structures that can take advantage of the large number of cores available in modern computer systems. Centralized implementations can also be quite fast for algorithms that are significantly parallelizable, since each module in the algorithm has access to the current global algorithmic state almost instantaneously. This means that there is very little *communication cost* between the modules (communication cost includes both the extra resources needed to facilitate communication and the associated delays introduced). Despite this, some environments may not be ideal for centralized approaches. The following list describes four environments where centralized computation may not be advisable:

1. If large amounts of data have to be transmitted to the centralized solver's location, a centralized architecture may not be ideal. Other architectures could allow processing of the data remotely to identify the "usefulness" of information prior to sending metadata to a centralized location (referred to as distributed processing in Smith and Davis (1981)). Additionally, more autonomous implementations could use this remote data to actually produce candidate assignments for these remote agents without ever communicating with a centralized location (referred to as distributed problem solving in Smith and Davis (1981)), thus creating localized cooperative assignments.
2. The solution speed of centralized solvers is throttled by the rate at which the important information reaches the computation structure(s). If communications are slow, unreliable, or expensive, it becomes harder to justify passing large amounts of possibly irrelevant data through the network.
3. If most of the information needed by each agent to create its cooperative assignments is local, much faster reaction times might be obtained by keeping

all of the computations local as well and sharing only the results of these local computations with other agents through communication.

4. In an abstract sense, distributing the computation on multiple machines might also be preferred if there are very different types of calculations that need to be performed. If, for example, the computation has large portions that can be sped up using an optimized computational structure such as a GPU, it would be desirable to outsource these computations to more appropriate hardware.

If the above criteria are not a concern for a particular application, centralized parallel algorithms are likely to perform quite well.

60.3.2 Distributed

This section defines distributed algorithms as algorithms consisting of separate modules that are able to utilize reliable message passing. Distributed systems can be implemented on a single machine or on separate machines. In a distributed system, this communication between the agents can be either passed over a network or implemented as internal memory passing, but the aspect that defines the algorithm as distributed computation (as opposed to centralized computation) is that it utilizes separate memory partitions between the distributed algorithmic pieces. This introduces an extra layer of complexity over centralized solutions because shared information must be transmitted through a communication channel. Given this description, the defining characteristic of algorithms that are designed to operate in distributed environments is that *strong* message-passing channels are assumed to exist. A connection is defined as strong if each distributed node has knowledge of the existence of all other nodes it is able to communicate with, knows how to transmit messages to each of these nodes, has low message latency between each of these nodes, and has guarantees that messages will arrive reliably.

One application of distributed algorithms is if the computational load is too immense for a single (possibly multi-core) machine. This strategy would utilize some number of machines communicating through messages in a local and reliable way. The trade-off to consider when moving from centralized to distributed computation strategies is that time lost in message communication must be offset by the additional computation available by using multiple machines. Distributed algorithms also perform well (as compared to centralized algorithms) when information is being acquired remotely and external processing can be utilized before metadata is transmitted to other modules. This is because the overall communication load on the system can be reduced dramatically for the same amount of computation. For example, in cooperative planning environments, this could involve an agent observing a local event, changing its own plan based on this newly observed information, and then reliably communicating the results to the rest of the distributed modules. Again, the most important aspect of these distributed algorithms is that they rely on stable communications and depend on sharing information among other modules reliably. If communication links are not sufficiently reliable, performance may degrade significantly, and decentralized assumptions may be more appropriate.

60.3.3 Decentralized

Decentralized architectures are most useful in systems where information sharing is performed exclusively through messages (e.g., separate agents linked via a communication network but with no direct memory sharing). Decentralized algorithms are designed for environments where there are no rigid constraints placed on message delays, network connectivity, program execution rates, and message arrival reliability. This means that decentralized algorithms are the most robust to catastrophic changes in the communication environment. The price of this robustness may be conservative performance when communication conditions are actually favorable, but for truly decentralized environments, this trade-off may be preferable. Fully decentralized algorithms allow a high degree of autonomy for the sparsely connected agents. This is because decentralized architectures do not place rigid rules on how agents must interact; therefore, there is flexibility on how individual agents can make decisions. Given that communications between agents may be weak, decentralized algorithms could allow large teams to interact efficiently, without bogging the entire network down with an overly restrictive infrastructure. As discussed below in Sect. 60.3.4, decentralized architectures are also able to handle asynchronous communications much more naturally than distributed and centralized approaches.

60.3.4 Synchronous Versus Asynchronous

Parallelized computation can be used to increase the performance in cooperative assignment algorithms. Given this, choices can be made about how to organize this parallel computation: there are highly structured, synchronous algorithms that enforce constraints on when computation can be done or more flexible structures that utilize asynchronous computation. Algorithms that use *synchronous* computation define rigid rules about when computation can happen. Often, synchronization takes the form of queuing up computation that may only be executed after certain event driven triggers. This is to ensure that the algorithm has predictable state during program execution. Synchronization is heavily utilized in most iterative algorithms. In these synchronous iterative algorithms, a set of modules often perform parallel computations, share state variables, and then wait until a certain trigger before computing the next iteration of the algorithm. In this process, guarantees can be made about the state of each agent. Furthermore, these synchronous structures are also useful because they allow algorithms to make assumptions about the information state of other modules. The cost of using synchronous algorithms is that, in some environments, it may take significant effort to enforce synchronous behavior. In distributed and decentralized algorithms, the computation triggers are often nonlocal and thus may need to be inferred through inter-module communication. In the literature, the effort involved in enforcing this synchronization has been referred to as the synchronization penalty (Bertsekas and Tsitsiklis 1989). In centralized approaches,

the mechanisms required to enforce synchronization are fairly lightweight, but when information is being shared across a network and modules must spend time waiting for messages to arrive from physically separated machines, this penalty can become severe. Conversely, *asynchronous* computation is used when modules of the algorithm can run relatively independently of one another. Because of this, these structures work well in decentralized algorithms where they can utilize information whenever available and not on a rigid schedule. This introduces a fundamental trade-off between the rigidity of forcing an algorithm to run on a synchronous schedule versus allowing the algorithm to run asynchronously but take a performance hit for losing information assumptions about the state of the other modules. Typically, in centralized and distributed environments, the synchronization penalty may be weak enough that it makes sense to utilize synchronous computation, whereas in decentralized approaches, the synchronization penalty can become crippling, and asynchronous computation often becomes necessary.

60.3.5 Performance Metrics

Once the computational structures of the algorithm are chosen, there are several metrics that the designer will care about during the run time of the algorithm. Depending on the relative importance of each of the following metrics, different algorithms may be preferred:

- *Score performance* specifically refers to how important it is that the final solution optimizes the global cost function. There can be cases where significant coupling exists in the score function, increasing the difficulty of finding a good plan. In these problems, it may be desirable for an algorithm to have strong performance guarantees with respect to the optimal. Conversely, in other scenarios it may be easy to find near-optimal allocations and other considerations may be more important.
- *Run time* specifically refers to how much computational time the entire algorithm takes to complete, and, as will be contrasted below, this is typically thought of as a global measure for the entire team. For off-line solutions, acceptable measures of run time can be in the hours (or possibly days). When algorithms are run in the field, run time is typically thought of in the seconds range.
- *Convergence detection* can be a difficult thing to quantify in general. It is usually trivial to post process when convergence occurred, but in some cases (especially decentralized implementations), it may be difficult to determine if the algorithm has actually converged. This leads to different definitions of what convergence should be. Does the algorithm require global convergence? Does it just require local convergence (only the individual module)? Does the algorithm require only partial convergence (in some cases, being sure about the next task to execute is enough)? The answers to these questions will impact what type of information is communicated as well as how much total communication is actually needed.
- *Convergence time* is typically identical to run time in terms of global convergence metrics, but for local convergence or partial convergence metrics, it measures the

time until this local convergence is detected. With convergence time, typically smaller is better, and the smaller this number is, the faster the planner can react to dynamic changes in the environment. In very dynamic environments, convergence time may be one of the most important metrics for the mission designer.

- *Reaction time to situational awareness* is closely related to convergence time. It looks specifically at the turn around time between acquiring information and incorporating it into the plan cycle. In some algorithms, information cannot be incorporated mid-convergence (while others allow for the inclusion of new information at any time). In designing algorithms, trade-offs will need to be made between convergence time and score performance to enable faster reaction times to situational awareness.

60.3.6 Coordination Techniques

Coupling between the goals of distributed agents in decentralized environments is typically required for most missions. Because of this, the assignment variables that specify mission behavior can contain nontrivial constraints between them. As a result, cooperative allocation algorithms must be able to produce consistent values for these coupled variables. In centralized environments, the problem of checking that constraints between modules are satisfied can be handled trivially by using shared memory. In distributed and decentralized algorithms, the problem becomes more complex because the modules are communicating using messages. There are three main types of coordination approaches to consider:

1. One technique is to create an a priori partition in the assignment space, so each distributed agent can trivially generate conflict-free plans.
2. A second technique, commonly known as *implicit coordination*, involves performing situational awareness (SA) consensus (i.e., consensus on all variables that are not assignments) and hope that, given sufficiently close information, the agents are able to independently create conflict-free allocations.
3. The last technique is to directly incorporate the constraint feasibility of the allocations into the convergence process of the planning algorithm.

In general, cooperative algorithms may use combinations of the three techniques listed above. The following discusses the main factors associated with determining which of these techniques is more appropriate given the application at hand.

1. Creating an a priori partition in the task environment effectively chops up the task space and only allows each agent to bid on a subset of the overall task set. Given a higher-level task allocation or a human operator, it may be reasonable to use this method for small teams in relatively static environments. This is because, for some environments, it may be obvious what agents should be servicing which tasks. However, in environments with large numbers of relatively homogeneous agents, or in dynamic environments, the task partitioning problem can become very complicated. By creating this partition, the algorithm is placing artificial

constraints on which allocations are available, which may result in arbitrarily poor task assignments.

2. The main emphasis of *implicit coordination* is to arrive at *situational awareness consensus* (where situational awareness consensus refers to the process by which all agents agree on all variables relevant to the initial conditions of the task allocation problem) before starting the planning algorithm, then run effectively centralized planners on each agent independently. The basic premise of this method is that a consensus algorithm can generate consistent parameter values across an entire team, in which case each agent will generate identical final allocations. This method has been explored in the literature (Ren et al. 2005, 2007; Ren and Beard 2005; Olfati-Saber and Murray 2004; Alighanbari and How 2008a; Moallemi and Roy 2006; Olshevsky and Tsitsiklis 2006; Hatano and Mesbahi 2005; Wu 2006; Tahbaz-Salehi and Jadbabaie 2006) and is widely used because it is a relatively straightforward way to decentralize a task allocation algorithm. The benefit of using implicit coordination over task space partitioning (see 1 above) is that by not limiting the assignment space a priori, the algorithm is able to account for changes in information explicitly (e.g., dynamic tasks and additional state information). Using task space partitioning, any changes in information are likely to necessitate a repartitioning and redistribution of the task space, and if this is not executed, the resulting assignment is likely to perform poorly. In contrast, with implicit coordination, agents explicitly account for information changes through the situational awareness consensus protocol every time they replan, thus producing more relevant higher-performing assignments. Implicit coordination also has the added benefit that if the task environment is highly coupled (with many agent-to-agent and task-to-task constraints), then the plans produced are able to recognize and exploit this structure, thus producing assignments that include highly coordinated behaviors between the agents.

All of these benefits come with the caveat that the situational awareness consensus must happen before the planner can start producing plans, and, in order to guarantee that agents produce the same assignments, this consensus process may require large amounts of communication bandwidth (Alighanbari and How 2008a). A potential difficulty with algorithms that utilize this implicit coordination strategy is that they often require the planning parameters to be known precisely, thus requiring that the consensus process be conducted until errors in situational awareness become very small. In fact, it is often the case that, if the final estimates of the planning parameters do not converge to within arbitrarily tight bounds, there is no *guarantee* that the resulting assignment will be conflict free.

3. The third technique ignores gaining any insight about the situational awareness of other agents and only requires that assignments between the team remain conflict free. The power of this approach is that all of the consensus effort is spent on achieving consistent values for the agent assignments. This solution is preferable if there are few inter-agent constraints and inter-task constraints (the more coupled the task environment becomes, the more difficult the task consensus problem becomes) or if the communication environment is not necessarily reliable such that it would be difficult to reach complete team-wide consistent situational awareness. For many decentralized applications, the primary goal is achieving a conflict-free distribution

of tasks that performs well, rather than requiring intense cooperation between agents to achieve the best possible assignment (such as the CBBA algorithm discussed in Sect. 60.4.1.1). Given that in these environments the communication links are often non-robust, especially across larger distances, only broadcasting information directly relevant to the assignment constraints may be preferable. There is a trade-off between implicit coordination and algorithms using assignment consensus. The choice is really between deciding if it is easier to converge on a consistent assignment vector or converge to arbitrarily tight bounds on all other significant state variables.

60.3.7 Consensus

As discussed above, a key component of cooperative decision making involves performing *consensus* among agents, which is defined as reaching an agreement on quantities of interest, such as plans, situational awareness, or other desired parameters. Most distributed planning approaches employ *consensus algorithms*, which are sets of rules, or protocols, that determine how information is exchanged between agents to ensure that the team will converge on the parameters of interest.

As a simple illustrative example, the following linear consensus protocol can be used to converge on a continuous parameter z :

$$\begin{aligned}\dot{x}_i(t) &= \sum_{j \in N_i} (x_j(t) - x_i(t)) \quad \forall i, \\ x_i(0) &= z_i \in \mathbb{R},\end{aligned}\tag{60.34}$$

where each agent i computes errors with its set of neighbors N_i and uses these to correct its parameter estimate (Olfati-Saber et al. 2007). Collectively, the team dynamics for n agents can be written as an n th order linear system:

$$\dot{\mathbf{x}}(t) = -\mathcal{L}\mathbf{x}(t),\tag{60.35}$$

where $\mathcal{L} = D - A$ is known as the graph Laplacian, which is computed using an adjacency matrix A describing connectivity between agents (the elements a_{ij} are 1 if j is a neighbor of i and 0 otherwise) and a degree matrix $D = \text{diag}(d_1, \dots, d_n)$, with elements $d_i = \sum_{j \in N_i} a_{ij}$. The maximum degree, denoted as $\Delta = \max_i d_i$, is useful in bounding the eigenvalues of \mathcal{L} , which, for an undirected connected network, can be ordered sequentially as

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2\Delta.\tag{60.36}$$

The eigenvalues of \mathcal{L} can be used to predict convergence rates and stability properties of these linear consensus algorithms (in particular, λ_2 is related to speed of convergence, and λ_n provides stability bounds in time-delayed networks).

As shown above, the nontrivial eigenvalues are all positive, and exploiting the fact that Eq. 60.35 describes a linear system, the consensus algorithm is globally asymptotically stable and converges exponentially to an equilibrium with rate given by λ_2 (Olfati-Saber et al. 2007). Furthermore, for the system described above, the algorithm is guaranteed to achieve a unique equilibrium $\bar{z} = 1/n \sum_i z_i$, which is the average of all the agents' initial values.

Recent research has explored the effects of more realistic mission environments on these types of linear consensus algorithms for multi-agent teams. Some examples include analyzing the impact of time-delayed messages and dynamic network topologies on convergence and stability properties of the consensus algorithms. The work in Olfati-Saber et al. (2007) shows that global exponential convergence guarantees can be extended to dynamic networks as long as the network remains connected at each time t . The agents are guaranteed to reach consensus with convergence rate greater than or equal to $\lambda_2^* = \min_t \lambda_2(G(t))$, where $\lambda_2(G(t))$ is the second eigenvalue of the Laplacian for graph $G(t)$. Similar guarantees can be made for time-delayed networks, where messages are received after a delay τ instead of instantaneously. The system dynamics can be modified as follows:

$$\dot{\mathbf{x}}(t) = -\mathcal{L}\mathbf{x}(t - \tau), \quad (60.37)$$

and global exponential convergence guarantees are retained for delays within the range $\tau < \pi/2\lambda_n$. Note that convergence rates and robustness to time-delays can be improved by actively controlling the network structure (modifying G and \mathcal{L}), which is an active area of research (Ren and Beard 2005; Ren et al. 2007; Jadbabaie et al. 2003; Blondel et al. 2005; Olfati-Saber et al. 2007).

Consensus algorithms have been applied to a wide variety of distributed decision-making applications, ranging from flocking to rendezvous (Jadbabaie et al. 2003; Beard and Stepanyan 2003; Lin et al. 2003; Fax and Murray 2004; Ren 2006; Ren et al. 2007; Olfati-Saber et al. 2007). Most of these consensus algorithms are computationally inexpensive and guarantee convergence of the situational awareness, even over large, complex, and dynamic network topologies (Hatano and Mesbahi 2005; Wu 2006; Tahbaz-Salehi and Jadbabaie 2006). A common issue with classical consensus algorithms is that agents' observations are often treated with equal weight, whereas in reality, some agents may have more precise information than others. Extending classical consensus algorithms to account for this uncertainty in local information, Kalman consensus approaches have been developed that approximate the inherent uncertainty in each agent's observations using Gaussian distributions (Ren et al. 2005; Alighanbari and How 2008b). These algorithms produce consensus results that are more heavily influenced by agents with smaller covariance (therefore higher certainty) in their estimates. A limitation of Kalman consensus approaches is that Gaussian approximations may not be well suited to model systems with arbitrary noise characteristics, and applying Kalman filter-based consensus methods to the mean and covariance of other distributions can sometimes produce biased steady-state estimation results (Fraser et al. 2009).

Other Bayesian decentralized data and sensor fusion methods have been explored to determine the best combined Bayesian parameter estimates given a set of observations (Waltz and Llinas 1990; Grime and Durrant-Whyte 1994; Makarenko and Durrant-Whyte 2006). A major challenge, however, is that these decentralized data fusion approaches require *channel filters* to handle common or repeated information in messages between neighboring nodes. These channel filters are difficult to design for arbitrary network structures, and generic channel filter algorithms have not been developed other than for simple network structures (e.g., fully connected and tree networks), thus limiting the applicability of decentralized data fusion methods (Grime and Durrant-Whyte 1994). Recent work has addressed this issue by showing that, through a combination of traditional consensus-based communication protocols and decentralized data fusion information updates, scalable representative information fusion results can be achieved, without requiring complex channel filters or specific network topologies (Olfati-saber 2006; Xiao et al. 2005; Fraser et al. 2009, 2012). In particular, the work in Olfati-saber (2006) utilized dynamic-average consensus filters to achieve an approximate distributed Kalman filter, while (Xiao et al. 2005) implemented a linear consensus protocol on the parameters of the information form of the Kalman filter, permitting agents to execute a Bayesian fusion of normally distributed random variables. However, as previously noted, these Kalman-based methods are derived specifically for normally distributed uncertainties (Olfati-saber 2006; Xiao et al. 2005), and thus can produce biased results if the local distributions are non-Gaussian. Recent work has extended these combined filtering and data fusion approaches to allow networked agents to agree on the Bayesian fusion of their local uncertain estimates under a range of non-Gaussian distributions (Fraser et al. 2012). In particular, the approach exploits *conjugacy* of probability distributions (Gelman et al. 2004) and can handle several different types of conjugate distributions including members of the exponential family (e.g., Dirichlet, gamma, and normal distributions) (Fraser et al. 2009, 2012). The approach in Fraser et al. (2012) is termed *hyperparameter consensus* and has demonstrated flexibility in handling several realistic scenarios, including ongoing measurements and a broad range of network topologies, without the need for complex channel filters.

60.4 Decentralized Algorithms

This section reviews current decentralized approaches proposed in the literature, describing the benefits, challenges, and limitations of each.

60.4.1 Market-Based Approaches

Market-based algorithms are integer programming-based formulations that are able to solve distributed and decentralized cooperative assignment problems (Dias and Stentz 2000; Dias et al. 2006). Many successful market-based approaches use an

auction mechanism (Bertsekas 1989, 2007). Auctions may be run via an auctioneer (Gerkey and Mataric 2002; Mataric et al. 2003; Gerkey and Mataric 2004), where each agent computes the reward associated with a task based on their own local understanding of the world and then uses this reward value to place *bids* on the most desirable tasks. Each agent communicates their bid to the auctioneer, and the auctioneer determines the winner for each task. These types of methods guarantee conflict-free solutions since the auctioneer only selects one agent as the winner. They are distributed because the bids are calculated at spatially separated locations, but they are synchronous in that they rely on one winner at every iteration.

Some auction-based protocols do not need to designate a single agent as the auctioneer but utilize a different protocol where the winner is decided based on a set of self-consistent rules (Castanon and Wu 2003). Other such methods, called distributed auction algorithms (Sariel and Balch 2005; Ahmed et al. 2005; Atkinson 2004; Lemaire et al. 2004), have been shown to efficiently produce suboptimal solutions when reward functions satisfy a property called submodularity. One such method is the consensus-based bundle algorithm (CBBA) (Choi et al. 2009), which is a polynomial-time, market-based approach that uses consensus to reach agreement on the cooperative assignment. The following section describes CBBA and illustrates how distributed auction-based allocation algorithms can produce provably good approximate solutions in distributed and decentralized environments.

60.4.1.1 Consensus-Based Bundle Algorithm (CBBA)

CBBA is a distributed auction algorithm that provides provably good task assignments for multi-agent, multi-task allocation problems. This algorithm has been shown to work well in practice because it can handle complicated constraints in a real-time distributed way. The algorithmic structure of CBBA consists of iterations between two phases: a *bundle-building* phase where each agent greedily generates an ordered bundle of tasks and a *task consensus* phase where conflicting assignments are identified and resolved through local communication between neighboring agents. These two phases are repeated until the algorithm has reached convergence.

To further explain the relevant details of the algorithm, some notation will first be formalized:

1. A *bid* is represented as a triple: $s_{ij} = (i, j, c_{ij})$, where i represents the bidding agent's index, j represents the task's index, and c_{ij} represents the bid value for this agent-task pair.
2. A *bundle* is an ordered data structure internal to each agent i , $\mathbf{b}_i = \{s_{ij_1}, \dots, s_{ij_n}\}$ that consists of all n of its current bids. When new bids are made, they are appended to the end of the bundle; thus, the order in the bundle reflects the relative age of each bid.
3. The *bid space* is an unordered set of bids, defined as $\mathcal{A} = \{s_{i_1 j_1}, \dots, s_{i_N j_N}\}$. This set contains a globally consistent set of the current winning bids in the team. A local bid space \mathcal{A}_i is defined as a set that contains agent i 's current local understanding of the global bid space. In a fully connected network, $\mathcal{A}_i = \mathcal{A}$ after each consensus phase, but in general, the geometry of agents in the network may lead to information propagation latencies and thus nonidentical bid spaces.

Bundle-Building Phase For each agent i , the bundle-building phase is run independently.

1. At the beginning of the bundle-building phase, the previous bundle is cleared and all bids that were won by agent i located in the local bid space, \mathcal{A}_i , are also removed. This step is required for the optimality guarantees of the algorithm, but in most cases, the agent will re-add each of the tasks it has just dropped from \mathcal{A}_i as it is building back up the bundle.
2. For each task j available in the environment, each agent i uses its local internal score function $F_{ij}(\mathbf{b}_i) = c_{ij}$, which is a function of its current bundle, to create a score c_{ij} . These scores are then ordered from highest to lowest.
3. The ordered scores c_{ij} are compared with the winning bid information for the corresponding task j located in the agent's local bid space \mathcal{A}_i . The first (and thus largest) score that would outbid the current winner in the local bid space is chosen as agent i 's next bid. A bid s_{ij} is created and placed at the end of the bundle and replaces the current bid on task j in the local bid space. Steps 2 and 3 are repeated until no tasks are able to outbid \mathcal{A}_i or the maximum bundle length is reached, at which point the bundle-building phase terminates. It is worth noting that in this formulation, the values c_{ij} are used to rank the tasks, and the values s_{ij} are communicated as bids to other agents. Algorithm performance can be increased by explicitly separating these two values (i.e., internal scores vs. external bids) (Johnson et al. 2012), enabling a larger class of allowable (more representative) score functions while still ensuring that properties required for algorithmic convergence are met.

Consensus Phase After the bundle-building phase completes, each agent i synchronously shares its current local bid space \mathcal{A}_i with each of its adjacent neighbors. This local bid space, in combination with time-stamp information, is then passed through a decision table (see Choi et al. 2009, Table 1 for details) that provides all of the conflict resolution logic to merge local bid spaces. In general, the consensus logic prefers larger and more recent bids. If the consensus phase has occurred more than twice the network diameter times without any bids changing, the algorithm has converged and terminates; if not, each agent reenters the bundle-building phase and the algorithm continues.

Score Function One requirement that is fundamental for all of the convergence and performance guarantees of CBBA is that the score functions used must satisfy a diminishing marginal gains (DMG) property. This DMG condition is a subset of the well-studied submodularity condition, was recognized in the original description of CBBA (Choi et al. 2009), and further studied in recent work (Johnson et al. 2012). It is defined as

$$F_{ij}(\mathbf{b}_i) \geq F_{ij}(\mathbf{b}_i \oplus_{\text{end}} s_{ik^*}) \quad \forall k^* \in \{1, \dots, N_i\}, \quad (60.38)$$

where $\mathbf{b}_i \oplus_{\text{end}} s_{ik^*}$ refers to adding a bid s_{ik^*} to an already existing bundle \mathbf{b}_i . Roughly, this condition means that no bids s_{ik^*} can be made that would increase c_{ij} , agent i 's score for task j . Given score functions that satisfy DMG, CBBA is guaranteed to converge to a conflict-free solution despite possible inconsistencies

in situational awareness and to achieve at least 50% optimality (Choi et al. 2009) (although empirically its performance has been shown to be above 90% optimality for some UAV environments (Bertuccelli et al. 2009)). The bidding process of CBBA runs in polynomial time, demonstrating good scalability with increasing numbers of agents and tasks, making it well suited to real-time dynamic environments. The real-time implementation of CBBA has been demonstrated for heterogeneous teams, and the algorithm has been recently extended to account for timing considerations associated with task execution, asynchronous communication environments, cooperative planning missions, and uncertain operation environments (Ponda et al. 2010, 2011, 2012; Whitten et al. 2011; Johnson et al. 2011).

60.4.2 Decentralized Markov Decision Processes

As mentioned previously in Sect. 60.2.2, the centralized multi-agent MDP formulation implicitly assumed that each agent had individual full observability of the joint state. When agents are making decisions and observations locally, this is generally an impractical assumption, since it would require each agent to communicate their observations to every other agent at every time step with zero communication cost (commonly referred to as a *free comm* environment). In realistic missions, however, there are operational costs and constraints associated with communication (e.g., bandwidth or dynamic network topologies) which need to be accounted for when planning. To address this issue, several decentralized MDP formulations have been proposed which extend the centralized multi-agent MDP (or POMDP) framework to explicitly account for communication between agents. However, this additional layer increases the dimensionality of the problem which exacerbates computational intractability. This section reviews some of these decentralized algorithms, discusses associated challenges, and presents approximations that can be made to address computational complexity.

The decentralized MDP (Dec-MDP), originally proposed in Bernstein et al. (2002), relaxes the limiting assumption of individual full observability by replacing it with a *joint full observability* requirement (i.e., together the agents have full observability of the state, but individual agents are not required to know the full state on their own). Agents can share parts of the state with each other through communication after a certain number of time steps, but this additional level of communication increases the computational complexity required by the algorithm (double exponential complexity or NEXP). As a reminder, the complexity hierarchy is given by $P \subseteq NP \subseteq PSPACE \subseteq EXP \subseteq NEXP$ denoting which classes of problems are subsets of other classes with regard to increasing computational complexity. For more details and discussion on complexity classes, the reader is referred to Bernstein et al. (2002), Papadimitriou and Tsitsiklis (1987), and Papadimitriou (2003). Similar to the Dec-MDP, the Dec-POMDP (Bernstein et al. 2002) is the decentralized version of the POMDP, where there is additional uncertainty in the state given information provided by the observations (the global state cannot be uniquely determined given a joint observation). Like the Dec-MDP,

the Dec-POMDP assumes that agents have different partial information about the global state. If it is assumed that each agent communicates its observations to every other agent at every time step with zero cost, then the Dec-MDP and Dec-POMDP reduce to the MMDP or MPOMDP described in Sect. 60.2.2. This situation is referred to as *free comm*, and the complexity of the Dec-MDP/Dec-POMDP reduces to that of its MDP/POMDP counterpart (P or PSPACE complexity, respectively). Therefore, the lack of immediate communication (or isolation) between agents can be associated with a jump in computational complexity (Goldman and Zilberstein 2004). Another framework similar to the Dec-POMDP that has been considered in the literature is the multi-agent team decision problem (MTDP) (Pynadath and Tambe 2002) which also consists of decentralized MDP solutions for cooperating agents. The MTDP and the Dec-POMDP have been shown to be equivalent in terms of computational complexity and expressiveness of representation (Seuken and Zilberstein 2008).

Since communication is a key component of decentralized planning, the above formulations have been extended to explicitly model communication between agents. The resulting algorithms, namely, the Dec-POMDP-COM (Goldman and Zilberstein 2003) and the COM-MTDP (Pynadath and Tambe 2002), treat these communications as actions which agents must make explicit decisions about at every time step. In particular, sending a message is considered an action which incurs an associated cost, and each agent needs to decide what messages to send at every given time, where not sending anything is modeled as a null message with zero cost. The reward function is accordingly modified to include the cost of communication. As such, the problem formulation can be extended to make sequential decisions about which messages to send as well as which actions to perform. The tuple now becomes $\langle \mathcal{S}, \mathcal{A}, P, R, \Omega, \mathcal{O}, \Sigma, C_\Sigma, R_T, \gamma \rangle$, where the additional terms are defined below:

- Σ : Defines the joint message space $\Sigma = \Sigma_1 \times \dots \times \Sigma_n$ composed of a discrete set of message options, where Σ_i is the message space for agent i and joint message vector $\sigma \in \Sigma$ is of the form $\sigma = (\sigma_1, \dots, \sigma_n)$ representing the messages sent by each agent.
- C_Σ : Defines a function quantifying the cost of communication where $C_\Sigma(\sigma) : \Sigma \rightarrow \mathbb{R}$ is the cost of sending message σ and is typically a sum over all individual agent message costs, with a null message having zero cost.
- R_T : Defines a total reward function $R_T(s, a, s', \sigma) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \Sigma \rightarrow \mathbb{R}$ specifying the total rewards and costs associated with applying action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, transitioning to state $s' \in \mathcal{S}$, and communicating message $\sigma \in \Sigma$. Typically, $R_T(s, a, s', \sigma) = R(s, a, s') - C_\Sigma(\sigma)$.

In this framework, a joint policy $\pi = (\pi_1^A, \pi_1^\Sigma, \dots, \pi_n^A, \pi_n^\Sigma)$ contains individual agent policies $\pi_i = (\pi_i^A, \pi_i^\Sigma)$ composed of local policies for actions and for communications, respectively.

In Seuken and Zilberstein (2008), the computational complexity of the Dec-POMDP, MTDP, Dec-POMDP-COM, and COM-MTDP is analyzed, and all of these algorithms have been shown to be NEXP-complete (nondeterministic exponential time), suffering from bad scalability as the number of agents increases.

The varying degrees of complexity between the different problem formulations are directly connected to the assumptions on observability and communication (Seuken and Zilberstein 2008), motivating the development of approximate problem formulations that make assumptions on independence and communication between agents to reduce the computational complexity. These approximation algorithms exploit the structure of the domain to make intelligent decisions about which assumptions to make given the scenario at hand (independence, feature-based representations, etc.), leading to a trade-off between optimality and computational tractability. It is worth noting that these approximations are made in the problem formulation itself rather than in the solution algorithm employed. The most notable of these approximation algorithms is the transition-independent decentralized MDP (TI-Dec-MDP) (Becker 2004), which assumes that agents are independent collaborating entities connected through a global reward that is a function of all agents' states and actions. The TI-Dec-MDP extends the factored Dec-MDP, where the global state can be broken up into local agent states plus some world state that represents the environment variables of interest and which is independent of agents' states and actions ($\mathcal{S} = \mathcal{S}_0 \times \mathcal{S}_1 \times \dots \times \mathcal{S}_n$, where \mathcal{S}_i is each agent's local state space and \mathcal{S}_0 is the world state space). In this formulation, the transition and observation dynamics for each agent are independent (i.e., only functions of the local agent state and the world state $\mathcal{S}_0 \times \mathcal{S}_i$), and the only coupling between agents is given by the joint reward function. These independence assumptions reduce the computational complexity of the Dec-MDP; however, although this approach scales to higher dimensions better than many others, it requires an extensive list of state transition histories per agent, referred to as *events* (Becker 2004; Redding et al. 2012). Furthermore, the independence assumption does not allow the representation to explicitly model global situations that are functions of the combined agents' states (e.g., vehicle collisions), although these can be prevented by assigning very large negative rewards (which are joint), thus making collision avoidance part of the reward function instead of an explicit constraint. Some other approximation architectures include the decentralized sparse-interaction MDP (Dec-SIMDP) (Melo and Veloso 2011), which deals with computational limitations by separating the parts of the state space where agents need to cooperate from the parts where they can act independently; the group-aggregate Dec-MMDP (Redding et al. 2012), which uses features to compress other agents' state-action spaces to highlight the properties of interest for each agent (e.g., how many agents are in an area of interest vs. where is every agent in the environment); and the auctioned POMDP (Capitán et al. 2011), where each agent solves its own POMDP locally and communicates with other agents via an auction algorithm to achieve consensus on plans. Although these algorithms have demonstrated reduced computational complexity and real-time feasibility for large teams of cooperating agents, the approximation strategies involved are typically ad hoc and problem dependent, and developing good approximation strategies for cooperating agents remains an active area of research.

60.5 Conclusions and Future Directions

This chapter presents autonomous cooperative task allocation and planning methods for heterogeneous networked teams, providing an overview of the numerous approaches that have been considered in the literature. In particular, three standard planning frameworks are discussed: integer programming, Markov decision processes, and game theory. The chapter also outlines various architectural decisions that must be addressed when implementing online planning systems for multi-agent teams, providing insights on when centralized, distributed, and decentralized architectures might be good choices for a given application, and how to organize the communication and computation to achieve the desired mission performance. Algorithms that can be utilized within the various architectures are identified, and the main technical challenges and future research directions are discussed.

As mentioned throughout the various sections in this chapter, one of the major challenges associated with cooperative planning algorithms is maintaining computational tractability, especially for distributed and decentralized systems. Designing efficient approximation algorithms that can be used in real-time dynamic environments remains an active area of research in the integer programming, MDP, and game theory communities. Furthermore, designing algorithms that can handle message delays, dynamic networks, and unreliable communication links involves making trade-offs between performance and robustness which can vary depending on the scenario at hand. Many of the open issues in this field revolve around identifying problem-specific features that can be exploited to improve performance or reduce the computational effort. For example, submodularity in the score functions has recently been used to develop greedy algorithms that provide bounds on the suboptimality of the performance (Singh et al. 2009; Golovin and Krause 2010; Krause et al. 2006). The challenge in this case is ensuring that the problem formulation satisfies the submodularity (or adaptive submodularity) conditions.

Another major consideration is that planning algorithms rely on underlying system models, which are usually approximations of the real systems and thus are often inaccurate. Discrepancies between these planner models and the actual system dynamics can cause significant degradations in mission performance. Thus, the planning systems must be efficiently integrated with online parameter identification and learning algorithms to adapt these models in real time (Redding et al. 2010). This process could also include elements of active learning, wherein part of the task assignment is specifically focused on deciding how best to reduce uncertainty to improve mission performance (Bertuccelli and How 2011). Robust planning and model learning, especially in decentralized and dynamic environments, remains an active area of research.

Much of the literature to date has been devoted to developing algorithms that perform well in a single type of environment. Future algorithms should be able to exploit dynamic environmental conditions and adapt their solutions accordingly. For example, if the communication environment is particularly harsh, an algorithm should be robust enough to handle the situation. If conditions improve, however,

the algorithm should be able to leverage the additional communication structure and information to improve performance. These hybrid adaptive planners would be able to make efficient use of whatever information is available, yet remain robust to worst-case environments.

Finally, the algorithms presented in this chapter were discussed in the context of autonomous operations, but many future applications will require human operators on the loop (as opposed to being in the loop). In those scenarios, future research must develop techniques to efficiently allocate the roles and responsibilities of the operator and the autonomy/planning algorithms to ensure synergistic operations while performing the mission. Determining the objective function structure in the optimization to develop plans that are consistent with the human operator's overall objectives is also typically quite challenging, especially for decentralized operations (Cummings et al. 2012), and thus is a promising area for future research.

Acknowledgments This work was supported in part by the AFOSR and USAF under grant (FA9550-08-1-0086) and MURI (FA9550-08-1-0356). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. government.

References

- A. Ahmed, A. Patel, T. Brown, M. Ham, M. Jang, G. Agha, Task assignment for a physical agent team via a dynamic forward/reverse auction mechanism, in *International Conference on Integration of Knowledge Intensive Multi-Agent Systems* (IEEE, Piscataway, 2005)
- B. Alidaee, H. Wang, F. Landram, A note on integer programming formulations of the real-time optimal scheduling and flight path selection of UAVs. *IEEE Trans. Control Syst. Technol.* **17**(4), 839–843 (2009)
- B. Alidaee, H. Wang, F. Landram, On the flexible demand assignment problems: case of unmanned aerial vehicles. *IEEE Trans. Autom. Sci. Eng.* **8**(4), 865–868 (2011)
- M. Alighanbari, J.P. How, A robust approach to the UAV task assignment problem. *Int. J. Robust Nonlinear Control* **18**(2), 118–134 (2008a)
- M. Alighanbari, J.P. How, An unbiased Kalman consensus algorithm. *AIAA J. Aerosp. Comput. Inf. Commun.* **5**(9), 298–311 (2008b)
- G. Arslan, J. Marden, J. Shamma, Autonomous vehicle-target assignment: a game-theoretical formulation. *J. Dyn. Syst. Meas. Control* **129**, 584 (2007)
- M.L. Atkinson, Results analysis of using free market auctions to distribute control of UAVs, in *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit* (AIAA, Reston, 2004)
- A.G. Banerjee, M. Ono, N. Roy, B.C. Williams, Regression-based LP solver for chance-constrained finite horizon optimal control with nonconvex constraints, in *Proceedings of the American Control Conference* (San Francisco, 2011)
- R. Beard, V. Stepanyan, Information consensus in distributed multiple vehicle coordinated control. *IEEE Conf. Decis. Control* **2**, 2029–2034 (2003)
- R.W. Beard, T.W. McLain, M.A. Goodrich, E.P. Anderson, Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans. Robot. Autom.* **18**, 911–922 (2002)
- R. Becker, Solving transition independent decentralized Markov decision processes, in *Computer Science Department Faculty Publication Series*, 2004, pp. 208

- J. Bellingham, A. Richards, J.P. How, Receding horizon control of autonomous aerial vehicles. *Am. Control Conf.* **5**, 3741–3746 (2002)
- R. Bellman, *Dynamic Programming* (Dover, Mineola, 2003)
- A. Ben-Tal, A. Nemirovski, Robust convex optimization. *Math. Oper. Res.* **23**(4), 769–805 (1998)
- D. Bernstein, R. Givan, N. Immerman, S. Zilberstein, The complexity of decentralized control of Markov decision processes, in *Mathematics of operations research* (2002), pp. 769–805. <http://dl.acm.org/citation.cfm?id=2073951>
- D.P. Bertsekas, The auction algorithm for assignment and other network flow problems, Technical report, MIT, 1989
- D.P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. I–II, 3rd edn. (Athena Scientific, Belmont, 2007)
- D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods* (Prentice-Hall, Englewood Cliffs, 1989)
- D. Bertsimas, D. Brown, Constructing uncertainty sets for robust linear optimization. *Oper. Res.* **57**(6), 1483–1495 (2009)
- D. Bertsimas, R. Weismantel, *Optimization over integers* (Dynamic Ideas, Belmont, 2005)
- D. Bertsimas, D.B. Brown, C. Caramanis, Theory and applications of robust optimization. *SIAM review.* **53**(3), 464–501 (2011)
- L. Bertuccelli, J. How, Active exploration in robust unmanned vehicle task assignment. *J. Aerosp. Comput. Inf. Commun.* **8**, 250–268 (2011)
- L. Bertuccelli, H. Choi, P. Cho, J. How, Real-time multi-UAV task assignment in dynamic and uncertain environments, in *AIAA Guidance, Navigation, and Control Conference* (AIAA, Reston, 2009). (AIAA 2009–5776)
- B.M. Bethke, Kernel-based approximate dynamic programming using bellman residual elimination, Ph.D. thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, 2010
- B. Bethke, J.P. How, J. Vian, Group health management of UAV teams with applications to persistent surveillance, in *American Control Conference (ACC)*, Seattle (IEEE, New York, 2008), pp. 3145–3150
- L. Blackmore, M. Ono, Convex chance constrained predictive control without sampling, in *AIAA Proceedings*. (np) (2009)
- V.D. Blondel, J.M. Hendrickx, A. Olshevsky, J.N. Tsitsiklis, Convergence in multiagent coordination, consensus, and flocking, in *Proceeding of the IEEE Conference on Decision and Control* (2005)
- S. Bradtke, A. Barto, Linear least-squares algorithms for temporal difference learning. *J. Mach. Learn. Res.* **22**, 33–57 (1996)
- L. Buşoniu, R. Babuška, B. De Schutter, D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators* (CRC, Boca Raton, 2010)
- J. Capitán, M. Spaan, L. Merino, A. Ollero, Decentralized multi-robot cooperation with auctioned POMDPs, in *Sixth Annual Workshop on Multiagent Sequential Decision Making in Uncertain Domains (MSDM-2011)*, 2011, p. 24
- D. Castanon, J. Wohletz, Model predictive control for stochastic resource allocation. *IEEE Trans. Autom. Control* **54**(8), 1739–1750 (2009)
- D.A. Castanon, C. Wu, Distributed algorithms for dynamic reassignment. *IEEE Conf. Decis. Control* **1**, 13–18 (2003)
- P.R. Chandler, M. Pachter, D. Swaroop, J.M. Fowler, J.K. Howlett, S. Rasmussen, C. Schumacher, K. Nygard, Complexity in UAV cooperative control, in *American Control Conference (ACC)*, Anchorage, 2002
- A. Chapman, R. Micillo, R. Kota, N. Jennings, Decentralized dynamic task allocation using overlapping potential games. *Comput. J.* **53**, 1462–1477 (2010)
- W. Chen, M. Sim, J. Sun, C. Teo, From CVaR to uncertainty set: implications in joint chance constrained optimization. *Oper. Res.* **58**(2), 470–485 (2010)
- T. Chockalingam, S. Arunkumar, A randomized heuristics for the mapping problem: the genetic approach. *Parallel Comput.* **18**(10), 1157–1165 (1992)

- H.-L. Choi, L. Brunet, J.P. How, Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robot.* **25**(4), 912–926 (2009)
- T. Cormen, *Introduction to Algorithms* (MIT, Cambridge, 2001)
- J. Cruz Jr., G. Chen, D. Li, X. Wang, Particle swarm optimization for resource allocation in UAV cooperative control, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence (AIAA, Reston, 2004), pp. 1–11
- M.L. Cummings, J.P. How, A. Whitten, O. Toupet, The impact of human–automation collaboration in decentralized multiple unmanned vehicle control. *Proc. IEEE* **100**(3), 660–671 (2012)
- P. De Boer, D. Kroese, S. Mannor, R. Rubinstein, A tutorial on the cross-entropy method. *Ann. Oper. Res.* **134**(1), 19–67 (2005)
- E. Delage, S. Mannor, Percentile optimization for Markov decision processes with parameter uncertainty. *Oper. Res.* **58**(1), 203–213 (2010)
- M.B. Dias, A. Stentz, A free market architecture for distributed control of a multirobot system, in *6th International Conference on Intelligent Autonomous Systems IAS-6* (IOS, Amsterdam/Washington, DC, 2000), pp. 115–122
- M.B. Dias, R. Zlot, N. Kalra, A. Stentz, Market-based multirobot coordination: a survey and analysis. *Proc. IEEE* **94**(7), 1257–1270 (2006)
- Y. Eun, H. Bang, Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithms. *J. Aircr.* **46**(1), 338 (2010)
- A.M. Farahmand, M. Ghavamzadeh, C. Szepesvári, S. Mannor, Regularized policy iteration, in *Advances in Neural Information Processing Systems (NIPS)*, ed. by D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (MIT, Cambridge, 2008), pp. 441–448
- J. Fax, R. Murray, Information flow and cooperative control of vehicle formations. *IEEE Trans. Autom. Control* **49**(9), 1465–1476 (2004)
- C.A. Floudas, *Nonlinear and Mixed-Integer Programming – Fundamentals and Applications* (Oxford University Press, New York, 1995)
- C.S.R. Fraser, L.F. Bertuccelli, J.P. How, Reaching consensus with imprecise probabilities over a network, in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Chicago, 2009. (AIAA-2009-5655)
- C.S. Fraser, L.F. Bertuccelli, H.-L. Choi, J.P. How, A hyperparameter consensus method for agreement under uncertainty. *Automatica* **48**(2), 374–380 (2012)
- E.W. Frew, B. Argrow, Embedded reasoning for atmospheric science using unmanned aircraft systems, in *AAAI 2010 Spring Symposium on Embedded Reasoning: Intelligence in Embedded Systems*, Palo Alto (AAAI, Menlo Park, 2010)
- D. Fudenberg, J. Tirole, *Game Theory* (MIT, Cambridge, 1991)
- A. Gelman, J. Carlin, H. Stern, D. Rubin, *Bayesian Data Analysis*, 2nd edn. (Chapman and Hall, Boca Raton, 2004)
- A. Geramifard, F. Doshi, J. Redding, N. Roy, J. How, Online discovery of feature dependencies, in *International Conference on Machine Learning (ICML)*, ed. by L. Getoor, T. Scheffer (ACM, New York, 2011), pp. 881–888
- B. Gerkey, M. Mataric, Sold!: auction methods for multirobot coordination. *IEEE Trans. Robot. Autom.* **18**(5), 758–768 (2002)
- B.P. Gerkey, M.J. Mataric, A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **23**(9), 939–954 (2004)
- F. Glover, R. Marti, Tabu search, in *Metaheuristic Procedures for Training Neural Networks* (Springer, Boston, 2006), pp. 53–69
- C. Goldman, S. Zilberstein, Optimizing information exchange in cooperative multi-agent systems, in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems* (ACM, New York, 2003), pp. 137–144
- C. Goldman, S. Zilberstein, Decentralized control of cooperative systems: categorization and complexity analysis. *J. Artif. Intell. Res.* **22**, 143–174 (2004)
- D. Golovin, A. Krause, Adaptive submodularity: a new approach to active learning and stochastic optimization, *Proceedings of the International Conference on Learning Theory (COLT)*, 2010

- S. Grime, H. Durrant-Whyte, Data fusion in decentralized sensor networks. *Control Eng. Pract.* **2**(5), 849–863 (1994)
- C. Guestrin, D. Koller, R. Parr, Multiagent planning with factored MDPs, in *NIPS*, ed. by T.G. Dietterich, S. Becker, Z. Ghahramani (MIT, Cambridge, 2001), pp. 1523–1530
- Y. Hatano, M. Mesbahi, Agreement over random networks. *IEEE Trans. Autom. Control* **50**(11), 1867–1872 (2005)
- ILOG, CPLEX (2006), <http://www.ilog.com/products/cplex/>
- A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* **48**(6), 988–1001 (2003)
- L.B. Johnson, S.S. Ponda, H.-L. Choi, J.P. How, Asynchronous decentralized task allocation for dynamic environments, in *Proceedings of the AIAA Infotech@Aerospace Conference*, St. Louis (AIAA, Reston, 2011)
- L.B. Johnson, H.-L. Choi, S.S. Ponda, J.P. How, Allowing non-submodular score functions in distributed task allocation, in *IEEE Conference on Decision and Control (CDC)*, 2012 (submitted)
- Y. Kim, D. Gu, I. Postlethwaite, Real-time optimal mission scheduling and flight path selection. *IEEE Trans. Autom. Control* **52**(6), 1119–1123 (2007)
- E. King, Y. Kuwata, M. Alighanbari, L. Bertuccelli, J.P. How, Coordination and control experiments on a multi-vehicle testbed, in *American Control Conference (ACC)*, Boston, (American Automatic Control Council, Evanston; IEEE, Piscataway, 2004), pp. 5315–5320
- A. Krause, C. Guestrin, A. Gupta, J. Kleinberg, Near-optimal sensor placements: maximizing information while minimizing communication cost, in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on* (ACM, New York, 2006), pp. 2–10, 0–0
- M.G. Lagoudakis, R. Parr, Least-squares policy iteration. *J. Mach. Learn. Res.* **4**, 1107–1149 (2003)
- G. Laporte, F. Semet, Classical heuristics for the capacitated VRP, in *The Vehicle Routing Problem*, ed. by P. Toth, D. Vigo (Society for Industrial Mathematics, Philadelphia, 2002)
- S. Leary, M. Deittert, J. Bookless, Constrained UAV mission planning: a comparison of approaches, in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Barcelona (IEEE, Piscataway, 2011), pp. 2002–2009
- T. Lemaire, R. Alami, S. Lacroix, A distributed task allocation scheme in multi-UAV context. *IEEE Int. Conf. Robot. Autom.* **4**, 3622–3627 (2004)
- J. Lin, A. Morse, B. Anderson, The multi-agent rendezvous problem. *IEEE Conf. Decis. Control* **2**, 1508–1513 (2003)
- S. Mahadevan, M. Maggioni, C. Guestrin, Proto-value functions: a Laplacian framework for learning representation and control in Markov decision processes. *J. Mach. Learn. Res.* **8**, 2007 (2006)
- A. Makarenko, H. Durrant-Whyte, Decentralized Bayesian algorithms for active sensor networks. *Int. Conf. Inf. Fusion* **7**(4), 418–433 (2006)
- N.D. Manh, L.T.H. An, P.D. Tao, A cross-entropy method for nonlinear UAV task assignment problem, in *IEEE International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)* (IEEE, Piscataway, 2010), pp. 1–5
- J. Marden, A. Wierman, Overcoming limitations of game-theoretic distributed control, in *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference* (IEEE, Piscataway, 2009)
- J. Marden, G. Arslan, J. Shamma, Cooperative control and potential games. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **39**(6), 1393–1407 (2009)
- M.J. Mataric, G.S. Sukhatme, E.H. Ostergaard, Multi-robot task allocation in uncertain environments. *Auton. Robots* **14**(2–3), 255–263 (2003)
- I. Maza, F. Caballero, J. Capitán, J. Martínez-de Dios, A. Ollero, Experimental results in multi-UAV coordination for disaster management and civil security applications. *J. Intell. Robot. Syst.* **61**(1), 563–585 (2011)

- T.W. McLain, R.W. Beard, Coordination variables, coordination functions, and cooperative-timing missions. *J. Guid. Control Dyn.* **28**(1), 150–161 (2005)
- F.S. Melo, M. Veloso, Decentralized MDPs with sparse interactions. *Artif. Intell.* **175**, 1757–1789 (2011)
- C.C. Moallemi, B.V. Roy, Consensus propagation. *IEEE Trans. Inf. Theory* **52**(11), 4753–4766 (2006)
- D. Monderer, L. Shapley, Potential games. *Games Econ. Behav.* **14**, 124–143 (1996)
- R. Murphey, Target-based weapon target assignment problems. *Nonlinear Assign. Probl. Algorithms Appl.* **7**, 39–53 (1999)
- A. Nemirovski, A. Shapiro, Convex approximations of chance constrained programs. *SIAM J. Optim.* **17**(4), 969–996 (2007)
- I. Nikolos, E. Zografos, A. Brintaki, UAV path planning using evolutionary algorithms, in *Innovations in Intelligent Machines-1* (Springer, Berlin/New York, 2007), pp. 77–111
- Office of the Secretary of Defense, Unmanned aircraft systems roadmap, Technical report, OSD (2007), <http://www.acq.osd.mil/usd/UnmannedSystemsRoadmap.2007-2032.pdf>
- R. Olfati-saber, Distributed Kalman filtering and sensor fusion in sensor networks, in *Network Embedded Sensing and Control*, vol. 331 (Springer, Berlin, 2006), pp. 157–167
- R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **49**(9), 1520–1533 (2004)
- R. Olfati-Saber, A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems. *IEEE Trans. Autom. Control* **95**(1), 215–233 (2007)
- A. Olshevsky, J.N. Tsitsiklis, Convergence speed in distributed consensus and averaging, in *IEEE Conference on Decision and Control (CDC)* (IEEE, Piscataway, 2006), pp. 3387–3392
- C. Papadimitriou, *Computational Complexity* (Wiley, Chichester, 2003)
- C.H. Papadimitriou, J.N. Tsitsiklis, The complexity of Markov decision processes. *Math. Oper. Res.* **12**(3), 441–450 (1987)
- S. Paquet, L. Tobin, B. Chaib-draa, Real-time decision making for large POMDPs. *Adv. Artif. Intell.* **3501**, 450–455 (2005)
- K. Passino, M. Polycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint, M. Baum, Cooperative control for autonomous air vehicles, in *Cooperative control and optimization* (Kluwer, Dordrecht/Boston, 2002), pp. 233–271
- S.S. Ponda, J. Redding, H.-L. Choi, J.P. How, M.A. Vavrina, J. Vian, Decentralized planning for complex missions with dynamic communication constraints, in *American Control Conference (ACC)*, Baltimore, 2010
- S.S. Ponda, L.B. Johnson, H.-L. Choi, J.P. How, Ensuring network connectivity for decentralized planning in dynamic environments, in *Proceedings of the AIAA Infotech@Aerospace Conference*, St. Louis (AIAA, Reston, 2011)
- S.S. Ponda, L.B. Johnson, J.P. How, Distributed chance-constrained task allocation for autonomous multi-agent teams, in *American Control Conference (ACC)*, 2012
- A. Pongpunwattana, R. Rysdyk, J. Vagners, D. Rathbun, Market-based co-evolution planning for multiple autonomous vehicles, in *Proceedings of the AIAA Unmanned Unlimited Conference*, San Diego (AIAA, Reston, 2003)
- D. Pynadath, M. Tambe, The communicative multiagent team decision problem: analyzing teamwork theories and models. *J. Artif. Intell. Res.* **16**(1), 389–423 (2002)
- S. Rathinam, R. Sengupta, S. Darbha, A resource allocation algorithm for multivehicle systems with nonholonomic constraints. *IEEE Trans. Autom. Sci. Eng.* **4**(1), 98–104 (2007)
- J. Redding, A. Geramifard, A. Undurti, H. Choi, J. How, An intelligent cooperative control architecture, in *American Control Conference (ACC)*, Baltimore, 2010, pp. 57–62
- J.D. Redding, N.K. Ure, J.P. How, M. Vavrina, J. Vian, Scalable, MDP-based planning for multiple, cooperating agents, in *American Control Conference (ACC)* (2012, to appear)
- W. Ren, Consensus based formation control strategies for multi-vehicle systems, in *American Control Conference (ACC)* (American Automatic Control Council, Evanston; IEEE, Piscataway, 2006), pp. 6–12

- W. Ren, R. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **50**(5), 655–661 (2005)
- W. Ren, R.W. Beard, D.B. Kingston, Multi-agent Kalman consensus with relative uncertainty. *Am. Control Conf.* **3**, 1865–1870 (2005)
- W. Ren, R.W. Beard, E.M. Atkins, Information consensus in multivehicle cooperative control. *IEEE Control Syst. Mag.* **27**(2), 71–82 (2007)
- A. Richards, J. Bellingham, M. Tillerson, J.P. How, Coordination and control of multiple UAVs, in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Monterey (AIAA, Reston, 2002). AIAA Paper 2002–4588
- G.A. Rummery, M. Niranjan, Online Q-learning using connectionist systems (Technical Report No. CUED/F-INFENG/TR 166), Cambridge University Engineering Department (1994)
- R.O. Saber, W.B. Dunbar, R.M. Murray, Cooperative control of multi-vehicle systems using cost graphs and optimization, in *Proceedings of the 2003 American Control Conference, 2003*, vol. 3 (IEEE, Piscataway, 2003), pp. 2217–2222
- A. Salman, I. Ahmad, S. Al-Madani, Particle swarm optimization for task assignment problem. *Microprocess. Microsyst.* **26**(8), 363–371 (2002)
- S. Sariel, T. Balch, Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments, in *AIAA Workshop on Integrating Planning Into Scheduling (AAAI, Menlo Park, 2005)*
- K. Savla, E. Frazzoli, F. Bullo, On the point-to-point and traveling salesperson problems for Dubins' vehicle, in *American Control Conference (ACC)*, June 2005. pp. 786–791
- B. Scherrer, Should one compute the temporal difference fix point or minimize the Bellman Residual? The unified oblique projection view, *International Conference on Machine Learning (ICML)* (IEEE, Los Alamitos, 2010)
- D.G. Schmale, B. Dingus, C. Reinholtz, Development and application of an autonomous unmanned aerial vehicle for precise aerobiological sampling above agricultural fields. *J. Field Robot.* **25**(3), 133–147 (2008)
- C. Schumacher, P.R. Chandler, S. Rasmussen, Task allocation for wide area search munitions via network flow optimization, in *Proceedings of the American Control Conference*, Anchorage, 2002, pp. 1917–1922
- S. Seuken, S. Zilberstein, Formal models and algorithms for decentralized decision making under uncertainty. *Auton. Agents Multi-Agent Syst.* **17**(2), 190–250 (2008)
- T. Shima, S.J. Rasmussen, *UAV Cooperative Decision and Control: Challenges and Practical Approaches*, vol. 18 (Society for Industrial Mathematics, Philadelphia, 2009)
- T. Shima, S. Rasmussen, A. Sparks, K. Passino, Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Comput. Oper. Res.* **33**(11), 3252–3269 (2006)
- A. Singh, A. Krause, W. Kaiser, Nonmyopic adaptive informative path planning for multiple robots, in *International Joint Conference on Artificial Intelligence (IJCAI)* (AAAI, Menlo Park, 2009)
- R.G. Smith, R. Davis, Frameworks for cooperation in distributed problem solving. *IEEE Trans. Syst. Man Cybern.* **11**(1), 61–70 (1981)
- M.T.J. Spaan, N. Vlassis, Perseus: randomized point-based value iteration for POMDPs. *Int. J. Robot. Res.* **24**, 195–220 (2005)
- P. Stone, R.S. Sutton, G. Kuhlmann, Reinforcement learning for RoboCup-Soccer keepaway. *Int. Soc. Adapt. Behav.* **13**(3), 165–188 (2005)
- P.B. Sujit, D. Kingston, R. Beard, Cooperative forest fire monitoring using multiple UAVs, in *IEEE Conference on Decision and Control*, New Orleans (IEEE, Piscataway, 2007), pp. 4875–4880
- R.S. Sutton, Generalization in reinforcement learning: successful examples using sparse coarse coding, in *Advances in Neural Information Processing Systems 8* (MIT, Cambridge/London, 1996), pp. 1038–1044
- R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction* (MIT, Cambridge, 1998)
- R.S. Sutton, H.R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, E. Wiewiora, Fast gradient-descent methods for temporal-difference learning with linear function approximation,

- in *International Conference on Machine Learning (ICML)*, ICML '09 (ACM, New York, 2009), pp. 993–1000
- A. Tahbaz-Salehi, A. Jadbabaie, On consensus over random networks, in *44th Annual Allerton Conference*, 2006
- P. Toth, D. Vigo, *The Vehicle Routing Problem* (Society for Industrial and Applied Mathematics, Philadelphia, 2001)
- J.N. Tsitsiklis, B.V. Roy, An analysis of temporal difference learning with function approximation. *IEEE Trans. Autom. Control* **42**(5), 674–690 (1997)
- K. Tumer, D. Wolpert, A survey of collectives, in *Collectives and the Design of Complex Systems* (Springer, New York, 2004), pp. 1–42
- A. Undurti, J.P. How, A Cross-entropy based approach for UAV task allocation with nonlinear reward, in *AIAA Guidance, Navigation, and Control Conference (GNC)* (AIAA, Reston, 2010). AIAA-2010-7731
- U.S. Air Force Chief Scientist (AF/ST), Technology horizons: a vision for air force science & technology during 2010–2030, Technical report, United States Air Force (2010)
- U.S. Army UAS Center of Excellence, Eyes of the Army: U.S. Army unmanned aircraft systems roadmap 2010–2035, Technical report (2010), <http://www.fas.org/irp/program/collect/uas-army.pdf>
- M. Valenti, B. Bethke, J.P. How, D.P. de Farias, J. Vian, Embedding health management into mission tasking for UAV teams, in *American Control Conference (ACC)*, New York (IEEE, New York, 2007), pp. 5777–5783
- E. Waltz, J. Llinas, *Multisensor Data Fusion* (Artech House, Boston/London, 1990)
- R.V. Welch, G.O. Edmonds, Applying robotics to HAZMAT, in *The Fourth National Technology Transfer Conference and Exposition*, vol. 2 (2003), pp. 279–287
- A.K. Whitten, H.-L. Choi, L. Johnson, J.P. How, Decentralized task allocation with coupled constraints in complex missions, in *American Control Conference (ACC)*, 2011, pp. 1642–1649
- C.W. Wu, Synchronization and convergence of linear dynamics in random directed networks. *IEEE Trans. Autom. Control* **51**(7), 1207–1210 (2006)
- L. Xiao, S. Boyd, S. Lall, A scheme for robust distributed sensor fusion based on average consensus, in *International Symposium on Information Processing in Sensor Networks* (ACM, New York, 2005), pp. 63–70
- R. Zhou, E.A. Hansen, An improved grid-based approximation algorithm for POMDPs, in *International Joint Conference on Artificial Intelligence*, vol. 17, number 1 (Morgan Kaufmann, San Francisco, 2001), pp. 707–716

Matthew E. Argyle, Randal W. Beard, and David W. Casbeer

Contents

61.1 Introduction 1492

61.2 Task Assignment Problem 1494

61.3 Multi-team Task Assignment 1495

 61.3.1 Local Task Assignment 1495

 61.3.2 Task Sharing 1496

 61.3.3 Phase 1: Build Bundle 1499

 61.3.4 Phase 2: Communication 1500

61.4 Convergence Time Analysis 1501

61.5 Simulation Results 1503

References 1506

Abstract

In this chapter, the consensus-based bundle algorithm (CBBA) is incorporated into a hierarchical concept of operation. In the team CBBA, agents are divided into teams and each team plans for its agents to service a set of tasks. This team planning is carried out in a distributed manner using the traditional CBBA. An “outer-loop” team CBBA strategy is presented that coordinates teams’ plans. The hierarchical structure of the team CBBA facilitates a manageable architecture for large numbers of unmanned agents through

M. Argyle (✉)
Department of Electrical Engineering, Brigham Young University, Provo, UT, USA
e-mail: matt.argyle@gmail.com

R. Beard
Electrical and Computer Engineering Department, Brigham Young University, Provo, UT, USA
e-mail: beard@byu.edu

D.W. Casbeer
Control Science Center of Excellence, Air Force Research Laboratory, WPAFB, OH, USA
e-mail: david.casbeer@us.af.mil

human-centered operations. This is because each team is managed by a human operator with the team CBBA aiding coordination between teams.

61.1 Introduction

Currently deployed unmanned air vehicles (UAV) are individually controlled by a group of operators that actively manage different components of the UAV. One operator might control the motion of the vehicle, either directly through “stick-and-rudder” inputs or through a waypoint navigation mode. Simultaneously, different groups of operators remotely control the onboard sensors and evaluate any received data. Recent research has been focused on inverting this ratio of operators per UAV to have a single operator manage a team of UAVs (Cummings and Bruni 2009; Cummings et al. 2012).

Planning algorithms have been developed to aid the operator with the management of a team of UAVs and current research is improving on these algorithms (Shima and Rasmussen 2009; Rasmussen and Shima 2008; Choi et al. 2009). To date, planners are generally implemented centrally, at a base station. Four main factors motivate this centralized paradigm:

1. Poor inter-UAV communication
2. Inadequate processing capability (applies to the computation and algorithms needed for both planning and sensor data processing)
3. Operators’ desire for a continuous stream of video and images
4. Pilot safety in airspace with both unmanned and manned aircraft

First, trustworthy sense-and-avoid capability will increase safety in airspace with piloted and unmanned aircraft. However, until such capabilities are operational, a human operator is necessary to ensure safe operation in mixed environments, which leads to an architecture with planning performed centrally. Second, in military applications, commanders want as much information as possible to facilitate informed decisions, hence the desire for continual video and image footage. If a system is designed to feed high-bandwidth video data to a central human-processor for evaluation, then it also makes sense to plan centrally since communicating C2 data causes negligible change to bandwidth requirements when compared to the transmitted video and images. The continual video and image feeds are, also, necessary because of the UAV’s inadequate ability to autonomously evaluate sensor data and derive actionable intelligence, e.g., detection and classification (factor 2). Lastly, most UAVs do not possess the onboard processing capability to perform the complex calculations necessary for combinatorial optimization (factor 2), and there is not dependable communication between the UAVs, making a distributed planner impractical since the UAVs cannot reliably coordinate their actions.

Decentralized control of UAVs, where the UAVs operate independently from human monitors, is a long-term goal, and in order to accomplish this, the four factors mentioned above will need to be resolved. In the short term, human operators will need to be closely involved with the UAV operation. As more and more UAVs and UAV teams are deployed simultaneously, one could naively allow separate UAV teams to function independently, where each team is monitored by an operator with

his/her own tasks to accomplish. However, a logical step would be to allow these separate teams to share resources. Ideally, the mission, as a whole, could be realized more efficiently when compared with independent operation.

Thus, it is timely to consider alternative management schemes. More specifically, a system-of-systems management architecture needs to be considered that is capable of dynamic distributed task assignment across multiple autonomous vehicles, ranging from small to large. This requires a new distributed management architecture that can coordinate a large number of intelligent agents accomplishing a large number of tasks in time-critical, dynamically changing environments. Moreover, the management approach must still retain human operators, given mission objectives can vary constantly as does the ground situation. The operators' involvement would be supervisory, with attention focused at the mission execution level. Yet, the operators must also be kept informed of the autonomous vehicles' states and task progress, in order to benefit from the human's unique ability to provide critical intuition and judgment for the execution of successful collaborative missions.

Current stick-and-rudder UAV operation is a conservative first step toward fully autonomous aircraft. It allows a human to make decisions for the UAV, avoiding the problems associated with factors 2 and 4 listed above. The first step toward achieving functional autonomous UAVs is to reduce the workload on the UAV operator(s) by allowing the UAV system to takeover duties that current technology allows a machine to perform robustly. The first step for small UAVs was to allow an onboard autopilot to take over the low-level control of the UAV, ensuring stable flight by moving operator control to a higher level of control, e.g., waypoint navigation and heading/altitude commands (Chao et al. 2010).

Autopilots generally have different modes of operation: one mode for direct remote control (RC) and a separate autopilot mode. Such a system could loosely be classified as an adjustably autonomous system or a system that allows the operator to take direct control of an aircraft but also allows the operator to step back and manage the UAV(s) or even simply monitor the UAV(s). *Adjustable autonomy*, sometimes called incremental autonomy or variable autonomy (see Bonasso et al. 1997; Pell et al. 1998), refers to a system with the ability to continuously change the level of autonomy between full autonomous and manual modes during system operation (Dorais et al. 1998). Adjustable autonomy is different from increment assistance in that an autonomous system is able to function on its own while an "assistance system" solely aids the operator (Baudin et al. 1994).

The idea of adjustable autonomy really goes much farther than the basic modes currently used by autopilots today. Adjustable autonomy really has its roots in the artificial intelligence community, the autonomous system can range from manual control to a fully autonomous mode (Scerri et al. 2002, 2003b). In the UAV setting, this implies the UAV is able to observe itself and its surroundings, place itself, in context, within those surroundings, and in pursuit of its mission's objective make appropriate decisions on actions that effect itself and its surroundings.

To further reduce the operator workload, a team paradigm is needed, which reduces the man-hours needed to operate multiple UAVs (Christoffersen and Woods 2004; Sycara and Sukthankar 2006). In the human factors arena, there has been research performed to investigate the effects of managing a team of robots on the

human operator (Kilgore et al. 2007; Lewis et al. 2010; Crandall et al. 2011; Sycara and Lewis 2004). In order for the robots to function independently as part of a team, the agents must be capable of determining teammates intent, meaning the intent of both robot teammates, as well as human teammates (Grosz and Kraus 2003; Scerri et al. 2003a; Sycara et al. 2006, 2003). However, due to the large scope of this problem, there is a lot of research still needed to make mixed human-robot teams practical.

In this chapter, an overview is given of the team consensus-based bundle algorithm (TCBBA) that was presented in Argyle and Casbeer (2011). TCBBA is a hierarchical and distributed task assignment algorithm based on the paradigm that human operators are responsible for their assigned UAVs. Rather than having humans input tasks to a swarm/cloud of agents and then the autonomy deciding which agent will do which task, here the agents are fixed to an operator and the tasks are shifted between teams. This algorithm follows current rules of engagement and is a novel distributed task assignment algorithm, in that it places end responsibility for the welfare of the agents and the success or failure of the mission directly on the human operator. In the TCBBA, each team is composed of heterogeneous UAVs, and a human operator is responsible for the UAVs assigned to his/her team. Each team of UAVs is given tasks that must be accomplished. These tasks are allotted to specific agents in the team using a distributed algorithm called the CBBA (Choi et al. 2009). The teams try to assign all the tasks while maximizing the total score of some cost function. If there are tasks that are not assigned, a second stage starts. In the second stage, the teams offer and request help to and from neighboring teams. This second stage of bidding consists of a higher-level auction algorithm running “outside” a lower level CBBA auction.

The format of this chapter is as follows: Sect. 61.2 lays out the specific problem that is addressed by the TCBBA. Section 61.3 discusses the multi-team extension to the CBBA, which is followed by analysis in Sect. 61.4. Lastly, simulations results and conclusions are presented in Sects. 61.5 and 61.6.

61.2 Task Assignment Problem

Consider the multi-agent multitask assignment problem where a group of N_a agents attempt to service N_t tasks while trying to maximize a reward. This can be stated as:

$$\begin{aligned}
 &\text{maximize} && \sum_{j=1}^{N_a} \sum_{k=1}^{N_t} c_{jk}(\mathbf{x}_j, \mathbf{p}_j) x_{jk} \\
 &\text{subject to} && \sum_{j=1}^{N_a} x_{jk} \leq 1 \quad \forall k \in \{1, \dots, N_t\} \\
 &&& \sum_{k=1}^{N_t} x_{jk} \leq L_m \quad \forall j \in \{1, \dots, N_a\},
 \end{aligned} \tag{61.1}$$

where $x_{jk} \in \{0, 1\}$ is one if agent j services task k and zero otherwise, $\mathbf{x}_j \in \{0, 1\}^{N_t}$ is a vector whose k th element is x_{jk} , \mathbf{p}_j is the path for agent j and indicates the order agent j will service its assigned tasks, $c_{jk}(\cdot)$ is the non-negative reward, and L_m is the maximum number of tasks per agent.

Due to the curse of dimensionality, an exact solution to the multi-agent multitask problem becomes impractical for large numbers of agents (Bellman 1957). To overcome this problem, current operations have multiple operators controlling either one UAV or possibly a group of UAVs. Each group of operators would service tasks independently. In this setup, a human remains responsible for his or her team of agents. The teams interact, with other teams, through higher-level coordination. Here, the N_t tasks and N_a agents are divided between N_o operators. Each team $i \in \{1, \dots, N_o\}$ tries to service all of its assigned tasks $\mathcal{N}_{ti} \subset \{1, \dots, N_t\}$ with its assigned agents $\mathcal{N}_{ui} \subset \{1, \dots, N_a\}$. The team assignment problem can be stated as:

$$\begin{aligned}
 & \text{maximize} && \sum_{j \in \mathcal{N}_{ui}} \sum_{k \in \mathcal{N}_{ti}} c_{jk}(\mathbf{x}_j, \mathbf{p}_j) x_{jk} \\
 & \text{subject to} && \sum_{j \in \mathcal{N}_{ui}} x_{jk} \leq 1 \quad \forall k \in \mathcal{N}_{ti} \\
 & && \sum_{j \in \mathcal{N}_{ti}} x_{jk} \leq L_m \quad \forall j \in \mathcal{N}_{ui}.
 \end{aligned} \tag{61.2}$$

A problem occurs when a team is unable to perform all of its assigned tasks due to time constraints, resource constraints, or agent capabilities. The TCBBA is a hierarchical task assignment method, that allows neighboring teams to request and provide help when this situation occurs.

61.3 Multi-team Task Assignment

The team consensus-based bundle algorithm (TCBBA) consists of two main stages: local task assignment and task sharing. In the first stage, each team implements the traditional CBBA to create an initial task assignment. If there are tasks that were unable to be assigned, the teams attempt to correct this by implementing the CBBA between the teams.

61.3.1 Local Task Assignment

To create the initial task assignment, each team runs the CBBA introduced in Choi et al. (2009). This description of the CBBA follows the explanation given in Choi et al. (2010). The CBBA is a decentralized task assignment algorithm which

consists of iterations between two phases: a bundle-building phase where each agent generates an ordered bundle of tasks and a consensus phase where conflicting assignments are resolved through communication with nearby agents.

61.3.1.1 Phase 1 (Build Bundle)

The first phase of the CBBA is bundle construction, during which each agent adds tasks to its bundle until it is unable to add any more tasks. Each agent j in team i maintains four data vectors: a bundle $\mathbf{b}_{ij} \in (\mathcal{N}_{ti} \cup \{\emptyset\})^{L_m}$, the corresponding path $\mathbf{p}_{ij} \in (\mathcal{N}_{ti} \cup \{\emptyset\})^{L_m}$, a winning bid list $\mathbf{y}_{ij} \in \mathfrak{R}_+^{N_{ti}}$, and a winning agent list $\mathbf{z}_{ij} \in \mathcal{U}^{N_{ti}}$. Tasks in the bundle are ordered by decreasing marginal scores, while those in the path are ordered based on the order they will be performed. The marginal score of a task is defined as follows: Let $S_{ij}(\mathbf{p}_{ij})$ be defined as the total reward for agent j in team i performing the tasks within \mathbf{p}_{ij} . If a task k is added to the bundle \mathbf{b}_{ij} , it has the marginal score improvement $c_{ijk}(\mathbf{b}_{ij}) = \max_{n \leq |\mathbf{p}_{ij}|} S_{ij}(\mathbf{p}_{ij} \oplus_n \{k\}) - S_{ij}(\mathbf{p}_{ij})$ where $|\cdot|$ is the cardinality of the list and \oplus_n is the operation that inserts the second list after the n -th element of the first list. The bundle is updated by $\mathbf{b}_{ij} = \mathbf{b}_{ij} \oplus_{|\mathbf{b}_{ij}|} \{K_{ij}\}$ with $K_{ij} = \operatorname{argmax}_k (c_{ijk}(\mathbf{b}_{ij}) \times \prod (c_{ijk} > y_{ijk}) \times h_{ijk})$ where $\prod(\cdot)$ is one if the argument is true and zero otherwise, and h_{ijk} is one if agent j can perform task k and zero otherwise. The path is updated by $\mathbf{p}_{ij} = \mathbf{p}_{ij} \oplus_{n_{ijK_{ij}}} \{K_{ij}\}$ where $n_{ijK_{ij}} = \operatorname{argmax}_n (S_{ij}^{\mathbf{p}_{ij} \oplus_n \{K_{ij}\}})$.

61.3.1.2 Phase 2 (Conflict Resolution)

In the conflict resolution phase, three data vectors are transmitted to neighboring agents. Two were used in the bundle construction phase: the winning bid list \mathbf{y}_{ij} and the winning agent list \mathbf{z}_{ij} . The third data vector $\mathbf{s}_{ij} \in \mathfrak{R}^{N_{ai}}$ represents the time stamp of the last communication from each of the other agents within the same team. When agent j receives a message from agent m , it uses \mathbf{z}_{ij} , \mathbf{s}_{ij} , \mathbf{z}_{im} and \mathbf{s}_{im} to determine which agent's information is the most up to date for each task. There are three possible actions agent j can perform on task k : (1) update: $y_{ijk} = y_{imk}$, $z_{ijk} = z_{imk}$; (2) reset: $y_{ijk} = 0$, $z_{ijk} = \emptyset$; and (3) leave: $y_{ijk} = y_{ijk}$, $z_{ijk} = z_{ijk}$.

61.3.2 Task Sharing

After the initial plan has been created, the teams communicate with each other to determine if other teams can perform some of their tasks. This phase of the TCBBA has two variations. In the first version, the initial plan developed by the CBBA is fixed and each team attempts to insert the unassigned tasks into the initial plan. In the second version, the initial plan is only used to determine which tasks were able to be assigned. The plan itself is ignored. In both versions, each team is required to maintain the same data vectors. Furthermore, both versions of the TCBBA have three stages: initialization, creating the bundle, and communication and conflict resolution. The initialization stage is the only stage that is different between the two variations of the TCBBA.

61.3.2.1 Version 1: Insert in Path Initialization

This version of the TCBBA keeps the initial plan that was developed by the CBBA. Algorithm 1 describes how all of the data structures are initialized to accomplish this. After the initial plan has been created, each team identifies which of its allocated tasks were assigned to one of its agents, $\hat{\mathcal{N}}_{ti}$, and which of its tasks were unassigned, $\hat{\mathcal{N}}_{ti} = \mathcal{N}_{ti} \setminus \hat{\mathcal{N}}_{ti}$. Teams exchange their lists of unassigned tasks to come up with the complete list of unassigned tasks $\mathcal{T}^u = (\hat{\mathcal{N}}_{t1} \cup \hat{\mathcal{N}}_{t2} \cup \dots \cup \hat{\mathcal{N}}_{tN_o})$ (see (A1 : 1), where (A1 : 1) refers to line 1 of Algorithm 1). The team bundle $\mathbf{B}_i \in (\mathcal{T}^u \cup \{0\})^{L_m}$, winning bid list $\mathbf{Y}_i \in \mathbb{R}_+^{|\mathcal{T}^u|}$, and winner ID list $\mathbf{Z}_i \in \mathcal{U}^{|\mathcal{T}^u|}$ are initialized to empty values because they only include information on tasks in \mathcal{T}^u (A1 : 3–5). The team time-stamp vector, \mathbf{S}_i , is initialized to zeros. In addition, all of the agents' data vectors are enlarged while maintaining the same information to account for the unassigned tasks (A1 : 6–13).

After the initialization is done, it iterates between the bundle construction stage, described in Algorithm 3, and the communication and conflict resolution stage, described in Algorithms 4 and 5, until consensus has been achieved (A1 : 14–17).

Keeping the initial plan ensures that the final plan is relatively close to the original one. During the build bundle phase (Algorithm 3), the CBBA will attempt to insert the unassigned tasks into the plan without disrupting the current plan. The task will only be inserted if all of the tasks later on in the path can still be completed. This guarantees that any task that was assigned during the initial assignment will still be assigned in the final plan.

Algorithm 1: TCBBA version 1 initialization for team i

input: $\hat{\mathbf{b}}_i, \hat{\mathbf{p}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i, \mathcal{N}_{ti}$
output: $\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathbf{b}_i, \mathbf{p}_i, \mathbf{y}_i, \mathbf{z}_i$

- 1: $\mathcal{T}^u = \text{unassignedTasks}(\mathbf{b}_i, \mathcal{N}_{ti})$
- 2: $\mathcal{T}_i = \mathcal{N}_{ti} \setminus \mathcal{T}^u$
- 3: $\mathbf{B}_i = \{\}$
- 4: $\mathbf{Y}_i = \text{zeros}(|\mathcal{T}^u|)$
- 5: $\mathbf{Z}_i = \text{zeros}(|\mathcal{T}^u|)$
- 6: **foreach** $j \in \mathcal{N}_{ui}$ **do**
- 7: $\mathbf{b}_{ij} = \hat{\mathbf{b}}_{ij}$
- 8: $\mathbf{p}_{ij} = \hat{\mathbf{p}}_{ij}$
- 9: $\mathbf{y}_{ij} = \text{zeros}(|\mathcal{T}_i \cup \mathcal{T}^u|)$
- 10: $\mathbf{z}_{ij} = \text{zeros}(|\mathcal{T}_i \cup \mathcal{T}^u|)$
- 11: $\mathbf{y}_{ijk} = \hat{\mathbf{y}}_{ijk} \quad \forall k \in \mathcal{T}_i$
- 12: $\mathbf{z}_{ijk} = \hat{\mathbf{z}}_{ijk} \quad \forall k \in \mathcal{T}_i$
- 13: **end**
- 14: **while** *no consensus* **do**
- 15: $[\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathbf{b}_i, \mathbf{p}_i, \mathbf{y}_i, \mathbf{z}_i] = \text{buildBundle}(\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathcal{T}^u, \mathbf{b}_i, \mathbf{p}_i, \mathbf{y}_i, \mathbf{z}_i, \mathcal{T}_i)$
- 16: $[\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathbf{b}_i, \mathbf{p}_i, \mathbf{y}_i, \mathbf{z}_i] = \text{communicate}(\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathbf{b}_i, \mathbf{p}_i, \mathbf{y}_i, \mathbf{z}_i)$
- 17: **end**

Algorithm 2: Changes to Algorithm 1 for TCBBA version 2 initialization

```

1: foreach  $j \in \mathcal{N}_{ui}$  do
2:    $\mathbf{b}_{ij} = \{\}$ 
3:    $\mathbf{p}_{ij} = \{\}$ 
4:    $\mathbf{y}_{ij} = \text{zeros}(|\mathcal{T}_i \cup \mathcal{T}^u|)$ 
5:    $\mathbf{z}_{ij} = \text{zeros}(|\mathcal{T}_i \cup \mathcal{T}^u|)$ 
6: end

```

Algorithm 3: TCBBA build bundle for team i

```

input:  $\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathcal{T}^u, \hat{\mathbf{b}}_i, \hat{\mathbf{p}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i, \mathcal{T}_i$ 
output:  $\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathbf{b}_i, \mathbf{p}_i, \mathbf{y}_i, \mathbf{z}_i$ 

1:  $[\mathbf{b}_i, \mathbf{p}_i, \mathbf{y}_i, \mathbf{z}_i] = \text{CBBA}(\hat{\mathbf{b}}_i, \hat{\mathbf{p}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i, \mathcal{T}_i \cup \mathcal{T}^u)$ 
2: foreach  $j \in \mathcal{N}_{ui}$  do
3:   for  $k = 1$  to  $|\mathbf{b}_{ij}|$  do
4:     if  $\mathbf{b}_{ijk} \in \mathcal{T}^u$  then
5:        $n = |\mathbf{B}_i|$ 
6:       for  $m = |\mathbf{B}_i|$  to 1 do
7:         if  $\mathbf{Y}_{i\mathbf{B}_m} < \mathbf{y}_{ij\mathbf{b}_{ijk}}$  then
8:            $n = m - 1$ 
9:         end
10:       $\mathbf{B}_i = \mathbf{B}_i \oplus_n \mathbf{b}_{ijk}$ 
11:    end
12:     $\mathbf{Z}_{i\mathbf{b}_{ijk}} = i$ 
13:     $\mathbf{Y}_{i\mathbf{b}_{ijk}} = \mathbf{y}_{ij\mathbf{b}_{ijk}}$ 
14:  end
15: end
16: end

```

61.3.2.2 Version 2: Create New Plan Initialization

Modifying Algorithm 1 to ignore the initial plan is very straightforward. The data vectors \mathbf{p}_{ij} , \mathbf{b}_{ij} , \mathbf{y}_{ij} , and \mathbf{z}_{ij} are initialized to empty or zero (A2 : 2–5) instead of the initial plan (A1 : 8–9, 12–13).

This version of the TCBBA does not guarantee that all the tasks that were previously assigned in the initial assignment will be completed. Instead, this version will assign tasks that will maximize its score, even if it has to not perform some previously assigned tasks. The score can be improved further by running Algorithm 2 multiple times recomputing the unassigned tasks each time. This allows the teams to trade tasks to improve the overall score in exchange for more computation time and communication steps.

Algorithm 4: Conflict resolution for team i receiving from team j at time τ

input: $\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i, \mathcal{T}^u$
output: $\mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i$

```

1: SEND:  $\mathbf{Y}_i, \mathbf{Z}_i, \mathbf{S}_i$  to team  $j$ 
2: RECEIVE:  $\mathbf{Y}_j, \mathbf{Z}_j, \mathbf{S}_j$  from team  $j$ 
3: foreach  $k \in \mathcal{T}^u$  do
4:    $\hat{\mathbf{Y}}_i = \mathbf{Y}_{ik}, \hat{\mathbf{Z}}_i = \mathbf{Z}_{ik}$ 
5:    $\hat{\mathbf{Y}}_j = \mathbf{Y}_{jk}, \hat{\mathbf{Z}}_j = \mathbf{Z}_{jk}$ 
6:   Find action (Table 61.1)  $\Rightarrow$  UPDATE, RESET, or LEAVE
   ( $\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j, \hat{\mathbf{Z}}_i, \hat{\mathbf{Z}}_j, \mathbf{S}_i, \mathbf{S}_j$ )
7:   if UPDATE then
8:      $\mathbf{Y}_{ik} = \hat{\mathbf{Y}}_j, \mathbf{Z}_{ik} = \hat{\mathbf{Z}}_j$ 
9:   else if RESET then
10:     $\mathbf{Y}_{ik} = 0, \mathbf{Z}_{ik} = 0$ 
11:   end
12:   if UPDATE or RESET then
13:     for  $m = 1$  to  $|\mathbf{B}_i|$  do
14:       if  $\mathbf{B}_{im} = k$  then
15:          $\mathbf{Y}_{i, \mathbf{B}_{in}} = 0, \mathbf{Z}_{i, \mathbf{B}_{in}} = 0, \mathbf{B}_{in} = \emptyset \quad \forall m \leq n \leq L_m$ 
16:       end
17:     end
18:   end
19: end
20: for  $k = 1$  to  $N_o$  do
21:   if  $k = j$  then
22:      $\mathbf{S}_{ik} = \tau$ 
23:   else
24:      $\mathbf{S}_{ik} = \max(\mathbf{S}_{ik}, \mathbf{S}_{jk})$ 
25:   end
26: end
27: foreach  $j \in \mathcal{N}_{ui}$  do
28:   UpdateAgentList( $j, \mathbf{B}_i, \mathbf{Y}_i, \mathbf{Z}_i$ )
29: end

```

61.3.3 Phase 1: Build Bundle

First, the bids for the unassigned tasks and plans for each agent must be computed. This is done by running a CBBA between all of the agents within the team (A3 : 1). After the agent bundles have been built, the information must be extracted from the agents and put into the team data structures. To do this, it goes through each agent's bundle looking for tasks $\{k\} \in \mathcal{T}^u$ and putting them in the team's bundle \mathbf{B}_i in order of decreasing bid (A3 : 6–12). Sorting the bundle is necessary to satisfy

Algorithm 5: Conflict resolution for team i agent j

input: $\mathbf{B}_i, \mathcal{T}^u, \mathbf{b}_{ij}, \mathbf{p}_{ij}, \mathbf{y}_{ij}, \mathbf{z}_{ij}$
output: $\mathbf{b}_{ij}, \mathbf{p}_{ij}, \mathbf{y}_{ij}, \mathbf{z}_{ij}$

```

1: for  $k = 1$  to  $|\mathbf{b}_{i,j}|$  do
2:   if  $\mathbf{b}_{ijk} \in \mathcal{T}^u$  and  $\mathbf{b}_{ijk} \notin \mathbf{B}_i$  then
3:      $\mathbf{y}_{i,j,\mathbf{b}_{ijm}} = 0, \mathbf{z}_{i,j,\mathbf{b}_{ijm}} = 0$ 
4:      $\forall k \leq m \leq |\mathbf{b}_{ij}| \ \mathbf{b}_{ijm} = \emptyset \quad \forall k \leq m \leq |\mathbf{b}_{ij}|$ 
5:     Update  $\mathbf{p}_{ij}$ : For each task that was removed from  $\mathbf{b}_{ij}$  remove it from  $\mathbf{p}_{ij}$ 
6:   end
7: end

```

the diminishing marginal gain requirement mentioned in Choi et al. (2009). The winning bid list is set to the agent winning bid and the winner ID list is set to i (A3 : 13–14).

61.3.4 Phase 2: Communication

The communication phase consists of three steps: sending and receiving data, resolving conflicts in the team's bundle, and resolving conflicts in the agents' bundles. For team i , the team sends $\mathbf{Y}_i, \mathbf{Z}_i$, and \mathbf{S}_i and receives $\mathbf{Y}_h, \mathbf{Z}_h$, and \mathbf{S}_h from team h . It iterates through each task and determines what action to take. The possible actions are the same used in the CBBA and are repeated in Table 61.1.

Table 61.1 Action rule for agent i receiving from agent j (Choi et al. 2009)

| Sender's \hat{z}_j | Receiver's \hat{z}_i | Condition(s) | Action (Default: LEAVE) |
|-----------------------|-----------------------------|--|-------------------------|
| j | i | if $\hat{y}_j > \hat{y}_i$ | UPDATE |
| | j or 0 | | UPDATE |
| | $\eta \notin \{i, j\}$ | if $\mathbf{S}_{j\eta} > \mathbf{S}_{i\eta}$ or $\hat{y}_j > \hat{y}_i$ | UPDATE |
| i | i or 0 | | LEAVE |
| | j | | RESET |
| | $\eta \notin \{i, j\}$ | if $\mathbf{S}_{j\eta} > \mathbf{S}_{i\eta}$ | RESET |
| $\mu \notin \{i, j\}$ | i | $\mathbf{S}_{j\mu} > \mathbf{S}_{i,\mu}$ and $\hat{y}_j > \hat{y}_i$ | UPDATE |
| | j | if $\mathbf{S}_{j\mu} > \mathbf{S}_{i\mu}$ | UPDATE |
| | | else | RESET |
| | μ or 0 | if $\mathbf{S}_{j\mu} > \mathbf{S}_{i\mu}$ | UPDATE |
| | $\eta \notin \{i, j, \mu\}$ | if $\mathbf{S}_{j\mu} > \mathbf{S}_{i\mu}$ and $\mathbf{S}_{j\eta} > \mathbf{S}_{i\eta}$ | UPDATE |
| | | if $\mathbf{S}_{j\mu} > \mathbf{S}_{i\mu}$ and $\hat{y}_j > \hat{y}_i$ | UPDATE |
| | | if $\mathbf{S}_{j\eta} > \mathbf{S}_{i\eta}$ and $\mathbf{S}_{i\mu} > \mathbf{S}_{j\mu}$ | RESET |
| 0 | i or 0 | | LEAVE |
| | j | | UPDATE |
| | $\eta \notin \{i, j\}$ | if $\mathbf{S}_{j\eta} > \mathbf{S}_{i\eta}$ | UPDATE |

There are three possible outcomes from the decision table (A4 : 6): update, reset, or leave. An update for team i means team i 's information is changed to what team h knows (A4 : 8). A reset re-initializes the bid and winner ID to zero (A4 : 10), and the task is removed from the bundle (A4 : 15–16). A leave makes no change to any of the data vectors. If an update or a reset occurred on task k , then the reward values for the tasks that appear in the bundle after task k are no longer valid and all of the following tasks in the bundle need to be reset (A4 : 15–16). The time-stamp vector is then updated using the current time and \mathbf{S}_j (A4 : 21–27).

After the team information \mathbf{B}_i , \mathbf{Y}_i , and \mathbf{Z}_i has been updated, the individual agent's information needs to be updated (A4 : 26–28). Each agent $j \in \mathcal{N}_{ui}$ goes through its entire bundle \mathbf{b}_{ij} looking for any tasks that were removed from the team's bundle (A5 : 1–2). If it finds one then, for that task and each task that follows it in the bundle, the bid and the winner ID are reset to zero (A5 : 3) and the tasks are removed from the bundle and the path (A5 : 4–5).

61.4 Convergence Time Analysis

It is informative to determine the maximum number of communication steps required for all teams to arrive at a conflict-free solution in a static communication network and worst case scenario. Such analysis cannot be made on Algorithm 2 since it requires knowledge of the number of tasks assigned by each team during each iteration.

In the worst case situation, the diameter of the communication network equals the number of agents. Let D_i be the diameter of the communication network of the agents in team i , N_i be the number of tasks assigned to team i , D_T be the diameter of the team communication network, N_u be the total number of unassigned tasks after the local task assignment phase, N be the total number of tasks, and D be the diameter of the communication network of all the agents.

Choi et al. (2009) showed that the maximum number of communication steps needed for the CBBA to converge to a conflict-free solution is ND . Applying this result to the TCBBA, the number of communications steps needed in the worst case for the initial task assignment is $\max_i(N_i D_i)$. The second stage will have at most $N_u D_T$ iterations, and each iteration will have $\max_i N_u D_i$ communication steps between agents as well as one communication step between teams. The total number of communication steps is

$$N_c = \max_i(N_i D_i) + D_T N_u \left(\max_i(N_u D_i) + 1 \right). \quad (61.3)$$

In the worst case, $\max_i(N_u D_i) \gg 1$ so

$$N_c \approx \max_i(N_i D_i) + D_T N_u^2 \max_i(D_i). \quad (61.4)$$

Assuming that the agents and tasks are divided up equally among the N_o teams then

$$N_i = \frac{N}{N_o} \quad \text{and} \quad D_i = \frac{D}{N_o},$$

and (61.4) becomes

$$N_c = \frac{ND}{N_o^2} + \frac{D_T N_u^2 D}{N_o}. \quad (61.5)$$

Assuming worst case communication between the teams then $N_o = D_T$ and (61.5) becomes

$$\frac{ND}{N_o^2} + N_u^2 D = \left(\frac{N}{N_o^2} + N_u^2 \right) D. \quad (61.6)$$

Comparing (61.6) to the worst case convergence bound for the full CBBA assignment ND gives

$$\left(\frac{N}{N_o^2} + N_u^2 \right) D \leq ND, \quad (61.7)$$

and solving (61.7) for N_u gives the following:

$$N_u \leq \sqrt{1 - \frac{1}{N_o^2}} \sqrt{N}. \quad (61.8)$$

Notice as the number of teams increases, the number of allowed unassigned tasks, N_u approaches \sqrt{N} .

Given the assumption of full communication between the teams in lieu of worst case communication then $D_T = 1$ and (61.5) becomes

$$\frac{ND}{N_o^2} + \frac{N_u^2 D}{N_o} = \left(\frac{N}{N_o^2} + \frac{N_u^2}{N_o} \right) D. \quad (61.9)$$

Comparing (61.9) to the worst case convergence bound for the full CBBA assignment ND yields

$$\left(\frac{N}{N_o^2} + \frac{N_u^2}{N_o} \right) D \leq ND, \quad (61.10)$$

where, as with the analysis of above, (61.10) is solved for N_u to obtain

$$N_u \leq \sqrt{N_o - \frac{1}{N_o}} \sqrt{N}. \quad (61.11)$$

Figure 61.1 shows how the maximum number of unassigned tasks change as the number of teams increase while the number of tasks and agents stay the same. Notice in the worst case team communication case, the maximum number of

Fig. 61.1 Maximum number of unassigned tasks for TCBBA to converge quicker than CBBA in worst case as the number of teams changes ($N_t = 150$, $D = 12$)

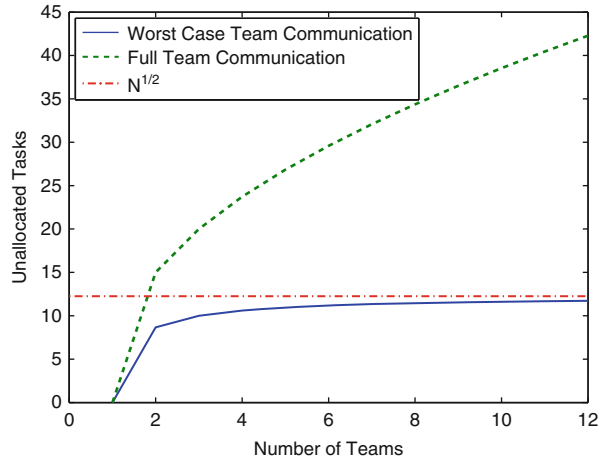
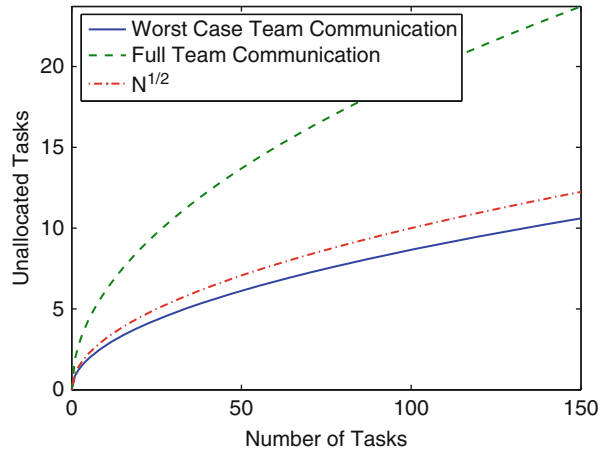


Fig. 61.2 Maximum number of unassigned tasks for TCBBA to converge quicker than CBBA in worst case as the number of tasks changes ($N_o = 4$, $D = 12$)



unassigned tasks quickly approaches the maximum value of $\sqrt{N_t}$. In the full team communication case, the number of unassigned tasks quickly exceeds $\sqrt{N_t}$.

Figure 61.2 shows how the maximum number of unassigned tasks changes as the number of tasks increase while the number of teams and agents stay the same. Notice that both the worst case team communication and full team communication increase proportionally to $\sqrt{N_t}$.

61.5 Simulation Results

Four scenarios are used to compare the TCBBA to the CBBA. In each scenario there are three teams of four agents within $1,200 \times 400$ m world. Each team is assigned N_u tasks which take 25 s to complete. There are four types of tasks

$\{A, B, C, D\}$ and four types of agents $\{a, b, c, d\}$. Each agent type is only able to service its corresponding task type and is modeled as a point-mass with a constant speed of $|\mathbf{v}| = 40$ m/s and no turning constraints, i.e., the agent's position is given by

$$\dot{\mathbf{r}} = \mathbf{v},$$

where the direction of \mathbf{v} points toward the next task to be visited. The time-discounted reward for the tasks is

$$S_i^{\mathbf{p}_i} = \sum \lambda_j^{\tau_i^j(\mathbf{p}_i)} c_j, \quad (61.12)$$

where $S_i^{\mathbf{p}_i}$ is the total score of agent i with path \mathbf{p}_i , $\lambda_j = 0.001$ is the discounting factor for task j , $\tau_i^j(\mathbf{p}_i)$ is the estimated time that agent i will service task j when following path \mathbf{p}_i , and $c_j = 100$ is the value of the task. The objective is the maximize the total score over 500 s.

The four scenarios are created by changing two parameters. The first parameter is if the teams have their own region of the map or if they are in the same overlapping region. If the teams are not overlapping, then they each have their own 400×400 m region with team 2 located in the center. If the teams are overlapping then the teams' agents and tasks are distributed randomly throughout the entire world. The second parameter is if there are 50 or 100 tasks per team.

Teams 1 and 3 do not include all types of agents, so they will be forced to cooperate with other teams in order to accomplish their tasks. Team 1 has two type a agents and two type b agents, team 2 has one of each type of agent, and team 3 has two type c agents and two type d agents. Each team is assigned an equal number of each task type which are randomly distributed within the team's assigned region. Providing an incomplete set of agents to teams 1 and 3 guarantees there will be unassigned tasks after the initial CBBA, it also guarantees that the proportion of unassigned tasks to total tasks will be higher than the threshold developed in the previous section. This almost guarantees that the TCBBA will take more communication steps than the CBBA.

Five hundred Monte-Carlo simulations are run for each scenario. In each simulation, the tasks are assigned using five different methods: running Algorithm 1 once, running Algorithm 2 once, running Algorithm 2 until the change in score is less than 2.5 %, running Algorithm 2 until either all the tasks have been assigned or the unassigned tasks list stops changing, and merging all the tasks and agents into one team and running the CBBA. Figure 61.3 shows the mean percent difference in total score between the four TCBBA versions and the CBBA assignment. Figure 61.4 shows the mean percent difference in communication steps needed to arrive at a conflict free solution between the TCBBA and the CBBA. Figure 61.5 shows the mean percent difference in computation time between the TCBBA and CBBA.

The results show several interesting things about the TCBBA. First, as seen in Fig. 61.3, all the variations of the TCBBA consistently have a lower overall score than the CBBA. This difference varies based on the which TCBBA variation is used

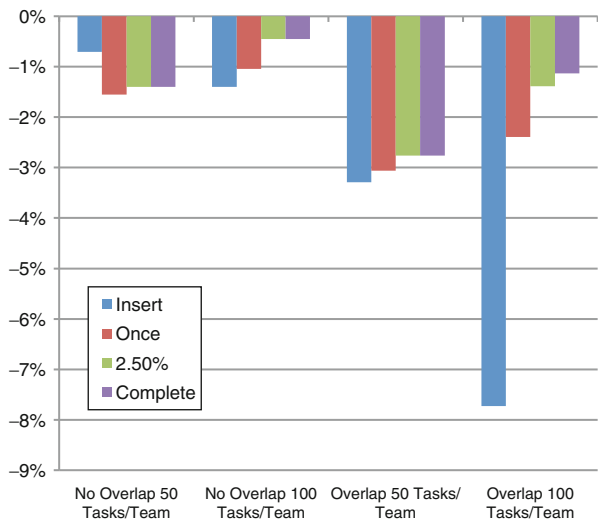


Fig. 61.3 Mean percent difference in overall score between TCBBA and CBBA

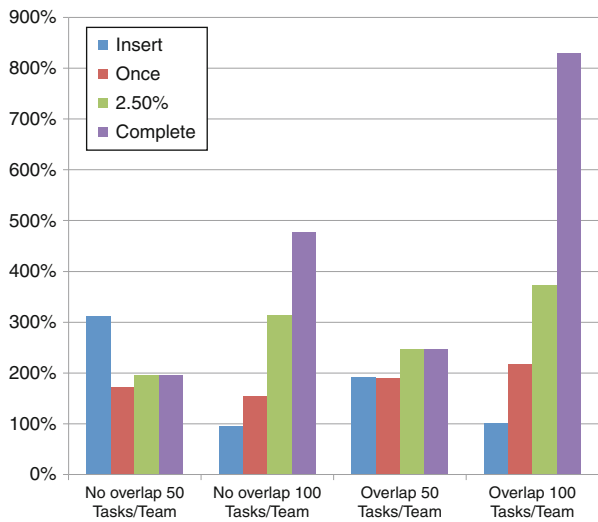


Fig. 61.4 Mean percent difference in communication steps between TCBBA and CBBA

but is typically within 3 %. What is not shown is that occasionally the TCBBA can do as well as the CBBA but in all 500 runs, it never beat it. Second, the TCBBA always required more communication steps than the CBBA as shown in Fig. 61.4. This result was expected because of the large number of unassigned tasks. Third, there is a point of diminishing returns, such that repeating the second variation of TCBBA until the unassigned task list is empty or unchanging is, generally, not worth

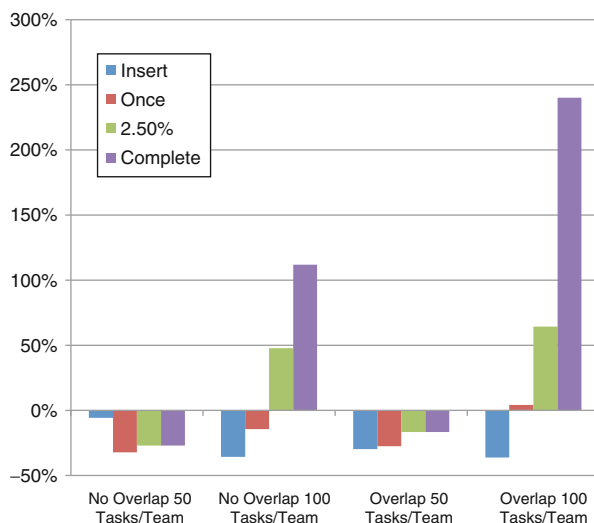


Fig. 61.5 Mean percent difference in computation time between TCBBA and CBBA

the computation. Finally, the amount of computation time required is proportional to the number of communication steps as seen in Fig. 61.5. Because the larger problem is partitioned into a set of smaller problems in the TCBBA, it is interesting to note that the TCBBA can take less time to arrive at a solution than the CBBA when implemented in a centralized manner.

Conclusion

This chapter presents a modification of the consensus-based bundle algorithm that handles the multi-team, multitask problem using a combination of auctions and consensus to achieve feasible, conflict free solutions. While these solutions are less optimal than running the CBBA, typically within 3 %, the TCBBA allows the agents to be divided into separate teams, where, following current rules of engagement, responsibility for each agent/team can be kept under the supervision of a human supervisor.

References

- M. Argyle, D.W. Casbeer, R. Beard, *Proceedings of the American Control Conference*, San Francisco, CA (IEEE, New York, 2011), pp. 5376–5381
- C. Baudin, S. Kedar, B. Pell, *Knowl. Acquis.* **6**(2), 179–196 (1994)
- R.E. Bellman, *Dynamic Programming* (Princeton University Press, Princeton, 1957)
- R. Bonasso, D. Kortenkamp, T. Whitney, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, Providence, Rhode Island (AAAI MIT, Palo Alto/Cambridge, 1997), pp. 949–956
- H. Chao, Y. Cao, Y. Chen, *Int. J. Control Autom. Syst.* **8**(1), 36–44 (2010)

- H.L. Choi, L. Brunet, J. How, *IEEE Trans. Robot.* **25**(4), 912–926 (2009)
- H.L. Choi, A.K. Whitten, J. How, *Proceedings of the American Control Conference*, Baltimore, MD (IEEE, New York, 2010), pp. 3057–3062
- K. Christoffersen, D.D. Woods, *Advances in Human Performance and Cognitive Engineering Research*, vol. 2 (Emerald Group Publishing Limited, Bingley, 2004), pp. 1–12
- J.W. Crandall, M.L. Cummings, M.D. Penna, P.M.A. de Jong, *IEEE Trans. Syst. Man Cybern.* **41**(2), 385–397 (2011)
- M.L. Cummings, S. Bruni, in *Handbook of Automation*, ed. by S.Y. Nof (Springer, New York/Dordrecht, 2009), pp. 437–447
- M.L. Cummings, J. How, A.K. Whitten, O. Toupet, *Proc. IEEE* **100**(3), 660–671 (2012)
- G. Dorais, R. Bonasso, D. Kortenkamp, B. Pell, D. Schreckenghost, *Proceedings of the First International Conference of the Mars Society*, vol. 1 (The Mars Society, Lakewood, 1998), pp. 369–378
- B. Grosz, S. Kraus, in *Foundations and Theories of Rational Agency*, ed. by A. Rao, M. Wooldridge (Kluwer Academic, Dordrecht, 2003)
- R.M. Kilgore, K.A. Harper, C.E. Nehme, M.L. Cummings, *Proceedings AIAA Infotech@Aerospace*, Sonoma, CA (AIAA, Reston, 2007)
- M. Lewis, H. Wang, S.Y. Chien, P. Velagapudi, P. Scerri, K.P. Sycara, *Hum. Factors* **52**(2), 225–233 (2010)
- B. Pell, S. Sawyer, D.E. Bernard, N. Muscettola, B. Smith, *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO, vol. 2 (IEEE, New York, 1998), pp. 289–313
- S.J. Rasmussen, T. Shima, *Int. J. Robust Nonlinear Control* **18**(2), 135–153 (2008)
- P. Scerri, D.V. Pynadath, M. Tambe, *J. Artif. Intell. Res.* **17**, 171–228 Norwell, Massachusetts (2002)
- P. Scerri, D. Pynadath, L. Johnson, P. Rosenbloom, M. Si, N. Schurr, M. Tambe, *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne, VIC (ACM, New York, 2003a), pp. 433–440
- P. Scerri, D.V. Pynadath, M. Tambe, in *Agent Autonomy*, ed. by H. Hexmoor, C. Castelfranchi, R. Falcone, G. Weiss, vol. 7 (Kluwer Academic Publishers, Norwell, 2003b), pp. 211–241
- T. Shima, S.J. Rasmussen (eds.), *UAV Cooperative Decisions and Control: Challenges and Practical Approaches* (SIAM, Philadelphia, 2009)
- K.P. Sycara, G. Sukthankar, Tech. Rep., Robotics Institute, Carnegie Mellon University, 2006
- K.P. Sycara, M. Lewis, in *Team Cognition: Understanding the Factor that Drive Process and Performance*, ed. by E. Salas, S. Fiore (American Psychological Association, Washington DC, 2004)
- K.P. Sycara, M. Paolucci, J. Giampapa, M. van Velsen, *Auton. Agents Multi-agent Syst.* **7**(1), 29–48 (2003)
- K.P. Sycara, K. Decker, A. Pannu, M. Williamson, D. Zeng, *IEEE Expert Intell. Syst. Appl.* **2**(6), 36–46 (2006)

Hyondong Oh, Hyo-Sang Shin, Seungkeun Kim, Antonios Tsourdos,
and Brian A. White

Contents

| | | |
|--------|--|------|
| 62.1 | Introduction | 1510 |
| 62.2 | Path Planning Using Dubins Paths | 1512 |
| 62.2.1 | Generating Dubins Path | 1513 |
| 62.2.2 | Conditions for the Existence of Dubins Paths | 1515 |
| 62.2.3 | Length of Dubins Paths | 1516 |
| 62.3 | Cooperative Mission Planning I: Road-Network Search Route Planning | 1517 |
| 62.3.1 | Road-Network Search Route by Multiple Unmanned Aerial Vehicles | 1518 |
| 62.3.2 | Optimization via MILP | 1519 |
| 62.3.3 | Approximation Algorithm | 1521 |
| 62.3.4 | Numerical Simulations | 1525 |
| 62.4 | Cooperative Mission Planning II: Communication Relay | 1531 |
| 62.4.1 | Waypoint Sequence Decision Making for Communication Relay | 1531 |
| 62.4.2 | Nonflying Zone Constraint Against UAV | 1534 |
| 62.4.3 | Speed Constraint of UAV | 1538 |
| 62.4.4 | Numerical Simulations | 1539 |
| | References | 1543 |

Abstract

This chapter addresses the cooperative mission and path-planning problem of multiple UAVs in the context of the vehicle-routing problem. Since the conventional vehicle-routing algorithms approximate their path to straight lines

H. Oh (✉) • H-S. Shin • A. Tsourdos • B.A. White
Department of Engineering Physics, School of Engineering, Cranfield University, Cranfield,
Bedfordshire, UK
e-mail: h.oh@cranfield.ac.uk; h.shin@cranfield.ac.uk; a.tsourdos@cranfield.ac.uk;
b.a.white@cranfield.ac.uk

S. Kim
Department of Aerospace Engineering, Chungnam National University, Daejeon, South Korea
e-mail: skim78@cnu.ac.kr

to reduce computational load, the physical constraints imposed on the vehicle are not to be taken into account. In order to mitigate this issue, this chapter describes a framework allowing integrated mission and path planning for coordinating UAVs using the Dubins theory based on the differential geometry concepts which can consider non-straight path segments. The main advantage of this approach is that the number of design parameters can be significantly reduced while providing the shortest, safe, and feasible path, which leads to a fast design process and more lightweight algorithms. In order to validate the integrated framework, cooperative mission and path-planning algorithms for two missions are developed: (1) road-network search route-planning patrolling every road segment of interest efficiently based on the optimization and approximation algorithm using nearest insertion and auction negotiation and (2) communication-relay route planning between a ground control station and the friendly fleet satisfying the constraints on the UAV speed and the avoidance of nonflying zones. Lastly, the performance of the proposed algorithms is examined via numerical simulations.

62.1 Introduction

The large scale of UAV applications has proliferated vastly within the last few years with the fielding of Global Hawk, Pioneer, Pathfinder Raven, and Dragoneye UAVs, among others (Samad et al. 2007). The operational experience of UAVs has proven that their technology can bring a dramatic impact to the battlefield. This includes, but is not limited to, obtaining real time, relevant situational awareness before making contact; helping commanders to lead appropriate decision making; and reducing risk to the mission and operation.

Groups of UAVs are of special interest due to their ability to coordinate simultaneous coverage of large areas or cooperate to achieve common goals. The intelligent and autonomous cooperation of multiple UAVs operating in a team/swarm offers revolutionary capabilities: improved situation awareness; significant reductions in manpower and risk to humans; the ability to perform in hostile, hazardous, and geometrically complex environments; and cost efficiency. Specific applications under consideration for groups of cooperating UAVs include, but are not limited to, border patrol, search and rescue, surveillance, mapping, and environmental monitoring. In these applications, a group of UAVs becomes a mobile resource/sensor and, consequently, tasks and routes of each UAV need to be efficiently and optimally planned in order to cooperatively achieve their mission. Therefore, this chapter addresses the cooperative mission and path-planning problem, which here is considered as a vehicle-routing problem of multiple UAVs for given missions.

Since general vehicle-routing algorithms approximate their paths to straight lines in order to reduce the computational load, the physical constraints imposed

on the vehicle are not to be taken into account. In order to mitigate this issue, this chapter describes a framework which allows integrated mission and path planning for coordinating UAVs. One of key enablers of this approach is the path-planning scheme which was developed in a previous study (Tsourdos et al. 2010) based on the differential geometry concepts, especially Dubins paths. Path-planning algorithms using differential geometry examine the evolution of guidance geometry over time to derive curvature satisfying UAV constraints. Guidance commands defining a maneuver profile can be then computed using the derived curvature of the guidance geometry. One of the main advantages of this approach is that the number of design parameters can be significantly reduced while maintaining the guidance performance. Therefore, this approach will enable not only a fast design process and more lightweight algorithms but will also generate safe and feasible paths for multiple UAVs. This is required for the integration of operational and physical constraints of the UAVs into the cooperative mission and path-planning solution.

Road-network search and communication-relay problems are also addressed to validate the integrated framework of mission and path planning. The road-network search routing problem enables multiple airborne platforms to efficiently patrol every road segment identified in the map of interest. This problem has been mainly handled in the operational research area (Gibbons 1999; Ahr 2004; Gross and Yellen 2003; Bektas 2006) and can be generally classified into two categories: one is the traveling salesman problem (TSP) which finds a shortest circular trip through a given number of cities, and the other is the Chinese postman problem (CPP) which finds the shortest path to travel along each road in the road network. The TSP using multiple UAVs can be considered as a task assignment problem to minimize the mission time or energy by assigning each target to an UAV, for which binary linear programming (Bellingham et al. 2003), iterative network flow (Chandler et al. 2002), tabu search algorithm (Ryan et al. 1998), and receding horizon control (Ahmadzadeh et al. 2006) have been proposed. Recently, Royset and Sato (2010) proposed a route optimization algorithm for multiple searchers to detect one or more probabilistically moving targets incorporating other factors such as environmental and platform-specific effects. Meanwhile, the CPP is normally used for ground vehicle applications such as road maintenance, snow disposal (Perrier et al. 2007), boundary coverage (Easton and Burdick 2005), and graph searching and sweeping (Alspach 2006; Parsons 1976).

The communication-relay problem described in this chapter is used to extend the mission area of a main friendly fleet (FF) such as aircraft and ground convoys using relatively inexpensive subsystems, swarms of UAVs. Therefore, the focus is on cooperative route planning of multiple UAVs to maintain communication between the FF and a ground control station (GCS). UAVs in this problem are used as an effective communication-relay platform in environments characterized by poor radio frequency connectivity, which includes urban, forested, or mountainous regions where no line-of-sight exists between ground transmitters and receivers (Cerasoli and Eatontown 2007). In the late 1990s, a feasibility study which uses

the UAV as a communication relay develops a Battlefield Information Transmission System (BITS) system (Pinkney et al. 1996). The main objective of this study was to provide beyond line of sight communications within an area of operations without using scarce satellite resources. The study predicted that, with the advances in miniaturization technology and improved transmitter efficiencies, the UAV could carry multifunction and multiband transponders to handle the communication relay within size, weight, and power dissipation budgets. Cerasoli (Cerasoli and Eatontown 2007) assessed the practical effectiveness of a UAV communication relay in an urban area using a ray tracing method. The paper concluded that a UAV at 2,000m provided coverage for over 90 % of the ground receivers with 10 dB of LOS path loss by analysis of UAVs placed at various positions and heights over an approximately 500-m² urban area.

The remainder of this chapter is organized as follows. Section 62.2 introduces an overview of path planning using Dubins paths and explains how to allow for physical constraints of the fixed-wing UAVs in Dubins path planning. Then, Sect. 62.3 proposes road-network search route-planning algorithms for multiple UAVs based on mixed integer linear programming (MILP) optimization and the approximation algorithm using nearest insertion and auction negotiation. Section 62.4 describes route-planning and decision-making algorithms for a group of UAVs in order to guarantee communication between the GCS and the FF. Simulation results and analysis of each problem are also included in each section. Conclusions are given in Sect. 62.4.4.2.

62.2 Path Planning Using Dubins Paths

Several path-planning methodologies based on differential geometry have been proposed: Dubins curves, clothoid arcs, and Pythagorean hodograph curves (Shanmugavel et al. 2007; Tsourdos et al. 2010). This section will briefly introduce the concept of Dubins path planning based on the reference (Shanmugavel 2007; Kim et al. 2011), and the detailed algorithm will be developed to design flyable and safe Dubins path transiting between waypoints for cooperative missions of multiple UAVs. The Dubins path is considered because it is the shortest path, has simple geometry, and is computationally efficient.

Path planning is defined as the geometric evolution of curves between two desired poses in free space, C_{free} . The pose in 2D comprises the position coordinates (x, y) and the orientation θ . A simple case of producing path between two poses is first considered. This can be extended into any number of waypoints/poses:

$$P_s(x_s, y_s, \theta_s) \xrightarrow{r(q)} P_f(x_f, y_f, \theta_f), \quad r(q) \in C_{\text{free}}, \quad |\kappa(t)| \leq \kappa_{\text{max}} \quad (62.1)$$

where P_s and P_f denote a starting and final pose, respectively, $\kappa(t)$ represents the curvature of path $r(q)$ with parameter q , and κ_{max} is the maximum curvature

imposed by the UAV dynamic constraints such as maximum lateral acceleration. Motion in the plane is composed of either rectilinear or turning or angular motions. A straight line provides the shortest rectilinear motion, and the circular arc provides the shortest turning or angular motion. Also, the arc provides a constant turning radius which also satisfies the maximum curvature constraint, that is, the minimum turning radius, which is a function of speed and maximum lateral acceleration. The Dubins path (Dubins 1957) is the shortest path between two vectors in a plane, and the path meets the minimum bound on turning radius. The Dubins path is a composite path formed either by two circular arcs connected by a common tangent or three consecutive tangential circular arcs or a subset of either of these two. The first path is a CLC path, and the second one is a CCC path, where “C” stands for circular segment and “L” stands for line segment. Either of these two curves will form the shortest path between two poses and so is a good approach for UAV path planning. This section focuses on Dubins paths of CLC type, and the details of CCC type paths can be found in Oh et al. (2011a).

62.2.1 Generating Dubins Path

The Dubins path can be produced by solving (Eq. 62.1). However, it is computationally efficient if it is produced by geometric principles owing to its simple geometry that it is formed by two circular arcs connected by their common tangents. Therefore, the principles of analytic geometry are used to produce the Dubins paths. There are two possible tangents between the arcs: (1) an external tangent, where the start and finish maneuvers have same turning directions, and (2) an internal tangent, where the turning maneuvers have opposite turning directions (e.g., if the start maneuver is clockwise, the finish maneuver will be anticlockwise and vice versa). Here, only the Dubins path with an external tangent is derived (the case of an internal tangent is analogous). Referring to Fig. 62.1, the input parameters for producing the Dubins path are:

- i) Initial pose: $P_s(x_s, y_s, \theta_s)$
- ii) Finish pose: $P_f(x_f, y_f, \theta_f)$
- iii) Initial turning radius: $\rho_s (= \frac{1}{\kappa_s})$
- iv) Finish turning radius: $\rho_f (= \frac{1}{\kappa_f})$

1. Find the centers of the turning circles $O_s(x_{cs}, y_{cs})$ and $O_f(x_{cf}, y_{cf})$:

$$(x_{cs}, y_{cs}) = (x_s \pm \rho_s \cos(\theta_s \pm \pi/2), y_s \pm \rho_s \sin(\theta_s \pm \pi/2)) \quad (62.2a)$$

$$(x_{cf}, y_{cf}) = (x_f \pm \rho_f \cos(\theta_f \pm \pi/2), y_f \pm \rho_f \sin(\theta_f \pm \pi/2)) \quad (62.2b)$$

where O_s and O_f are called primary circles represented by C_s and C_f , respectively.

2. Draw a secondary circle of radius $|\rho_f - \rho_s|$ at O_f for $\rho_s \leq \rho_f$.

3. Connect the centers O_s and O_f to form a line c , called center line, where $|c| = \sqrt{(x_{cs} - x_{cf})^2 + (y_{cs} - y_{cf})^2}$.
4. Draw a line $O_s T'$ tangent to the secondary circle C_{sec} .
5. Draw a perpendicular straight line from O_f to $O_s T'$, which intersects the primary circle C_f at T_N , which is called as a tangent entry point on C_f .
6. Draw a line $T_X T_N$ parallel to $O_s T'$, where T_X is called as a tangent exit point on C_s .
7. Connect the points P_s and T_X by an arc of radius ρ_s and T_N and P_f by an arc of radius ρ_f .
8. The composite path is then formed by the starting arc $P_s T_X$, followed by the external tangent line $T_X T_N$ and the finishing arc $T_N P_f$.

The triangle $\triangle O_s O_f T'$ is a right-angled triangle with hypotenuse $O_s O_f$, and the other two sides are $O_f T'$ and $O_s T'$, where $\|O_f T'\| = |\rho_f - \rho_s|$. The included angle between $O_s O_f$ and $O_s T'$ is ϕ_e , where

$$\phi_e = \arcsin \left(\frac{\rho_f - \rho_s}{|c|} \right) \quad (62.3)$$

The slope of the line c is ψ , where

$$\psi = \arctan \left(\frac{y_{cf} - y_{cs}}{x_{cf} - x_{cs}} \right) \quad (62.4)$$

The angles $\phi_{ex} = \angle(X_1 O_s T_X)$ and $\phi_{en} = \angle(X_2 O_f T_N)$ are calculated from Table 62.1 where $\overrightarrow{O_s X_1}$ and $\overrightarrow{O_f X_2}$ are parallel to the positive

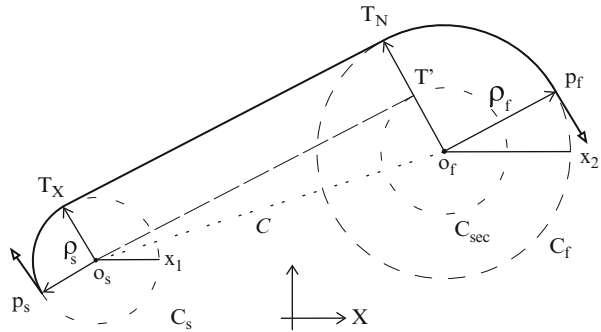
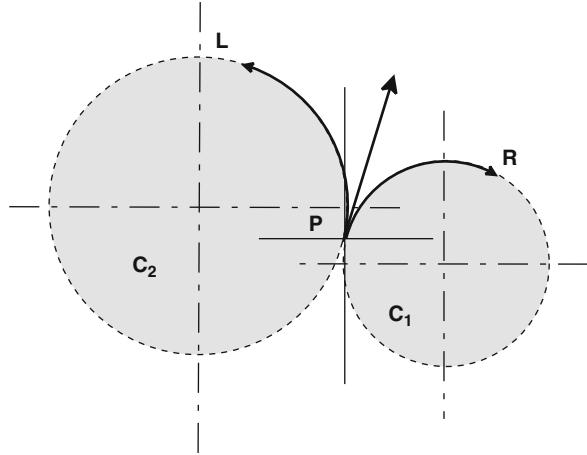


Fig. 62.1 Dubins – design of CLC path

Table 62.1 Calculation of tangent exit and entry points

| Start-turn | Finish-turn | ϕ_e | ϕ_{ex} | ϕ_{en} |
|---------------|---------------|--|---------------------------------|---------------------------------|
| Clockwise | Clockwise | $\arcsin \left(\frac{\rho_f - \rho_s}{c} \right)$ | $\phi_e + \frac{\pi}{2} + \psi$ | $\phi_e + \frac{\pi}{2} + \psi$ |
| Anticlockwise | Anticlockwise | $\arcsin \left(\frac{\rho_f - \rho_s}{c} \right)$ | $\phi_e - \frac{\pi}{2} + \psi$ | $\phi_e - \frac{\pi}{2} + \psi$ |

Fig. 62.2 Tangent circles

direction of x -coordinate. The values of ϕ_{ex} and ϕ_{en} , the tangent exit and entry points, are calculated as

$$T_X = (x_{cs} + \rho_s \cos(\phi_{ex}), y_{cs} + \rho_s \sin(\phi_{ex})) \quad (62.5a)$$

$$T_N = (x_{cf} + \rho_f \cos(\phi_{en}), y_{cf} + \rho_f \sin(\phi_{en})) \quad (62.5b)$$

From the figure, it can be seen that the turning maneuvers have clockwise rotations. Similarly, other type of Dubins path using the internal tangent can be produced by drawing the secondary circle of radius equal to $|\rho_s + \rho_f|$. It is worth pointing out that the calculation of the tangent exit and entry points T_X and T_N is central in generating the Dubins path.

For a given pose, there are two circles tangent to it. Referring to Fig. 62.2, the pose P has a right turn R on the arc C_1 and a left turn L on the arc C_2 . If θ_s and θ_f are fixed, four Dubins paths are possible between P_s and P_f , which are $\{RSR, RSL, LSR, LSL\}$, where represents the tangent. However, if the final orientation is taken either $\pm\theta_f$, the number of Dubins paths between P_s and P_f will become eight. Figure 62.3 shows the eight possible Dubins paths: four paths each from the primary circle C_1 to the secondary circles C_3 and C_4 and four from C_2 to C_3 and C_4 .

62.2.2 Conditions for the Existence of Dubins Paths

From Sect. 62.2.1, it can be seen that the existence of the Dubins path is determined by the common tangents between the turning arcs. These tangents vanish under two conditions. The external tangent vanishes when one of the primary circles C_s and C_f contains the other, while the internal one vanishes when the primary circles intersect. Both of these conditions together determine the existence of the Dubins

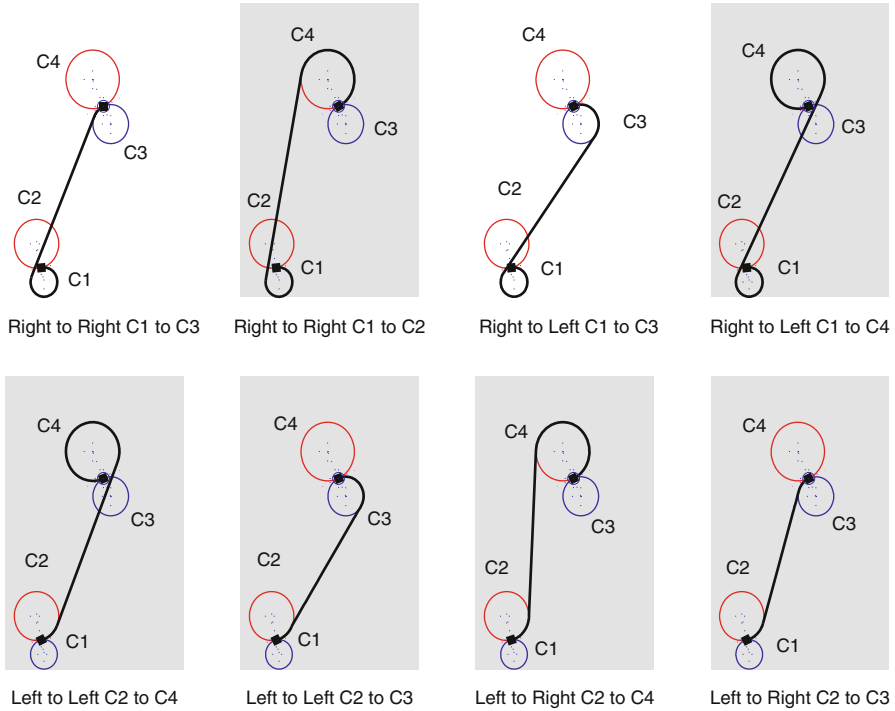


Fig. 62.3 Dubins paths with θ_f as a free variable

path between given two poses, which in turn are a function of the turning radii and hance curvature:

$$\text{External tangent : } (c + \rho_s) > \rho_f, \rho_f > \rho_s \quad (62.6a)$$

$$\text{Internal tangent : } c > (\rho_s + \rho_f), \rho_f > \rho_s \quad (62.6b)$$

62.2.3 Length of Dubins Paths

The Dubins path is a composite path of two circular arcs and a straight line. Hence, the path length s_{Dubins} is the sum of the lengths of individual path segments. Since the length of the common tangent connecting the arcs is determined by the radii of the arcs, the length is also the function of the turning radii. Hence, the length of the path can be varied by changing the radii (curvatures). Also, any two paths can be made equal in length by simply varying the curvature of the arcs:

$$s_{\text{Dubins}} = s_s + s_t + s_f \quad (62.7a)$$

$$= |\rho_s \alpha_s| + \|T_X T_N\| + |\rho_f \alpha_f| \quad (62.7b)$$

$$s_{\text{Dubins}} = f(\rho_s, \rho_f) \quad (62.7c)$$

where s_{Dubins} is the length of the Dubins path, α_s and α_f are the included angles, $\alpha_s = \phi_{ex}$, $\alpha_f = \phi_{en}$, and $\|T_X T_N\| = \sqrt{(y_N - y_X)^2 + (x_N - x_X)^2}$. Using this length of the Dubins path, a reference velocity for each UAV can be computed in order to control the time taken for the UAV to traverse each path for cooperative mission planning.

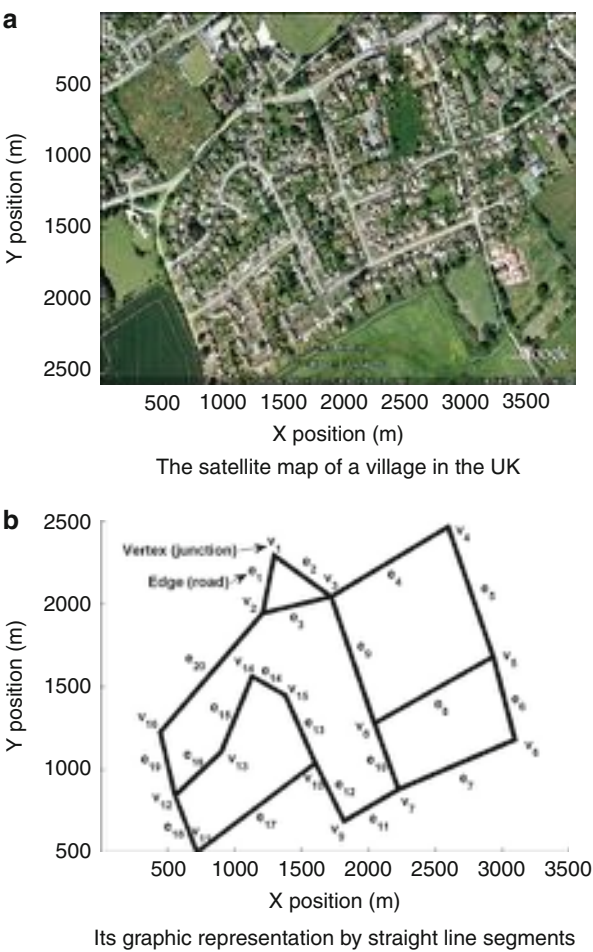
62.3 Cooperative Mission Planning I: Road-Network Search Route Planning

For a road search route-planning mission, a road network is established as a set of straight line connecting a set of waypoints. These waypoints are located either on road junctions or along the roads with sufficient separations between them to allow for accurate representation of a curved road by using a set of straight lines. A sample road network is shown in Fig. 62.4a, which is based on the Google map of a village in the UK. This road network can be translated to a graph composed of straight line segments connecting a set of vertices, as shown in Fig. 62.4b (Oh et al. 2011b). In order to search all the roads within the map of interest, there are generally two typical routing problems (Ahr 2004):

Traveling Salesman Problem (TSP): A salesman has to visit several cities (or road junctions). Starting at a certain city, the TSP finds a route with minimum travel distance on which the salesman traverses each of the destination cities exactly once (and for the closed TSP, leads him back to his starting point). *Chinese Postman Problem (CPP)*: A postman has to deliver mail for a network of streets. Starting at a given point, for example, the post office, the CPP finds a route with minimum travel distance which allows the postman to stop by each street at least once (and for the closed CPP, leading him back to the post office).

Consider the CPP and its variants, which involve constructing a tour of all the roads with the shortest distance of the road network. Typically, the road network is mapped to an undirected graph $G = (V, E)$, having edge weights $w : E \rightarrow R_0^+$, where the roads are represented by the edge set $E = \{e_1, \dots, e_n\}$ and the road junctions are represented by the vertex set $V = \{v_1, \dots, v_m\}$ as numbered in Fig. 62.4b. Each edge $e_i = \{v_{ei,1}, v_{ei,2}\}$ is weighted with its length or the amount of time required to traverse it. The basic CPP algorithm involves first constructing an even graph from the road network which has a set of vertices with an even number of edges attached to them producing a pair of entry and exit points. When the road-network graph has junctions with an odd number of edges, some roads therefore must be selected as exceptions for multiple visits by the postman to make the even graph. The search pattern (tour) of the even graph is calculated by determining the Euler tour of the graph (Gross and Yellen 2003), which visits every edge of the even graph exactly once.

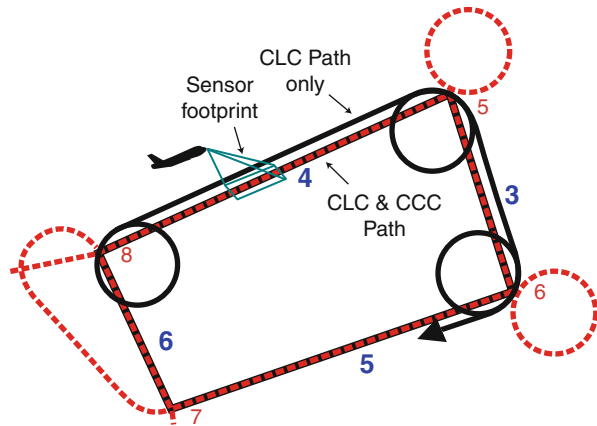
Fig. 62.4 The graphic representation of a road network. **(a)** The satellite map of a village in the UK. **(b)** Its graphic representation by straight line segments



62.3.1 Road-Network Search Route by Multiple Unmanned Aerial Vehicles

The conventional CPP algorithm has been applied to a fully connected road network for use by ground vehicles. However, since UAVs do not have any restrictions such that they must only move along the roads, the CPP algorithm needs to be modified for the case that UAVs search a general road map having unconnected road segments. The modified CPP (mCPP) generates a tour of the road network traveling all the roads once no matter what the type area of interest map is: an even or odd graph. Even searching the area having no road somewhere in it can be tackled by the mCPP algorithm by generating a virtual road pattern with a lawnmower (Maza and Ollero 2007) or spiral-like (Nigam and Kroo 2008) algorithm.

Fig. 62.6 An example of CCC and CLC paths following the road



62.3.2.1 Dubins Path Planning

Once the shortest edge permutations are determined, the next step is to compute and to store the cost (path length or flight time) and to then compute the Dubins paths to connect them. Although the CLC path is being used in general case, this study also explores the CCC path for a densely distributed road environment, because the CLC path cannot be applied in all cases (cf. Sect. 62.2.2). Moreover, to follow the road precisely taking into account the sensor footprint coverage, the path should consist of both CCC and CLC forms of Dubins path. Figure 62.6 shows an example of a road search path using CLC and CCC path for a small sensor footprint, which results in a detour at the road intersection.

62.3.2.2 Generation of the Shortest Edge Permutation

First of all, unordered feasible edge (i.e., road) permutations to be visited by the UAV are generated for all possible cases with a given petal size. The petal size means the maximum number of edges that can be visited by one UAV and is determined by the amount of resources available to it. If the end vertex of one edge and any vertex of the next edge are not connected, they are connected with an additional edge which has a shorter distance. Then, the shortest order-of-visit edge permutations considering the initial position of each UAV are computed under the assumption that a path is represented as a straight line.

62.3.2.3 MMKP Formulation and MILP Optimization

The final step of the proposed algorithm is to allocate the edge permutations to each UAV so as to cover all the roads with a minimum flying time. This can be expressed by a MMKP formulation as

$$\min J = \sum_{i=1}^{N_{\text{UAV}}} \sum_{j=1}^{N_{p_i}} T_{ij} x_{ij} \quad (62.8)$$

$$\text{s.t. } \sum_{i=1}^{N_{\text{UAV}}} \sum_{j=1}^{N_{p_i}} E_{kj} x_{ij} \geq 1, \text{ for } k = 1, \dots, N_{\text{edge}} \quad (62.9)$$

$$\sum_{j=1}^{N_{p_i}} x_{ij} = 1, x_{ij} \in \{0, 1\}, \text{ for } i = 1, \dots, N_{\text{UAV}}, j = 1, \dots, N_{p_i} \quad (62.10)$$

where N_{UAV} , N_{edge} , and N_{p_i} denote the number of UAVs, the edges to be visited, and the permutations generated by the i -th UAV, respectively. T_{ij} represents the mission cost (flight time) of the j -th permutation of the i -th UAV, and E_{kj} represents the matrix whose k -th element of the j -th permutation is 1 if edge k visited; otherwise, 0 and x_{ij} is either 0, implying permutation j of the i -th vehicle is not picked or 1 implying permutation j of the i -th UAV is picked. The first constraint states that the UAVs should visit every edge once or more, and the second constraint represents the allocation of exactly one edge permutation to the each UAV. This MMKP problem is solved using a SYMPHONY MILP solver (Ralphs et al. 2010). It should be noted that, depending on the petal size, the computational burden of the mission cost T_{ij} of all edge permutations can be significant.

62.3.3 Approximation Algorithm

62.3.3.1 Nearest Insertion-Based mDCPP

Due to the complexity of the problem, instead of using the optimization method explained above, an approximation algorithm is developed as a more practical solution to the mCPP (Oh et al. 2011a, 2012). To develop the approximation algorithm, the TSP algorithm is first studied. Although there are a lot of algorithms for the TSP (Rosenkrantz et al. 2009), one heuristic approach, a nearest insertion method, is adopted here since it is fast and easy to implement. The basic idea of the insertion method is to construct the approximate tour by a sequence of steps in which tours are constructed for progressively larger subsets of the nodes of the graph. It produces a tour no longer than twice the optimal regardless of the number of nodes in the problem and runs in a time proportional to the square of the nodes (Rosenkrantz et al. 2009). Having this in mind, the nearest insertion-based mDCPP (NI-mDCPP) algorithm is developed as illustrated in Fig. 62.7. The NI-mDCPP algorithm for the single UAV is described as follows.

Algorithm Description

1. Start from a certain point or road junction and select the nearest road to it using the Dubins path length.
2. Make and grow a tour by finding the nearest road to any of the selected tour roads.

3. Compute the cost of insertion to decide whether to insert before or stack after the closest road to the tour.
4. Insert the selected road in the decided position.
5. 2 ~ 4 are repeated until all roads are included in the tour.

62.3.3.2 Euclidean Distance Order Approximation

To reduce the computation burden further for the dynamic environment, an additional approximation algorithm which uses the Euclidean distance order is incorporated into the NI-mDCPP algorithm. This algorithm is described as follows. First of all, make an ascending order of road list for both the Euclidean distance and the Dubins path length between all pairs of end points of the road network, and find the maximum number, $n_{\text{order,max}}$ which guarantees that road list of Euclidean distance within that number contains the shortest Dubins path. Note that although $n_{\text{order,max}}$ is determined before running the algorithm given information of the road network, a size of $n_{\text{order,max}}$ would remain almost the same against minor changes of road information for an uncertain dynamic environment. Then, when finding the nearest roads, make the ascending order list of distances between the edge of interest and all the other roads using the Euclidean distance first, and find the nearest road whose Dubins path length is the shortest among the roads in the $n_{\text{order,max}}$ Euclidean distance order. In other words, this method computes only $n_{\text{order,max}}$ Dubins path distances instead of computing all the Dubins lengths between one road the other roads.

62.3.3.3 Negotiation for Multiple UAVs

Having developed the NI-mDCPP algorithm for the single UAV, it can be extended to the case of multiple UAVs using auction-based negotiation. The algorithm is described as follows.

Algorithm Description

1. Start with N initial positions or roads of N UAVs.
2. Make and grow a tour by using single NI-mDCPP algorithm while storing the cost (path length or flight time) between selected tour roads and remaining edges (which was needed for finding the nearest vertex).
3. When conflict occurs, that is, more than one UAV wants the same road for the next tour, the auction algorithm using stored cost is used to match UAVs with the task (road) to minimize the cost.

Figure 62.8 illustrates the procedure of the algorithm. Since road 3 is not searched yet in Fig. 62.8d, each UAV sends its cost for the given task (in this case, visiting road 3), and then an auction or bipartite (linear programming) approach is used to match UAVs with the task to minimize the cost. Although overlapping road segments are avoided using the auction algorithm, collision between UAVs might occur during the transition from one road to the next. In that case, if necessary, the path can be replanned either by visiting a new road or by modifying the curvature of the arc of the Dubins path (Tsourdos et al. 2010). For simplicity, it is assumed that the collision avoidance is done by a local flight controller or by operating the UAVs

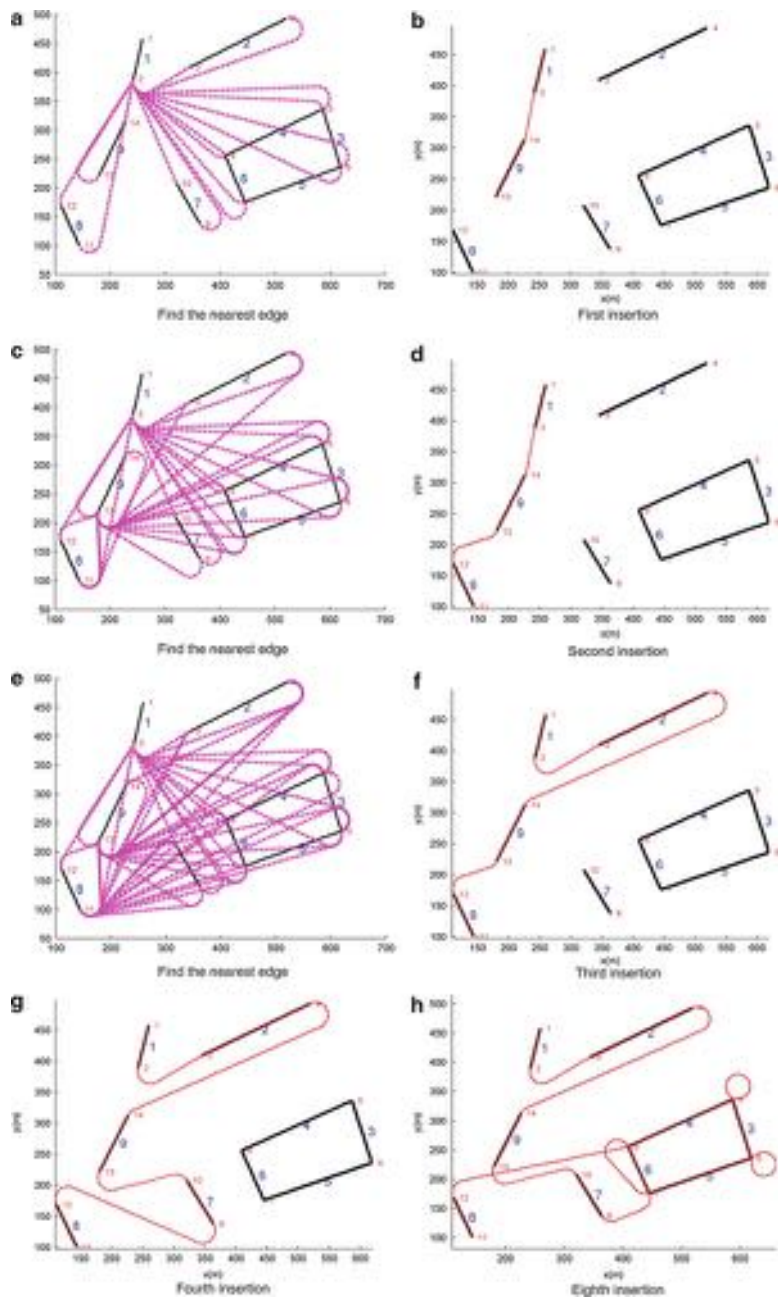


Fig. 62.7 An example simulation of the NI-mDCPP road search algorithm. (a) Find the nearest edge. (b) First insertion. (c) Find the nearest edge. (d) Second insertion. (e) Find the nearest edge. (f) Third insertion. (g) Fourth insertion. (h) Eighth insertion

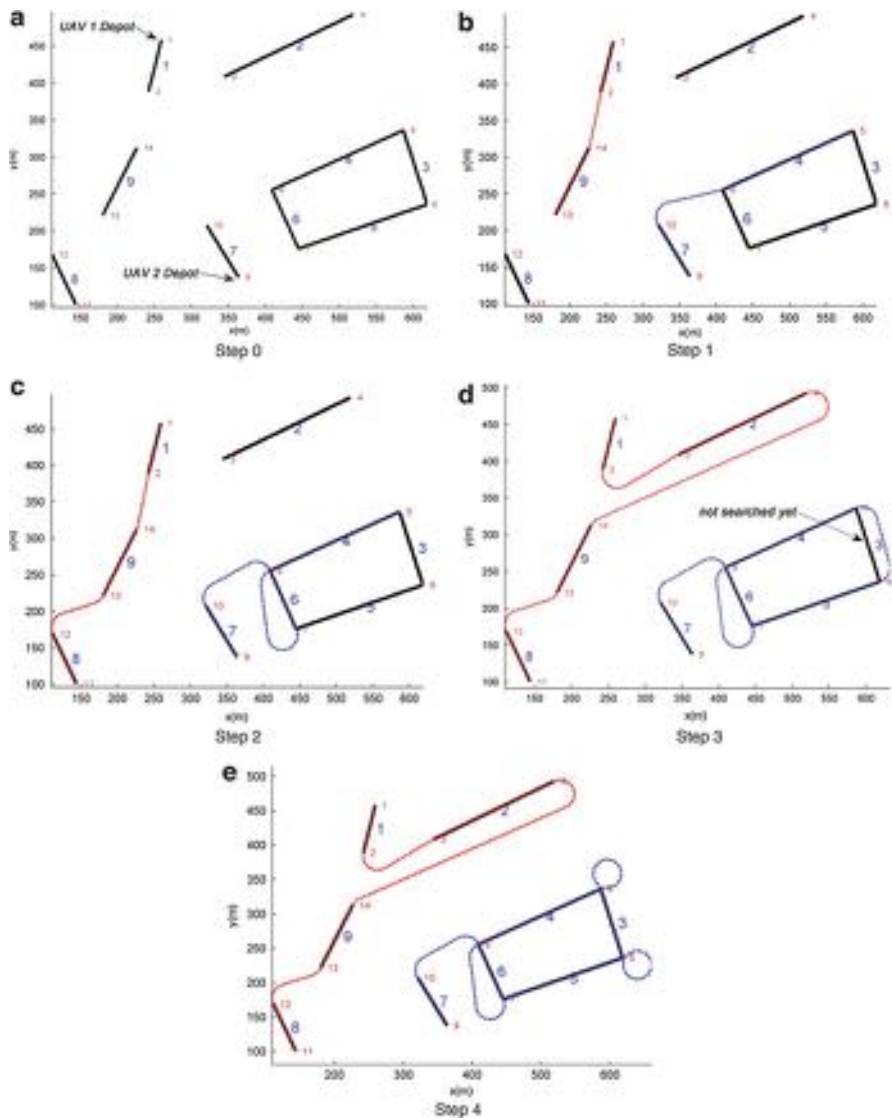


Fig. 62.8 Negotiation procedure for multiple UAVs. (a) Step 0. (b) Step 1. (c) Step 2. (d) Step 3. (e) Step 4

at different altitudes. The proposed algorithm is rather simple but straightforward and can be run in real time. Moreover, by including additional factors such as different minimum turning radii and total path length (or the number of roads) assigned to each UAV into the cost, the auction-based negotiation can be made very flexible and can deal with heterogeneous UAVs.

62.3.4 Numerical Simulations

62.3.4.1 Performance Comparison

To evaluate the performance of the proposed road-network search algorithms for multiple UAVs, numerical simulations are performed for a specific scenario with four UAVs, and the road network shown in Fig. 62.4a. Each UAV has different dynamic constraints given by:

- Minimum turning radius ρ_{\min} : [100 90 80 70] m
- Maximum cruise speed $V_{c,\max}$: [60 50 40 30] m/s

The maximum curvature κ_{\max} of the UAVs are given by $\kappa_{\max} = 1/\rho_{\min}$. The UAVs are assumed to have maximum cruising speed during the entire mission, and the maximum petal size of the edge permutation is set to five. Figure 62.9a shows the result of the road-network search using MILP optimization. The flight path is smooth and flyable due to the Dubins path planning, and since the UAV does not need to only fly along the road, the results include additional paths connecting some of the unconnected roads. The total flight length of all UAVs is 2,798.1 m, and its flight duration is 583.1 s. In this scenario, the total computation time exceeds a reasonable limit (>5 min) using a normal PC system (Core 2 CPU, 2.0 Ghz, and 512 MB RAM). Figure 62.9b shows the search results of the NI-mDCPP algorithm. The NI-mDCPP gives a solution within less than a second having about 12 % longer flight time (653.4 s) than that of the MILP optimization. Considering both the computation time and performance, the NI-mDCPP can be regarded as a preferable approach for the given sample map or a more complex scenario.

62.3.4.2 Monte Carlo Simulation of Approximation Method

To evaluate the properties and performance of the proposed approximation algorithm, Monte Carlo simulations are performed using a random map with different parameters defining the map size and the number of UAVs. The map environment is composed of 10 by 10 vertices, numbered as shown in Fig. 62.10, and the road edges are generated by connecting two randomly selected vertices. To check the impact of the map size on the Dubins path planning, the distance d_{map} between the adjacent vertex is set to be proportional to the minimum turning radius ρ_{\min} of the UAV as

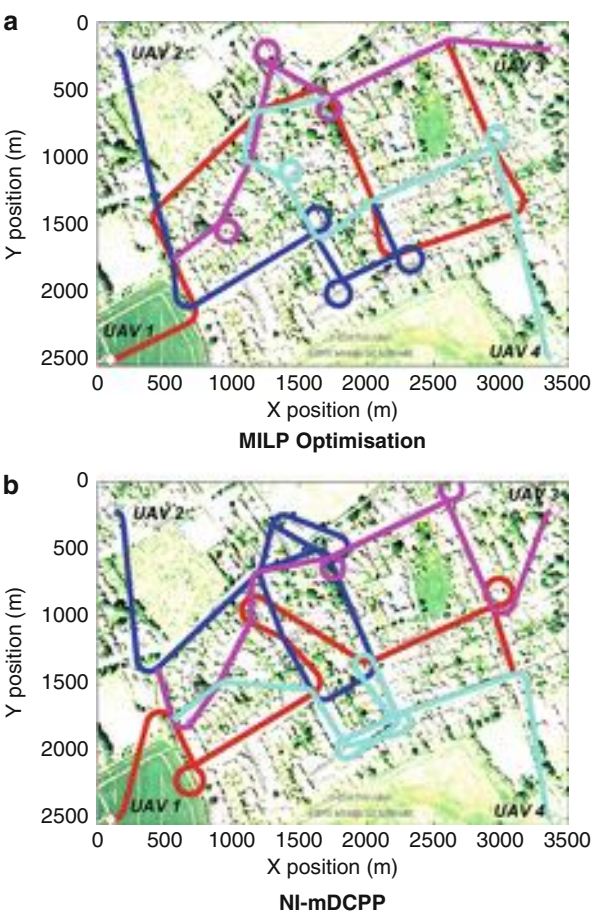
$$d_{\text{map}} = K_s \times \rho_{\min} \quad (62.11)$$

where K_s is the scale factor. Moreover, some of the selected edges whose lengths are longer than three times d_{map} are discarded to get a well-distributed road network and to distribute the roads to each UAV with a similar length. Figure 62.10 shows the sample road network with 20 randomly chosen edges. By Monte Carlo simulations, the effect of three major factors for the road-network search route planning can be investigated. These are:

The distribution density of road network: densely or sparsely distributed road relative to the minimum turning radius of UAVs

The type of path planning: straight line or Dubins path

Fig. 62.9 Road-network search route-planning results using multiple UAVs. **(a)** MILP optimization. **(b)** NI-mDCPP



The number of UAVs: computation time, path length, and the longest path length of a single UAV

This will provide information on priorities for the efficient use of the UAV group in the planning phase of the autonomous search mission.

In the simulations, the UAVs are assumed to have a constant velocity and minimum turning radius of $\rho_{\min} = 50$ m. The simulation results are the average of 50 runs.

Single UAV Case

The first set of simulations are performed using a single UAV with different road map scales. For the rest of this section, the terms of mDCPP and the mECPP are used for the NI-mDCPP and the NI-mECPP, respectively. One of the search route-planning results using the mDCPP for a random map is shown in Fig. 62.11, which covers all the roads satisfying turning constraints of UAVs. Figure 62.12a displays the computation time ratio between the Dubins path (mDCPP) and the

Fig. 62.10 A sample road network with 20 randomly chosen edges
($K_s = 1$, $\rho_{\min} = 50$ m)

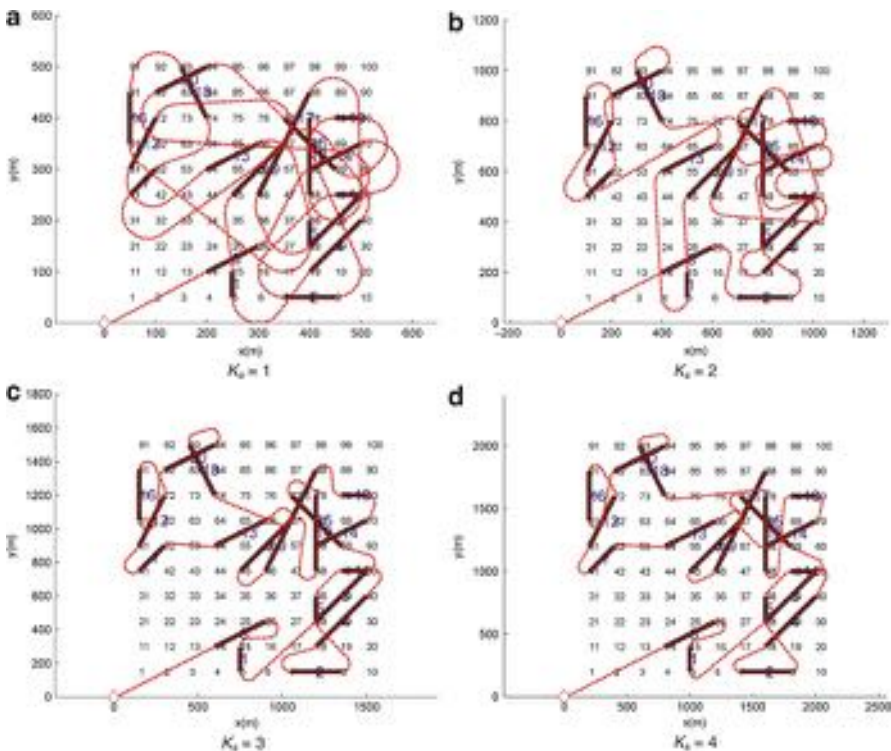
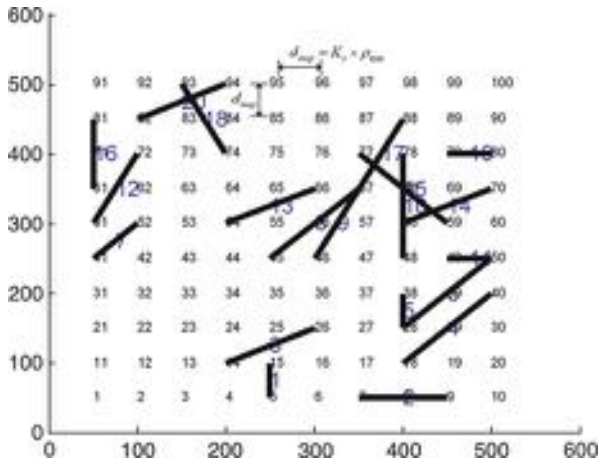


Fig. 62.11 NI-mDCPP road search path with different map scale factors. (a) $K_s = 1$. (b) $K_s = 2$. (c) $K_s = 3$. (d) $K_s = 4$

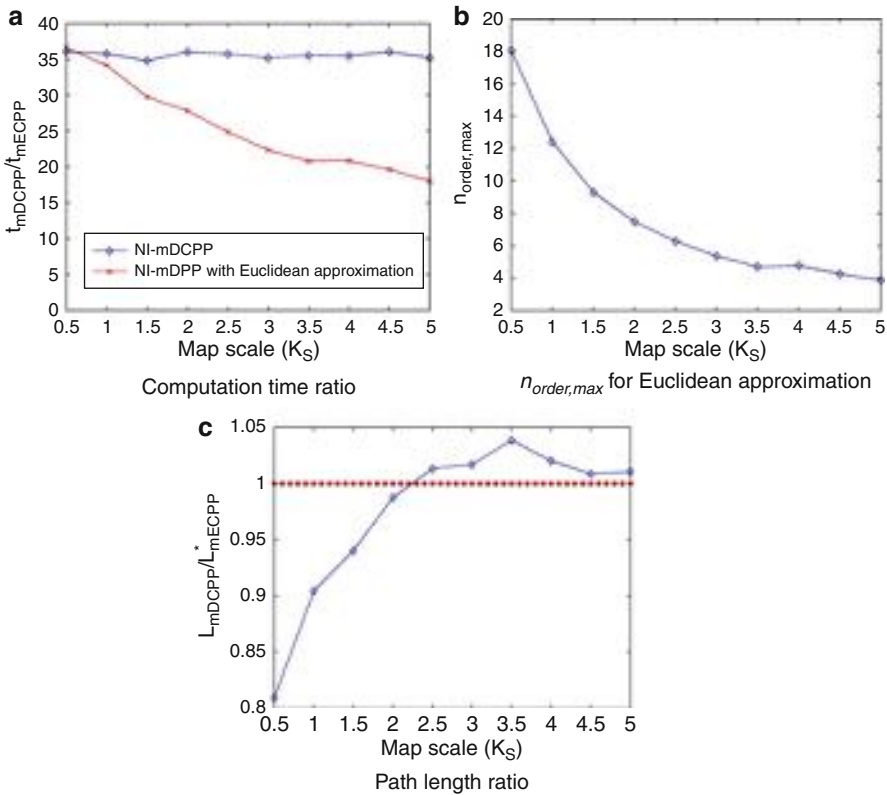


Fig. 62.12 NI-mDCPP results with different map scale factors, average for 50 simulations. (a) Computation time ratio. (b) $n_{order,max}$ for Euclidean approximation. (c) Path length ratio

straight line (mECPP). Regardless of the map scale, the mDcPP algorithm is around consistently 35 times slower than the mECPP. Meanwhile, the computation time of the mDCPP along with the Euclidean distance order approximation, as explained in Sect. 62.3.4, decreases as the map scale increases as a result of a decrease in the maximum order $n_{order,max}$ as shown in Fig. 62.12b. Figure 62.12c compares the total path length to cover the entire road map using the mDCPP and the mECPP. For fair comparison, the length of the mECPP (denoted by L_{mECPP}^*) is computed by road search route using the mECPP algorithm but connecting the roads using a Dubins path. This is because although road search route planning is performed using the mECPP, a real trajectory of the UAV connecting the roads should be of Dubins restricted by its maximum curvature. When the minimum turning radius is relatively small compared to the distance between roads, that is, when the map scale is small, the path length of the mDCPP is shorter than that of the mECPP. However, as the map scale gets bigger, the path length ratio gets closer to one (or even greater than one) since the road search order using Dubins paths is almost the same as using a straight line, as one can expect from Fig. 62.11.

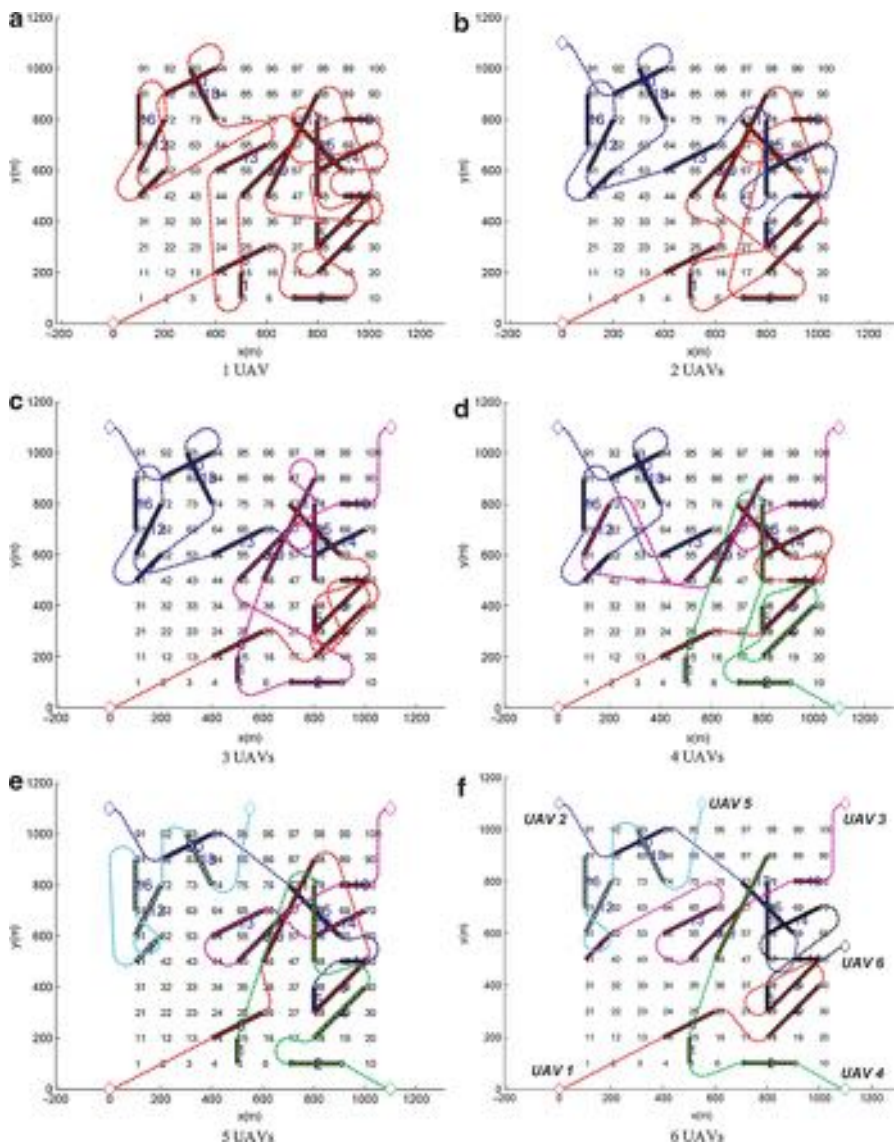


Fig. 62.13 NI-mDCPP road search path with different number of UAVs ($K_s = 2$). (a) 1 UAV. (b) 2 UAVs. (c) 3 UAVs. (d) 4 UAVs. (e) 5 UAVs. (g) 6 UAVs

Multiple UAVs Case

The second simulation is performed using multiple UAVs with different road map scales, and one of the search route-planning results using the mDCPP is shown in Fig. 62.13. The initial position of each UAV is equally distributed around the road area. Figure 62.14 shows the normalized simulation results for

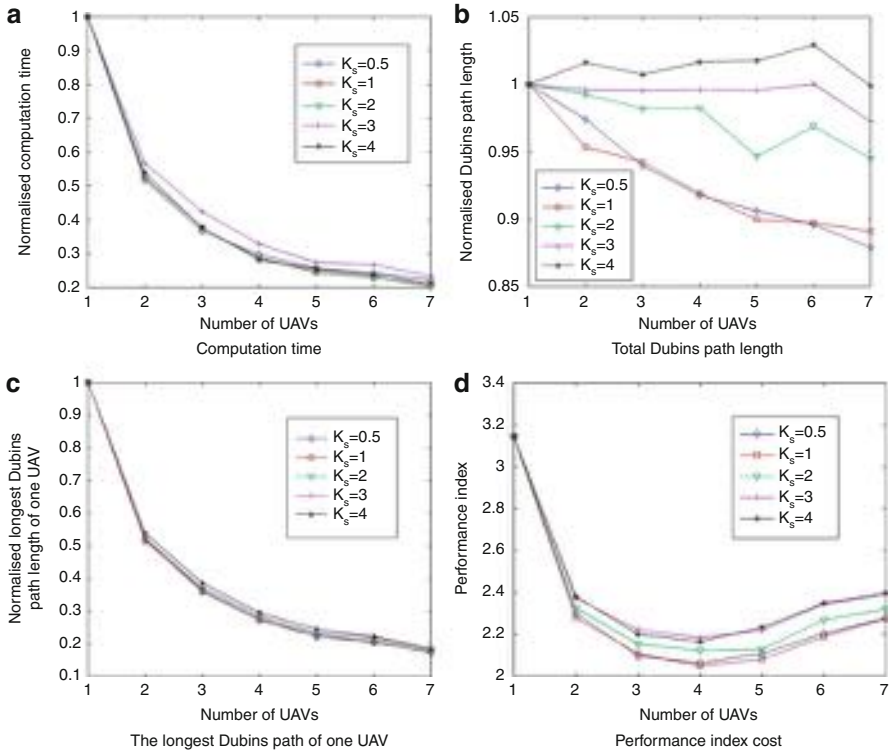


Fig. 62.14 NI-mDCPP results with different number of UAVs ($K_s = 2$), average for 50 simulations. (a) Computation time. (b) Total Dubins path length. (c) The longest Dubins path of one UAV. (d) Performance index cost

a single UAV. In particular, the longest path length (Fig. 62.14c) of the UAV is of interest since it is equivalent to the mission completion time of the entire UAV team. The normalized computation time and the longest path length of the UAV decrease as the size of the UAV team is increased, regardless of the map scale as each UAV takes partial charge of the road search mission cooperatively using the auction-based task assignment. The total path length (Fig. 62.14b) is significantly affected by the map scale. When the map scale is small, the total path length decreases in proportion to the number of UAVs. Whereas in a relatively big map environment, the normalized path length remains close to one since each UAV will fly a long distance from the initial position or from one road to another road. Apparently, the simulation results show that the bigger the UAV team size is, the better performance it shows in terms of the computation time and path length. However, using a large number of UAVs on the team require more operational cost, such as fuel and communication resources. Therefore, the performance index to determine the optimal size of the UAV team for the search mission is proposed as shown in (Eq. 62.12), which includes an additional operational cost for each UAV,

normalized by the number of UAVs (by maximum seven UAVs in this case), \bar{n}_v , and its weighting factor, w_{n_v} :

$$J = w_t \bar{t} + w_l \bar{l} + w_L \bar{L} + w_{n_v} \bar{n}_v \quad (62.12)$$

where w_t , w_l , and w_L represent the weighting factors of computation time, the total path length and the longest path length, of one UAV, respectively. Under the assumption that all of the weighting factors are set to one, the number of UAVs to minimize the performance index J can be selected as four for all map scale factors as shown in Fig. 62.14d.

62.4 Cooperative Mission Planning II: Communication Relay

This section presents the development of trajectory planning for multiple UAVs making use of Dubins path theory in order to maintain and optimize the communication relay between a FF (friendly fleet) performing a main mission and a GCS (ground control station) centrally administrating the whole mission (Kim et al. 2011). To apply Dubins path theory to the optimization of communication between UAVs and with the GCS, various strategic or dynamic constraints have to be considered: mission planning of the FF, the positions of the GCS, any existing nonflying zones, and the limits of UAV dynamics. These constraints affect the decision making of the waypoint sequences of the UAV members. (The final, definitive version of the Section 62.4 of this chapter has been published in Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 225/1, Jan/2011 (<http://pig.sagepub.com/content/225/1/12.abstract>) by SAGE Publications Ltd, All rights reserved. © 2011, Institution of Mechanical Engineers.)

62.4.1 Waypoint Sequence Decision Making for Communication Relay

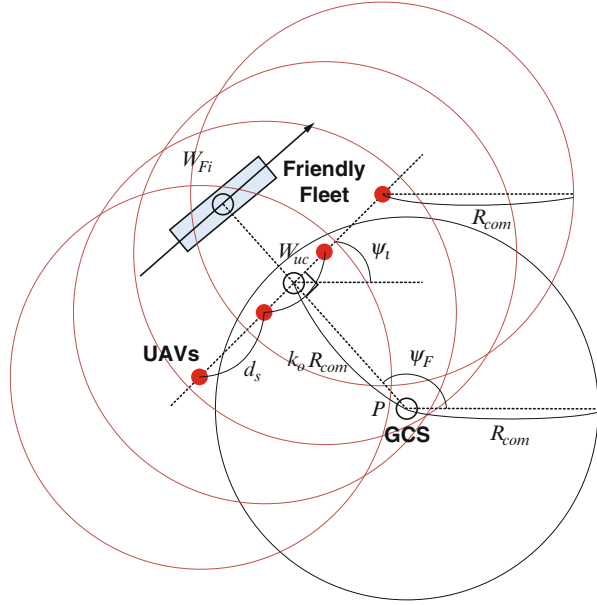
The decision making of the UAVs will be based on the waypoint sequence W_F of the FF which is a priori known since the friendly fleet's future movement can be broadcasted to the UAV swarm and the GCS:

$$W_F = [W_{F0}, W_{F1}, W_{F2}, \dots, W_{Fn}] \quad (62.13)$$

where $W_{Fi} = (x_{Fi}, y_{Fi})^T$ and n is the number of waypoints to be passed by the FF. Awareness of the asset speed V_F makes it possible to get the arrival-time sequence T_F at the waypoint sequence defined in (Eq. 62.13):

$$T_F = [t_{F0}, t_{F1}, t_{F2}, \dots, t_{Fn}] \quad (62.14)$$

Fig. 62.15 Decision making on spreading out the UAV members: the distance from the GCS to the FF is shorter than twice of the communication range $k_o R_{com}$



where t_{Fi} is the arrival time at the i -th waypoint. This can be computed using the speed information of the FF which is assumed constant over the mission. Given these waypoint and arrival-time sequences of the FF, the waypoint sequence of N_u UAVs can be determined by considering communication relay between the GCS and the FF.

Let us consider a ratio of the distance from the GCS position $P = (x_p, y_p)^T$ to the FF over the communication range R_{com} as shown in Fig. 62.15 given as

$$k_F = \max_i \frac{|W_{Fi} - P|}{R_{com}}, \quad i \in \{0, 1, \dots, n\} \quad (62.15)$$

It is assumed that each UAV has the same communication range as the GCS.

Firstly, consider the case where the distance from the GCS to the FF is shorter than $(1 + k_o)$ -times the communication range R_{com} , that is, $k_F \leq (1 + k_o)$ as shown in Fig. 62.15 (k_o is given by (Eq. 62.16)). If the FF's next waypoint and arrival time are given in (Eqs. 62.13) and (62.14), the target center position of UAVs, W_{uc} , is assigned as the intersection between a circle of the communication range adjusted by an overlap coefficient k_o and the line-of-sight from the GCS position to the asset's next waypoint W_{Fi} , where

$$W_{uc} = P + k_o R_{com} [\cos \psi_F, \sin \psi_F]^T \quad (62.16)$$

and where

$$\psi_F = \tan^{-1} \frac{y_{Fi} - y_p}{x_{Fi} - x_p} \quad (62.17)$$

The distribution of UAVs on the boundary of the communication range aims at maximizing the coverage of communication when the FF goes out of the

communication range. Note that $0 < k_o \leq 1$ is adjustable to make the boundaries between the communication circles overlap in order to prevent loss of communication. The choice of the design parameter k_o is made by a trade-off analysis.

A target waypoint of an individual UAV is next determined to distribute all of the swarm members relatively to the target center position W_{uc} . All the UAV members should spread out in a single line, as shown in Fig. 62.15, because the other UAVs enable redundant communication channels in an emergency, although in this case, a single UAV is enough to maintain the communication coverage. The distribution line needs to be perpendicular to the line-of-sight $\overrightarrow{PW_{Fi}}$ in order to maximize the lateral communication coverage. Thus, the azimuth of the distribution line relative to the GCS site, ψ_l , is obtained using the azimuth of the FF relative to the GCS site, ψ_F , computed with (Eq. 62.17) to give

$$\psi_l = \psi_F - \pi/2 \quad (62.18)$$

As a result, the position vector of the j -th UAV is decided by a rotational transformation using the number of UAVs N_u , the angle of the distribution line ψ_l , and the target center position W_{uc} :

$$W_u^j = W_{uc} + \begin{bmatrix} \cos(\psi_l) & -\sin(\psi_l) \\ \sin(\psi_l) & \cos(\psi_l) \end{bmatrix} \begin{pmatrix} (-(N_u - 1)/2 + (j - 1))d_s \\ 0 \end{pmatrix} \quad (62.19)$$

where $j \in \{1, 2, \dots, N_u\}$ and d_s is a separation distance between the UAVs at the target waypoint.

In the general case, consider the case where the distance from the GCS to the FF exceeds $(1 + k_o)$ -times the communication range R_{com} , that is, $k_F > (1 + k_o)$ in (Eq. 62.15). To maintain the communication relay, the distribution of the UAVs has to be restructured since a single UAV cannot link the communication between the FF and the GCS. This study adopts a restructuring policy such that some of the UAV members can move to communication-relay positions having the shape of a chain defined by k_F . First, choose the number of UAVs N_m that should move to the new communication-relay positions in order not to lose the communication as

$$N_m = \langle \frac{k_F}{k_o} - 1 \rangle \quad (62.20)$$

where $\langle x \rangle$ rounds the value of x to the nearest integer toward minus infinity. Note that the closer that k_F to k_o is, the fewer UAVs are needed, but the smaller the overlapping areas are. For example, when the FF flies to the waypoint defined by $2k_o \leq k_F < 3k_o$ with the help of a four-UAV swarm, N_m becomes 1; hence, only the fourth UAV member of the swarm should move to the new relay waypoint point W_{ur}^4 , as shown in Fig. 62.16. In a similar manner, generalize the relay position of W_{ur}^k by

$$W_{ur}^k = W_{Fi} + \begin{bmatrix} \cos(\psi_F) & -\sin(\psi_F) \\ \sin(\psi_F) & \cos(\psi_F) \end{bmatrix} \begin{pmatrix} -(N_u - k + 1)k_o R_{com} \\ 0 \end{pmatrix} \quad (62.21)$$

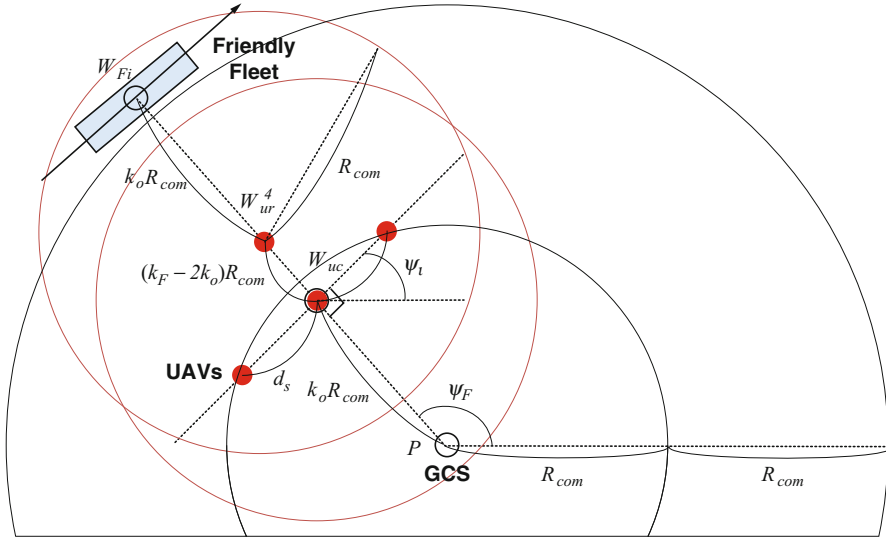


Fig. 62.16 Decision making on spreading out the UAV members: the distance from the GCS to the FF is longer than twice of the communication range $k_o R_{com}$

where $k \in \{N_u - N_m + 1, N_u - N_m + 2, \dots, N_u\}$. In short, N_m number of UAVs move to the relay positions W_{ur}^k as defined in (Eq. 62.21), while the other $N_u - N_m$ number of UAVs are distributed on the line as defined in (Eq. 62.19). In this way, the communication between the GCS and the FF can be maintained in virtue of the reconstruction of the UAV formation no matter that the FF maneuvers beyond the communication boundary of the GCS.

Finally, the following waypoint and arrival time sequences of the j -th UAV can be obtained using (Eqs. 62.19), (62.21), and (62.14):

$$W_U^j = [W_{u0}^j, W_{u1}^j, W_{u2}^j, \dots, W_{un}^j] \quad (62.22)$$

$$T_U^j = [t_{u0}^j, t_{u1}^j, t_{u2}^j, \dots, t_{un}^j] \quad (62.23)$$

where $t_{ui}^j = t_{ai}$ since the UAV has to arrive at the same time as the assets to protect them effectively.

62.4.2 Nonflying Zone Constraint Against UAV

In a battlefield or an urban area, a NFZ (nonflying zone) could exist as an obstacle dangerous to a swarm of UAVs generated by an enemy, a high building/structure, or an airport space. When a threat is detected, the path planner generates an intermediate waypoint so that the threat is avoided. Consider a flyable trajectory $r(q)$ generated for a given set of poses/waypoints. The schematic of the concept

Fig. 62.17 Threat handling by intermediate pose

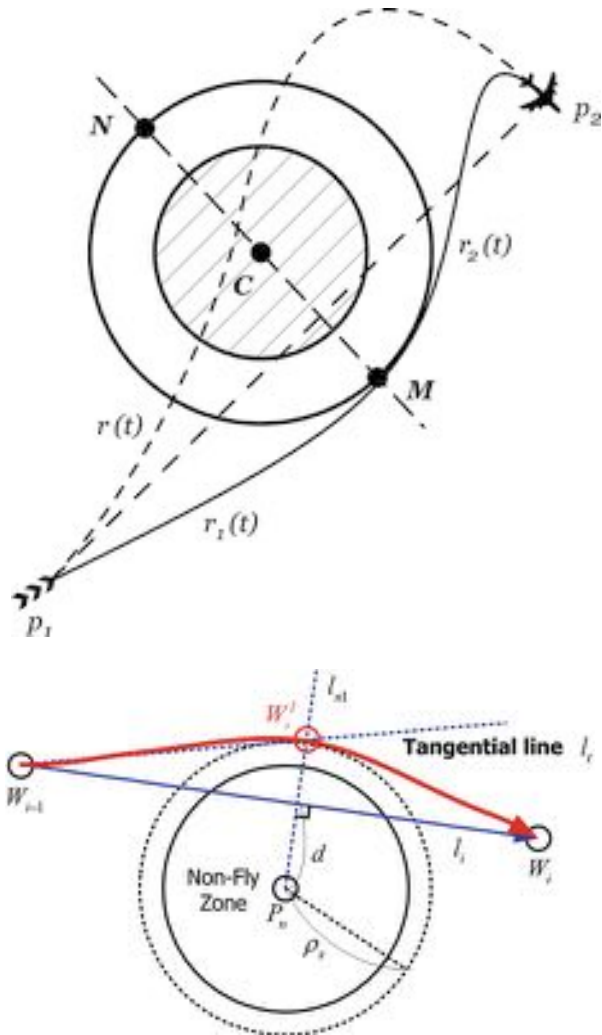


Fig. 62.18 Case 1: The line between the target waypoints crosses the NFZ

is shown in Fig. 62.17. In the figure, the central hatched circle is the threat. The intermediate waypoints are generated by first drawing a line between the current waypoint p_1 and the next waypoint p_2 . If a line orthogonal to this is constructed, it will intersect the safety circle at two points N and M . These are then designated as the potential intermediate waypoints. If the center C is left to the line p_1p_2 , the intermediate waypoint M is selected on the right-hand side of the threat region and vice versa. Consider two cases relating to the nonflying zone avoidance.

62.4.2.1 Case 1: The Line Between the Target Waypoints Crosses the NFZ

The first case happens when the line between the current waypoint W_{i-1} and the next waypoint W_i cuts a nonflying zone as shown in Fig. 62.18.

Define a vector of positions of target waypoints, the intermediate waypoint to be designed, and the nonflying zone center by

$$W_{i-1} = [x_{i-1}, y_{i-1}]^T \quad (62.24a)$$

$$W_i = [x_i, y_i]^T \quad (62.24b)$$

$$W_i^I = [x_i^I, y_i^I]^T \quad (62.24c)$$

$$P_n = [x_n, y_n]^T \quad (62.24d)$$

The following equation of a line l_i is obtained with a simple geometrical relation between two successive waypoints, W_{i-1} , W_i :

$$y - y_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(x - x_i) \quad (62.25)$$

To check the violation of the NFZ, the distance from the NFZ center to the line l_i is calculated as

$$d = \frac{|(y_i - y_{i-1})(x_n - x_i) - (x_i - x_{i-1})(y_n - y_i)|}{\sqrt{(y_i - y_{i-1})^2 + (x_i - x_{i-1})^2}} \quad (62.26)$$

If $d < \rho_s$, a intermediate target waypoint W_i^I needs to be defined for the UAV to make a detour around the nonflying zone. For this, the tangential line l_t is required, and its gradient can be obtained using the following second-order polynomial equation which represents the relationship between the tangential line l_t and the safety circle radius ρ_s :

$$am^2 + bm + c = 0 \quad (62.27)$$

where the coefficients are defined as

$$a = (x_n - x_{i-1})^2 - \rho_s^2 \quad (62.28a)$$

$$b = 2(x_n - x_{i-1})(y_{i-1} - y_n) \quad (62.28b)$$

$$c = (y_{i-1} - y_n)^2 - \rho_s^2 \quad (62.28c)$$

This equation has two roots, and the root m_t closest to the gradient of the line l_i is chosen. The resulting equation of the line l_t is given by

$$y - y_{i-1} = m_t(x - x_{i-1}) \quad (62.29)$$

The equation of the line l_{n1} is defined using the property that it is perpendicular to the line l_i and is located at the NFZ center P_n :

$$y - y_n = m_{n1}(x - x_n) \quad (62.30)$$

where $m_{n1} = -\frac{x_i - x_{i-1}}{y_i - y_{i-1}}$. Finally, the intermediate target waypoint W_i^I is defined by the intersecting point of the lines l_t and l_{n1} . Solving (Eqs. 62.29) and (62.30) together gives

$$x_i^I = \frac{m_t x_{i-1} - y_{i-1} - m_{n1} x_n + y_n}{m_t - m_{n1}} \quad (62.31a)$$

$$y_i^I = m_t(x_i^I - x_{i-1}) + y_{i-1} \quad (62.31b)$$

$$x_i^I = \frac{m_t x_{i-1} - y_{i-1} - m_{n1} x_n + y_n}{m_t - m_{n1}} \quad (62.32a)$$

$$y_i^I = m_t(x_i^I - x_{i-1}) + y_{i-1} \quad (62.32b)$$

62.4.2.2 Case 2: The Next Target Waypoint Exists Inside the NFZ

For this case, a replacement target waypoint W_i^S needs to be defined as the next target waypoint exists inside the NFZ as shown in Fig. 62.19. In a similar manner to case 1 in Sect. 62.4.2.1, the line l_t tangential to the NFZ safety circle can be obtained using (Eqs. 62.27)–(62.29). Then, define a position vector of the substitute target waypoint as

$$W_i^S = [x_i^S, y_i^S]^T \quad (62.33)$$

The equation of the line l_{n2} is defined using the property that it is perpendicular to the line l_t and goes through the next waypoint W_i :

$$y - y_i = m_{n2}(x - x_i) \quad (62.34)$$

where $m_{n2} = -1/m_t$. Finally, the replacement target waypoint W_i^S is defined as the intersection point of lines l_t and l_{n2} . Solving (Eqs. 62.29) and (62.34) together gives

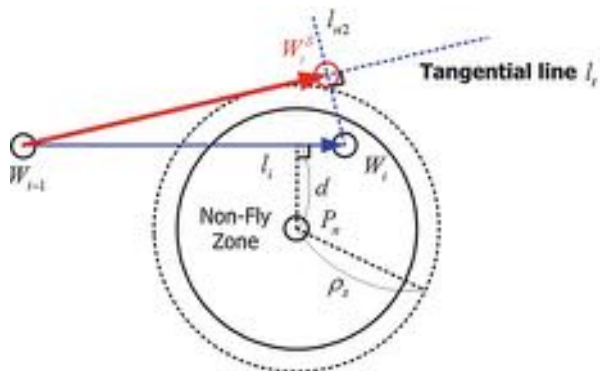
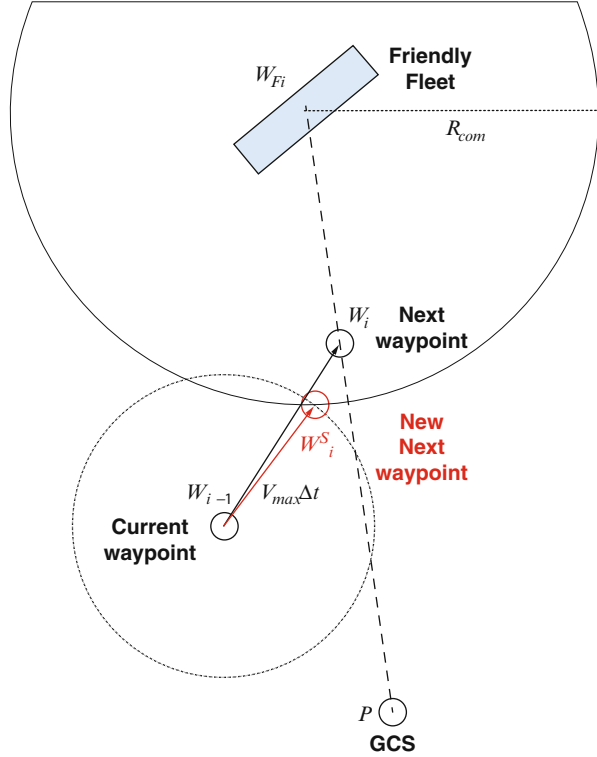


Fig. 62.19 Case 2: The next target waypoint exists inside the NFZ

Fig. 62.20 Decision making concept on speed constraints: An intersection position exists (*Left*) and does not exist (*Right*) between the maximum arrival circle of the UAV and the protection line



$$x_i^S = \frac{m_t x_{i-1} - y_{i-1} - m_{n2} x_i + y_i}{m_t - m_{n2}} \quad (62.35a)$$

$$y_i^S = m_t (x_i^S - x_{i-1}) + y_{i-1} \quad (62.35b)$$

62.4.3 Speed Constraint of UAV

The UAVs have a maximum speed constraint V_{\max} ; thus, at times they cannot arrive at the next target waypoint W_i in time from the previous waypoint W_{i-1} , as shown in Fig. 62.20. In this case, the next target waypoint has to be replaced by a new waypoint W_i^S . The decision making considers only the case in which there exists an intersection between the maximum arrival circle of the UAV and the communication range of the FF. To check existence of an intersection between the area of the maximum arrival circle and the next waypoint, the distance from the current waypoint W_i to the next waypoint W_{i+1} is calculated as

$$d = |W_i - W_{i+1}|. \quad (62.36)$$

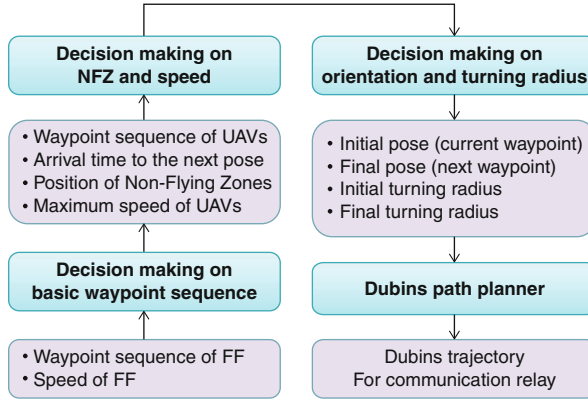


Fig. 62.21 The overall block diagram of the proposed decision-making and path-planning methodology

The radius of the maximum arrival circle is computed as $V_{\max} \Delta t$ where Δt is defined as $t_{ui}^j - t_{u(i-1)}^j$ using (Eq. 62.23). If $d > V_{\max} \Delta t$, an intersection position does not exist between the area of the maximum arrival circle of the UAV and the next waypoint, as shown in Fig. 62.20. In this case, since the UAV has to go inside the FF's communication range in order to maintain the communication relay, the new waypoint W_i should be determined by the intersection between the maximum arrival circle of the UAV and the communication range of the FF.

The starting and final positions have been defined to generate the Dubins path between the target waypoints of the UAVs. To complete the design of the Dubins path, the orientation also need be designed. The orientation of the UAVs between the i -th and $(i + 1)$ -th waypoints is synchronized with that of the friendly fleet in order to keep the line-of-sight stable from the GCS to the FF as

$$\theta_i = \tan^{-1} \frac{y_{Fi} - y_{F(i-1)}}{x_{Fi} - x_{F(i-1)}} \quad (62.37a)$$

$$\theta_{i+1} = \tan^{-1} \frac{y_{F(i+1)} - y_{Fi}}{x_{F(i+1)} - x_{Fi}} \quad (62.37b)$$

Figure 62.21 shows the overall block diagram of the proposed decision-making and path-planning methodology.

62.4.4 Numerical Simulations

62.4.4.1 Simulation Conditions

In order to demonstrate the performance and behavior of the path-planning system, numerical simulations are carried out on a baseline scenario as depicted in Fig. 62.22. The flight plan of the friendly fleet is shown as a bold red (outside

Fig. 62.22 An illustration of basic scenario

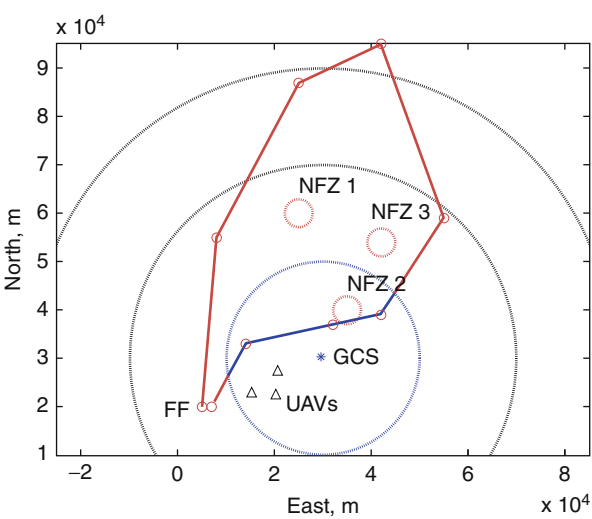


Table 62.2 Simulation parameters

| Variable | Description | Value | Unit |
|------------------|-------------------------------------|---|------|
| R_{com} | Communication range | 20 | km |
| P | Position of GCS | $[30, 30]^T$ | km |
| P_n | Positions of NFZ | $\begin{bmatrix} 25 & 35 & 42 \\ 60 & 40 & 54 \end{bmatrix}$ | km |
| ρ_s | Radius of NFZ | 3 | km |
| W_F | Waypoint sequence of FF | $\begin{bmatrix} 5 & 8 & 25 & 42 & 55 & 42 & 32 & 14 & 7 \\ 20 & 55 & 87 & 95 & 59 & 39 & 37 & 33 & 20 \end{bmatrix}$ | km |
| V_F | Speed of FF | 50 | m/s |
| d_s | Separation distance | 2.5 | km |
| V_{max} | Maximum speed of UAV | 50 | m/s |
| n_{max} | Maximum lateral acceleration of UAV | 1 | g |
| k_o | Overlap coefficient | 0.9 | N/A |

the GCS communication range) and blue (inside the GCS communication range) line passing through a series of predefined waypoints. The circumstances are such that the flight of the FF must pass outside of the GCS communication range for a significant period. Figure 62.22 shows that the FF flies up to three times the communication ranges. Three nonflying zones (red dotted circles) are also defined around and outside the GCS communication range. The speed and the flight plan of the FF for the whole mission is known to the UAV members in advance ahead the next waypoint. The UAV group consists of three members (defined by (Eq. 62.16)) whose initial positions are illustrated as triangles. The detailed default parameters can be found in Table 62.2.

Fig. 62.23 Simulation result on basic scenario with two UAVs: position histories of UAVs (x) and friendly fleet (o) (star: GCS, red dotted circle: NFZ, black dotted circle: GCS coverage)

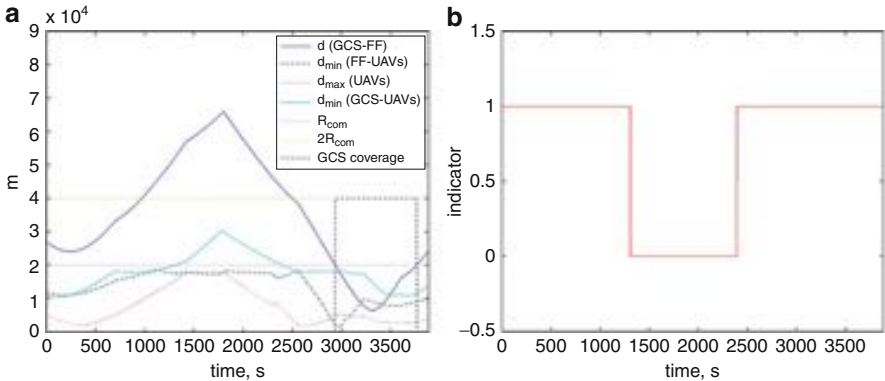
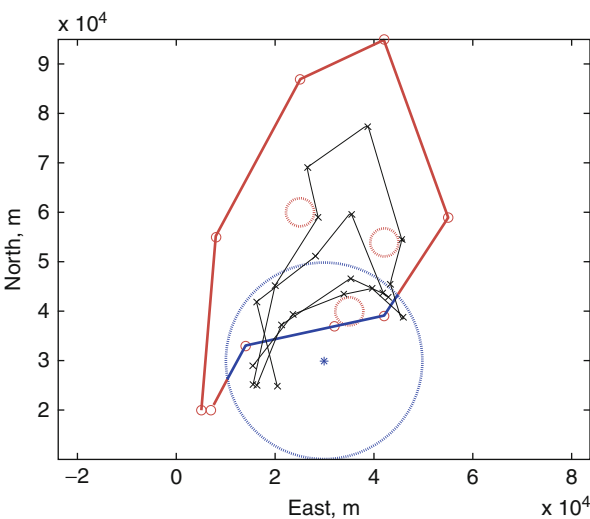


Fig. 62.24 Simulation result on basic scenario with two UAVs: (a) relative distance histories among GCS, UAVs, and friendly fleet and (b) indication on communication relay (1: success, 0: failure)

62.4.4.2 Simulation Results

To effectively compare performance, the proposed algorithm is applied to a two-UAV swarm. Figures 62.23 and 62.24 show that the movements of the UAV members fail to maintain the communication relay for about 1,200 s in the middle of the mission because the distance from the GCS to the UAV members exceeds the communication range R_{com} , which means the GCS loses the communication with the UAV. This implicates that more than two UAVs should be required to maintain the communication relay between the FF and the GCS.

Fig. 62.25 Simulation result on basic scenario with speed and NFZ constraints: position histories of UAVs (x) and friendly fleet (o) (star: GCS, red dotted circle: NFZ, black dotted circle: GCS coverage)

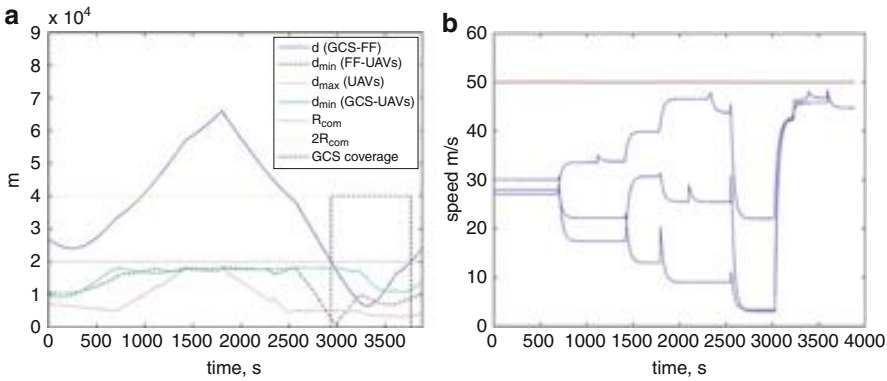
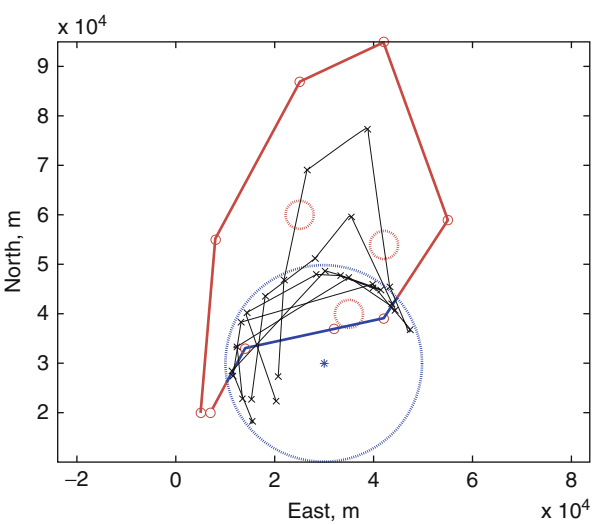


Fig. 62.26 Simulation result on basic scenario with speed and NFZ constraints: (a) relative distance histories among GCS, UAVs, and friendly fleet and (b) speed histories of UAVs (red line: V_{max})

The trajectories of the UAVs avoid entering the NFZ effectively, and at the same time, their speeds do not go over the maximum, as shown in Figs. 62.25 and 62.26. Also, this figure shows that the communication is successfully maintained. It can be shown that the minimum distance from the GCS to the UAVs, the maximum distance between the UAVs, and the minimum distance from the FF to the UAVs are always kept below the communication range R_{com} . Figure 62.27 shows snapshots of the communication boundaries of the UAV members, depicted as the red circles, where the communication between the GCS and the friendly fleet is successfully relayed by restructuring the UAV positions in a chain shape.

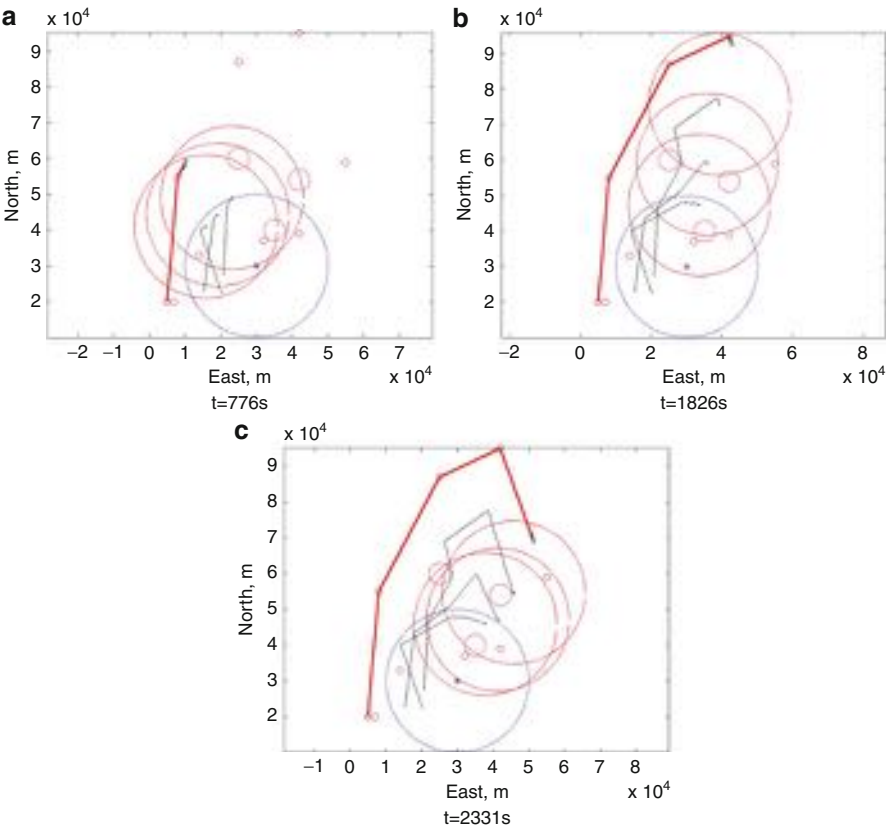


Fig. 62.27 Simulation result on basic scenario with speed and NFZ constraints: snapshots of communication ranges of UAVs. (a) $t = 776$ s. (b) $t = 1,826$ s. (c) $t = 2,331$ s

Conclusions

This chapter has developed a framework enabling cooperative mission and path planning of multiple UAVs in the context of the vehicle-routing problem. The vehicle-routing algorithms often approximate paths of UAVs to straight lines to reduce computational load. In order to resolve this issue, this study focused on integration of the differential geometry concepts, especially Dubins paths, into the vehicle route-planning design procedure to consider non-straight path components. Note that physical and operational constraints of the UAV can be taken into account by these non-straight components of paths. In order to validate this approach, cooperative mission and path-planning algorithms for two missions are developed: road-network search and communication relay. In the road-network search algorithm, the conventional Chinese Postman Problem (CPP) algorithm was firstly explained and modified and applied to the general type of road map which includes unconnected roads. Then, MILP optimization

and the nearest insertion algorithm along with auction negotiation were examined for multiple unmanned aerial vehicles. To realistically accommodate the maneuvering constraints of UAVs, the Dubins path planning was used to solve the modified CPP. The performance of the MILP optimization and the approximation algorithm was then compared for a specific road-network scenario. Particularly, the performance of the approximation algorithm was investigated via a Monte Carlo simulation framework by analyzing the effects of different map sizes, path-planning methods, and the number of UAVs. Based on these results, the efficient UAV team size and path-planning method were suggested for the road search route planning and hence showed that this framework can be applied to a variety of autonomous search missions in the initial phase of mission planning. In the communication-relay problem, a design methodology on vehicle route planning of multiple UAVs to maintain communication between the ground control station and the friendly fleet was proposed. Dubins theory was applied to get flyable paths, and a decision-making algorithm was developed to satisfy the constraints on the UAV speed and the avoidance of nonflying zones, as well as maintaining communication. The performance of the proposed algorithms is examined via numerical simulations.

References

- A. Ahmadzadeh, G. Buchman, P. Cheng, A. Jadbabaie, J. Keller, V. Kumar, G. Pappas, Cooperative control of UAVs for search and coverage, in *Conference on Unmanned Systems* (SPIE, Bellingham, 2006)
- D. Ahr, Contributions to multiple postmen problems. Ph.D. thesis, Heidelberg University, 2004
- B. Alspach, Searching and sweeping graphs: a brief survey. *Matematiche (Catania)* **59**, 5–37 (2006)
- T. Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures. *Int. J. Manag. Sci.* **34**(3), 209–219 (2006)
- J. Bellingham, M. Tillerson, A. Richards, J. How, *Multi-Task Allocation and Path Planning for Cooperating UAVs*, *Cooperative Control: Models, Applications and Algorithms* (Kluwer, Dordrecht, 2003)
- C. Cerasoli, N.J. Eatontown, An analysis of unmanned airborne vehicle relay coverage in urban environments, in *Proceedings of MILCOM* (IEEE, Piscataway, 2007)
- P. Chandler, M. Pachter, D. Swaroop, J. Hewlett, S. Rasmussen, C. Schumacher, K. Nygard, Complexity in UAV cooperation control, in *American Control Conference*, Anchorage (American Control Conference, Evanston, 2002)
- L.E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**(3), 497–516 (1957)
- K. Easton, J. Burdick, A coverage algorithm for multi-robot boundary inspection, in *IEEE International Conference on Robotics and Automation* (IEEE, Piscataway, 2005)
- A. Gibbons, *Algorithmic Graph Theory* (Cambridge University Press, Cambridge/New York, 1999)
- J. Gross, J. Yellen, *Handbook of Graph Theory* (CRC, Boca Raton, 2003)
- M. Hifi, M. Michrafy, A. Sbihi, A reactive local search-based algorithm for the multiple-choice multi-dimensional knapsack problem. *Comput. Optim. Appl.* **33**, 271–285 (2006)
- S. Kim, P. Silson, A. Tsourdos, M. Shanmugavel, Dubins path planning of multiple unmanned airborne vehicles for communication relay. *Proc. IMechE G* **225**, 12–25 (2011)

- I. Maza, A. Ollero, Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, in *Distributed Autonomous Robotic Systems 6*, vol. 5 (Springer, Tokyo/New York, 2007), pp. 221–230
- N. Nigam, I. Kroo, Persistent surveillance using multiple unmanned air vehicles, in *Aerospace Conference, 2008 IEEE*, Big Sky (IEEE, Piscataway, 2008)
- H. Oh, S. Kim, A. Tsourdos, B. White, Cooperative road-network search planning of multiple UAVs using dubins paths, in *AIAA Guidance, Navigation and Control Conference*, Portland (AIAA, Reston, 2011a)
- H. Oh, H. Shin, A. Tsourdos, B. White, P. Silson, Coordinated road network search for multiple UAVs using dubins path, in *1st CEAS Specialist Conference on Guidance, Navigation and Control*, Munich (Springer, Berlin/Heidelberg, 2011b)
- H. Oh, S. Kim, A. Tsourdos, B. White, Coordinated road-network search route planning by a team of UAVs. *Int. J. Syst. Sci.*, (2012) In press
- T. Parsons, Pursuit-evasion in a graph, in *Theory and Applications of Graphs* (Springer, Berlin, 1976)
- N. Perrier, A. Langevin, J. Campbell, A survey of models and algorithms for winter road maintenance. Part IV: Vehicle Routing and Fleet Sizing for Plowing and Snow Disposal. *Comput. Oper. Res.* **34**, 258–294 (2007)
- M.F.J. Pinkney, D. Hampel, S. DiPierro, Unmanned aerial vehicle (uav) communications relay, in *IEEE Military Communications Conference, 1996. MILCOM'96, Conference Proceedings* (IEEE, Piscataway, 1996)
- D. Rosenkrantz, R. Stearns, P. Lewis, An analysis of several heuristics for the traveling salesman problem. *Fundam. Probl. Comput.* **1**, 45–69 (2009)
- J. Royset, H. Sato, Route optimization for multiple searchers. *Nav. Res. Logist.* **57**, 701–717 (2010)
- J. Ryan, T. Bailey, J. Moore, W. Carlton, Reactive tabu search in unmanned aerial reconnaissance simulations. in *30th Conference on Winter Simulation*, Washington, DC, 1998
- T. Ralphs, L. Ladanyi, M. Guzelsoy, A. Mahajan, SYMPHONY 5.2.4 (2010), <http://projects.coin-or.org/SYMPHONY>
- T. Samad, J. Bay, D. Godbole, Network-centric systems for military operations in urban terrain: the role of UAVs. *Proc. IEEE* **95**(1), 92–107 (2007)
- M. Shanmugavel, Path planning of multiple autonomous vehicles. Ph.D. thesis, Cranfield University, 2007
- M. Shanmugavel, A. Tsourdos, B.A. White, R. Zbikowski, Differential geometric path planning of multiple UAVs. *J. Dyn. Syst. Meas. Control* **129**, 620–632 (2007)
- A. Tsourdos, B. White, M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles* (Wiley, Chichester, 2010)

Cooperative Task Assignment and Path Planning for Multiple UAVs

63

Sangwoo Moon, David Hyunchul Shim, and Eunmi Oh

Contents

| | | |
|--------|--|------|
| 63.1 | Introduction | 1548 |
| 63.2 | Problem Formulation | 1550 |
| 63.3 | Integrated Hierarchical Structure for Multi-UAV Coordination | 1551 |
| 63.4 | Negotiation-Based Task Assignment | 1552 |
| 63.5 | Intersection-Based Path Planning | 1554 |
| 63.5.1 | Additional Procedures | 1556 |
| 63.5.2 | Overall Procedure | 1560 |
| 63.6 | Potential Field-Based Collision Avoidance | 1561 |
| 63.7 | Simulation and Analysis | 1564 |
| 63.7.1 | Intersection-Based Path Planning | 1564 |
| 63.7.2 | Negotiation-Based Task Assignment | 1566 |
| 63.8 | Flight Experiments and Validations | 1568 |
| | References | 1576 |

Abstract

In this chapter, a hierarchical framework for path planning and task assignment for multiple unmanned aerial vehicles in a dynamic environment is presented. For multi-agent scenarios in dynamic environments, a candidate algorithm should be

S. Moon (✉)

Department of Aerospace and Mechanical Engineering, Korea Air Force Academy, Cheongju, South Korea

e-mail: dear.sangwoo@gmail.com

D.H. Shim

Department of Aerospace Engineering, Korean Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea

e-mail: hcshim@kaist.ac.kr

E. Oh

CNS/ATM & Satellite Navigation Research Center, Korea Aerospace Research Institute, Daejeon, South Korea

e-mail: emoh@kari.re.kr

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_82,

© Springer Science+Business Media Dordrecht 2015

1547

able to replan for a new path to perform the given tasks without any collision with obstacles or other agents. The path-planning algorithm proposed here is based on the visibility and shortest-path principles in Euclidean space. Instead of typical visibility graph-based methods that scan through all nodes, A^* algorithm is adopted to find an admissible path in a “best-first” approach during the search process. Since the direct outcome from such algorithms may not produce admissible paths in complex environments due to the problems including *cul-de-sac*, additional procedures are conceived to find a solution with a lower cost by avoiding local minima and eliminating any redundant nodes. The path planner is augmented with a potential field-based trajectory planner, which solves for a detouring trajectory around other agents or pop-up obstacles. Task assignment is achieved by a negotiation-based algorithm, which assigns a task with the lowest cost to each agent after comparing all task costs of all participating agents. These algorithms are implemented on MATLAB/Simulink, which can run with simulated vehicle models or actual UAVs through a communication network. In the simulations, the algorithms were validated to perform task assignment and path planning flawlessly. In actual flight tests, the proposed algorithms were tested with a number of fixed-wing UAVs in a fully realistic situation under various reality factors such as communication loss or tracking errors. The flight test shows, even in the presence of such uncertainties and logistic factors, the algorithms were able to perform all of the given tasks without any collision with other agents or obstacles.

63.1 Introduction

Unmanned aerial vehicles (UAVs) have been found highly effective in many applications where human presence in the aircraft is deemed unnecessary or dangerous. Furthermore, as the mission becomes more complicated, it has been suggested to perform the mission more effectively by “sharing the burden” among multiple UAVs. In such scenarios, UAVs are required to fly through the designated waypoints associated with tasks in the given mission. When tasks are performed by multiple agents, the question about “who is doing which task in what order” naturally arises. Also, when multiple UAVs fly together in a given area visiting waypoints, planning a safe and efficient path for each agent is very important. Finally, all agents should be able to avoid obstacles such as terrain or buildings by replanning the path dynamically.

When planning for a mission of a UAV, the flight paths for all UAVs in the scenario are determined by assigning all tasks to participating agents. When the order of the tasks to be performed by each agent is decided, the path for each agent can be easily determined by simply connecting the waypoints associated with the tasks sequentially. However, when the mission area are filled with obstacles, the original path should be modified to pass through the designated waypoints while staying clear from the forbidden zones, which may be known a priori or suddenly “pop-up.” Also, when the UAVs are required to fly close to the ground, they need

to avoid collisions with ground objects or buildings, which are already mapped or detected by onboard sensors such as laser scanners or radars.

Assigning tasks to UAVs is not a straightforward problem especially when the UAVs fly in a dynamic environment where obstacles are not fully mapped a priori or when the sensors for obstacle detection have errors. Mixed Integer Linear Programming (MILP) has been found as a powerful method because it can handle dynamic systems with discrete decision variables and there exist many efficient solvers. Bellingham used MILP for task assignment to handle waypoint visiting problems (Bellingham et al. 2001). However, since the complexity of the problem rapidly grows as the number of variables increases, it is often not suitable for real-time applications. Furthermore, since the MILP is formulated with object functions for the entire system, it needs to be implemented as a centralized system, which may not function well when the central control station or communication links are broken (Bertsekas 1991; Ren et al. 2007; Choi et al. 2009; Sujit et al. 2007; Viguria et al. 2010).

Path planning is another important problem for efficient and collision-free operations for multiple UAVs. A path planner should satisfy completeness, optimality, computational tractability, and scalability. In addition, collision and threat avoidance is required for multi-agent scenarios and there are many algorithms for this problem. Many researches have been conducted on the path-planning problem since 1960s (Latombe 1991; Lavalley 2006; Hart et al. 1968; Redding et al. 2007; Yang et al. 2010). However, the path-planning problem becomes more complicated when solving for dynamic environments with incompletely known obstacles or pop-up threats. For such cases, the planner should be able to find a conflict-free path in real time. Schouwenaars et al. used mixed integer programming for multi-vehicle path planning (Schouwenaars et al. 2001). Richards showed an aircraft trajectory planning with collision avoidance using MILP (Richards et al. 2003).

Also, some variations of MILP have been applied to this problem. Earl and D'Andrea developed an iterative MILP approach (Earl and D'Andrea 2005), and Jain and Grossmann proposed hybrid MILP/CP (constrained programming) models for a class of the optimization problems (Jain and Grossmann 2001). Chang and Shadden introduced the gyroscopic forces for collision avoidance, which is similar to the potential field approach (Chang et al. 2003). More recently, Shim et al. pioneered the dynamic path-planning problem using model predictive control (MPC) algorithm for a collision-avoidance problem of 16 homogeneous rotary UAVs with obstacles (Shim 2008).

In this chapter, a hierarchical framework for efficient path planning and task assignment for multiple UAVs in dynamic environments is presented. For task assignment, a negotiation-based algorithm is proposed. In this approach, each agent initially chooses the task with the lowest cost, which is assumed directly proportional to the distance to the waypoint associated with a task. The choices of all other agents are notified to each agent, and when there are conflicts, i.e., when two or more agents choose the same task, their costs are compared, and the agent with the lowest cost are assigned with the task, while the remaining agents repeat the same negotiation process until all agents are assigned with a task without any

conflicts. The proposed algorithms are computationally light enough to handle real-time path planning in partially known environments.

The task-assignment problem is essentially tightly coupled with the path planning. A path-planning algorithm based on the shortest-path principle in Euclidean space is chosen in this framework, where obstacles are initially modeled as polygons and a path between the two waypoints around obstacles is found using the visibility and the shortest-path principles in Euclidean space. In order to search for an admissible path without scanning over the entire solution set, A^* algorithm is deployed for an effective search. When the solution is found by running the A^* algorithm (and not by an exhaustive search), the solution of the planner may converge to a less optimal or even inadmissible path. Therefore, additional filters such as a *cul-de-sac* filter, a recursive filter, and a Fibonacci filter are introduced. These filters help to alleviate issues due to local minima while eliminating unnecessary segments in a candidate path. In addition, since the path solution found prior to the mission may have conflicts with pop-up obstacles or other agents, it is locally adjusted with a potential field-based collision-avoidance algorithm, which is responsible for solving a detouring trajectory around the obstacles intersecting with the original path solution.

The proposed algorithms are first validated in simulations, where multiple UAVs perform the given tasks while avoiding obstacles and other agents. The proposed algorithm implemented on MATLAB/Simulink is shown to be capable of generating admissible paths in completely/partially known or completely unknown environments. Then the algorithm is validated in a flight test using three fixed-wing UAVs to validate the task-assignment and path-planning algorithms together in an environment with pop-up threats. The proposed algorithms were shown to be capable of real-time path planning and task assignment in outdoor environments even with simulated dynamic obstacles.

63.2 Problem Formulation

In the following, the problem for task assignment for multiple UAVs in dynamic environments is defined. Hereinafter, tasks are defined as a set of actions to accomplish a job, a problem, or an assignment. It can be represented as a number of lower-level goals that are necessary for achieving the overall goal of the system (Pettersson and Doherty 2006; Maza et al. 2011). In this chapter, tasks is defined as visiting a waypoint, through which at least one agent should pass once to complete the associated task. First of all, the rules for the task assignment and the path planning are established as below (Moon and Shim 2010):

- (a) Obstacles are defined as the objects that obstruct the motion. In this research, they are represented as polygons.
- (b) The environment is modeled as a two-dimensional Euclidean space.
- (c) UAVs must avoid all obstacles. In other words, all UAVs must remain in admissible areas, clear from obstacles with a margin. Therefore, paths of all UAVs solved by the proposed procedure should be admissible. Path can intersect, but should not lead to collision among UAVs.

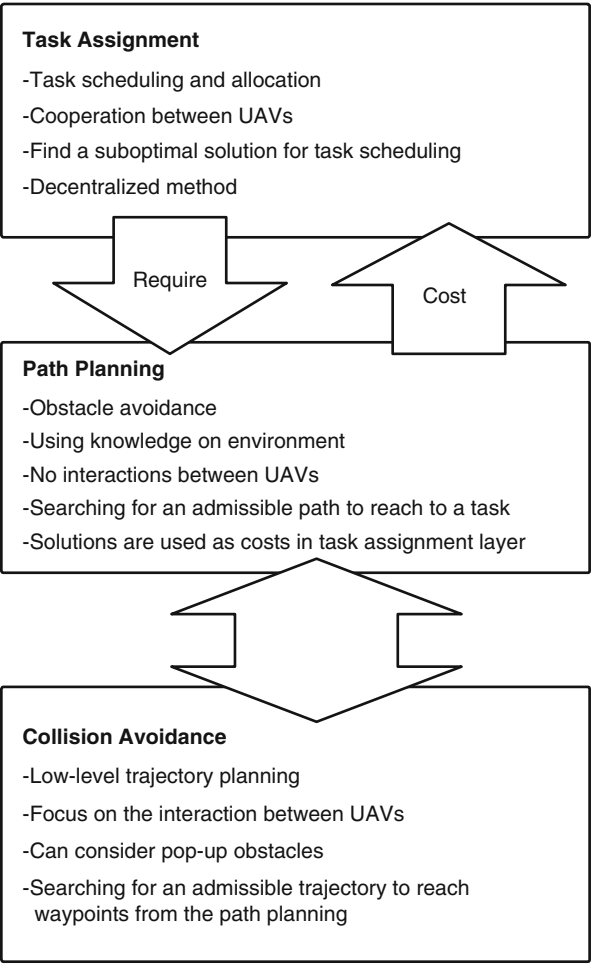


Fig. 63.1 Integration of task assignment, path planning, and collision avoidance

(e) The task assignment and the path planning should run in real time.

In the following, the task-assignment and the path-planning algorithms based on rules listed above are presented.

63.3 Integrated Hierarchical Structure for Multi-UAV Coordination

Figure 63.1 explains how the task-assignment layer, the path-planner layer, and the collision-avoidance layer are integrated altogether. The integration of the task assignment and the path planner is a crucial step for architecting a mission-planning system for multi-agent scenarios (Moon et al. 2013). In this research, the task-

assignment layer needs to communicate with the path planner to compute the cost for task assignment, which is a function of the distance to reach the waypoint. During the task assignment, the path-planning layer is repeatedly queried until a feasible solution for task assignment is found.

The next step is, if any unmapped obstacle is detected while flying along the original path, to find a collision-free path based on the original path. It is noted that the path-planner and the collision-avoidance layer may appear to do a similar task of finding a conflict-free path. The crucial difference is that the collision-avoidance layer finds a path locally detouring from the original path computed by the path planner, which only uses the obstacle information known before starting the mission. Therefore, these two algorithms work in a complementary manner to generate a collision-free path efficiently.

As can be seen in Fig. 63.1, these two procedures are in a hierarchical structure, and the path-planning algorithm is invoked when the situation is in one of the following three cases: (i) before carrying out the mission, (ii) a new task is allocated on an agent participating in the mission, or (iii) new obstacles or threats suddenly appear, or “pop-up.” The collision-avoidance algorithm is activated when the agent flies to the target point while following the path. For multi-UAV scenarios, the collision avoidance between two or more agents should be considered. The avoidance algorithm presented later in this chapter is used to generate a collision-free path.

63.4 Negotiation-Based Task Assignment

During the last decade, many algorithms have been developed for task-assignment problems (Bertsekas 1991; Ren et al. 2007; Choi et al. 2009; Viguria et al. 2010). The task-assignment algorithm presented in this chapter is based on the negotiation algorithm (Moon et al. 2013). The proposed method should generate a feasible solution within a reasonable time for real-time applications. In addition, this algorithm is an *event trigger-based process*, which means that it runs only when an agent sends a message to invoke this algorithm. This is a desirable attribute for decentralized systems running in real time with limited communication bandwidth.

Events can be classified into three cases: (i) a mission is given to the system and the agents start to achieve the goal of the mission, (ii) a task in the given mission is accomplished by an agent, and (iii) the given mission is completely finished. If any of these events occurs during the mission, the presented task-assignment process is activated. Upon activation, all agents can be assigned with the different tasks, and this result is dependent on the conditions of the tasks. On the other hand, the costs for negotiation are defined by using (63.1). This formula consists of two parts: the distance from the current location of an agent to the task location and the characteristics of an agent and the assigned task, i.e., the capability set of an agent and the type of the task. The total cost is a linear combination of these two elements, and there are weights assigned for each term:

| | UAV #1 | UAV #2 | UAV #3 |
|------------------------------------|--------------------------------------|---------------------------------|---------------------------------|
| Task Assignment Requirement | Send the message for task assignment | | |
| Before Negotiation | Calculate the cost individually | Calculate the cost individually | Calculate the cost individually |
| 1st Negotiation | Cost: 40.68 @ Task A | Cost: 40.96 @ Task A | Cost: 42.09 @ Task C |
| | Cost: 40.68 @ Task A | | Cost: 42.09 @ Task C |
| 2nd Negotiation | | Cost: 41.60 @ Task C | Cost: 42.09 @ Task C |
| | Cost: 40.68 @ Task A | Cost: 41.60 @ Task C | Cost: 45.76 @ Task B |

Fig. 63.2 Negotiation-based task allocation for three UAVs with three tasks

$$J_{\text{task}} = w_d J_{\text{dist}} + w_c J_{\text{character}}$$

(63.1)

In (63.1), J_{dist} is the cost related with the distance between the current position of an agent and the location of the given task. If there are multiple tasks with different characteristics, $J_{\text{character}}$ can be given with different costs to affect the behaviors of all agents in this scenario.

The negotiation-based task assignment is performed in the following manner. Initially, each agent chooses its first task, which has the lowest cost from the cost list. Here, without loss of generality, the cost associated with the task is the distance to the task. The choices of all agents are broadcast to other agent, and when there are conflicts, i.e., more than one agents choose the same task, the costs of conflicting agents are compared, and the agent with the lowest cost has the right to claim the task, while other agents without assigned tasks repeat the same compare-and-claim process until all conflicts are resolved.

In Fig. 63.2, an example with three UAVs is given. At first, each UAV chooses their task and the cost is compared. At the first negotiation, UAVs #1 and #2 choose the same task (task A), and they negotiate to find UAV #1 has a lower cost (40.68 vs. 40.96). So UAV #2 chooses a task other than A, which is task C with cost 41.60. This choice now conflicts with that of UAV #3 with cost 42.09, which is higher than the cost of UAV #2. Therefore, UAV #3 is forced to choose task B with cost 45.76.

This approach has no guarantee to converge to the global minimum with an exception that it does converge to the global minimum when the number of the agent

is equal to the number of tasks. However, this algorithm has a low computing loads and suitable for decentralized scenarios with limited communication bandwidth.

63.5 Intersection-Based Path Planning

In Euclidean space, the shortest path between two points is the straight line between them. Denoting these points as \mathbf{x}_s and \mathbf{x}_g , the minimum cost function, i.e., the distance of the shortest path, can be represented as (63.2):

$$\min J = \text{dist}(\mathbf{x}_s, \mathbf{x}_g) = \|\mathbf{x}_s - \mathbf{x}_g\| \quad (63.2)$$

where $\text{dist}(\bullet)$ function is the distance defined in the Euclidean space. However, if there are any obstacles or forbidden region on the line, the cost found by (63.2) is not valid since the path is not admissible.

In order to find an admissible path in this situation, the intersection point by the path from (63.2) and the boundary of the obstacle are used (Moon et al. 2013). The first step of the proposed algorithm is to find the intersection point. If an obstacle in the two dimensional space is modeled as a polygon, then the intersection point \mathbf{x}_m in Fig. 63.3 can be found easily. In such a case, a new path that passes the boundary points of a detected obstacle, $\mathbf{x}_{O_{i1}}$ or $\mathbf{x}_{O_{i2}}$ should be found. Therefore, these two points are the candidates of waypoints for path planning. The proposed algorithm uses A^* approach expressed as (63.3):

$$f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}) = \|\mathbf{x}_s - \mathbf{x}_w\| + \|\mathbf{x}_g - \mathbf{x}_w\| \quad (63.3)$$

In (63.3), \mathbf{x}_w is the boundary point of detected edge so that the first term of (63.3) is the distance from the starting point to a waypoint candidate and the other is the distance from the waypoint candidate to the goal point. If the detected line is not the first since the operation started, this equation should be modified to (63.4) because the vehicle is already assigned to a waypoint.

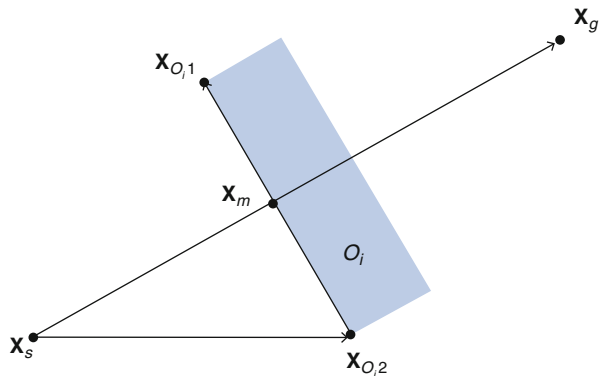


Fig. 63.3 Intersection point between shortest path and obstacle

Table 63.1 Pseudo code for path planning**Algorithm 1** Intersection-based path planning (fundamental)

```

1: P ← InitializePath( $x_s, x_g$ )
2:   $x_i \leftarrow x_s$ 
3:   while Dist( $x_g, x_i$ )  $\geq d_{\text{admissible}}$  do
4:      $x_m \leftarrow \text{DrawLine}(x_g, x_i)$ 
5:     if  $x_m$  exist
6:        $x_i \leftarrow \text{SelectOptimalNode}(x_g, x_s, x_1, x_2, \dots, x_i)$ 
7:       P ← AddPath(P,  $x_i$ )
8:     end if
9:      $i \leftarrow i+1$ 
10:  end while

```

$$f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}) = \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{x}_{(i-1)}\| + \|\mathbf{x}_g - \mathbf{x}_w\| \quad (63.4)$$

In the two-dimensional space, the suboptimal waypoints and the candidate waypoints appear as a pair, and the suboptimal waypoints receive a higher priority for calculation. If multiple obstacles are located in the area, the intersection point which is the closest among the intersection points and the obstacle of that point is considered as the next optimal candidate. As A^* algorithm is used in this procedure, all the nodes found by the intersection method are prioritized, and the best solution is chosen using the priority value. Therefore, those nodes related with the intersection points are applied by this method, and the procedures from (63.2) to (63.4) will be performed until the solved node is the goal point (see Table 63.1).

When A^* algorithm is applied from one root to another, the resulted path is not necessarily optimal. Therefore, a number of additional steps are introduced to refine the path although this process only guarantees to converge to a local minimum. Nonetheless, this process greatly helps to improve the quality of the resulted path, which has much less redundant segments and also avoids getting stuck in a *cul-de-sac*.

Figure 63.4 shows a result obtained using the proposed algorithm. The given environment consists of ten arbitrary-shaped obstacles, which do not have any concave parts that can cause the solver to fall into an infinite loop. Therefore, additional procedures should be added, which will be introduced in the next section. In addition, the configuration space is also considered because UAVs are assumed to occupy certain area (volume), not a point. The dashed line is the solved path, and the lines are the candidate paths to obtain the shortest path. Initially, the algorithm finds a path, which travels to the left side of an obstacle. However, this solution is discarded due to the longer traveling distance around the next obstacle. After several iterations, a new path is found (dashed line), which connects the starting point and the goal point without any conflict with all obstacles.

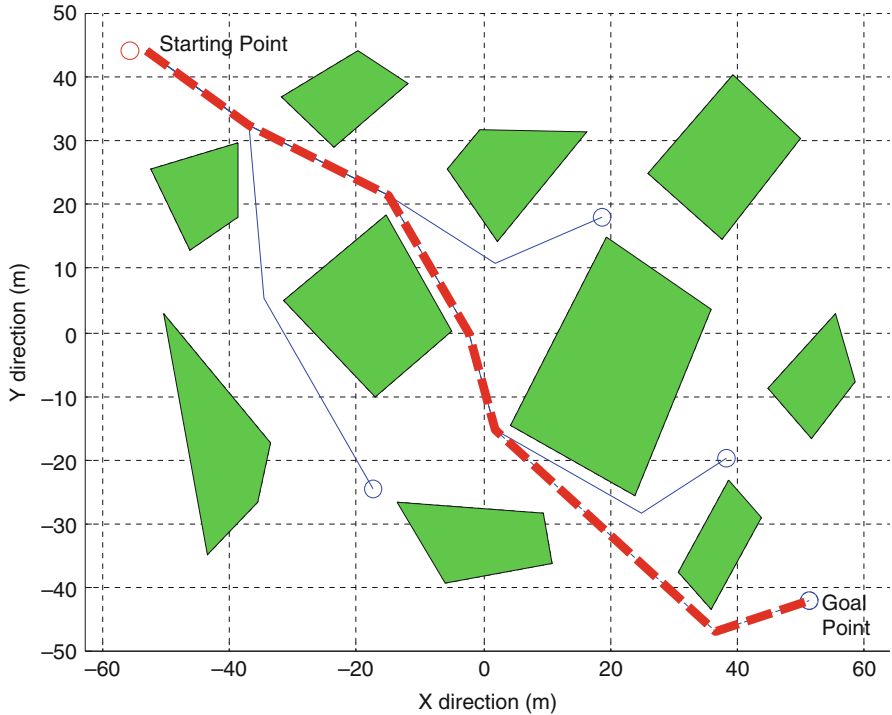


Fig. 63.4 Result from the algorithm proposed in Table 63.1

63.5.1 Additional Procedures

Although the algorithm proposed in Table 63.1 can generate an admissible path in most cases, it is also possible that the planner may get stuck to local minima. Even if the path is admissible, it may contain extra nodes that make the overall path less optimal. Therefore, in this section, three additional procedures are introduced: *cul-de-sac* filtering, Fibonacci filtering, and recursive filtering. These additional procedures further improve the proposed algorithm to find the optimal path more reliably although these procedures render the overall algorithm computationally more demanding.

63.5.1.1 Cul-de-sac Filtering

Cul-de-sac refers to a dead end where there is no outlet other than the inlet. If the proposed algorithm is applied to the situation with concave obstacles that creates *cul-de-sac*, the algorithm may fall into an infinite loop, thus unable to find any admissible path. Therefore, an additional filter is introduced to resolve this problem (see Table 63.2). In Fig. 63.5, a case with three concave obstacles is considered. The direct output from the intersection-based planning finds a path that falls into the *cul-de-sac* point located on the middle of the diagram. Two paths were generated

Table 63.2 Pseudo code for cul-de-sac filtering

| Algorithm 2 Cul-de-sac filtering | |
|---|--|
| 1: if $x_i = x_{i-1}$ | |
| 2: $x_m \leftarrow x_i$ | |
| 3: $x_n \leftarrow x_{i-1}$ | |
| 4: while $x_m = x_n$ do | |
| 5: $x_n \leftarrow \text{SelectNeighborVertex}(x_i)$ | |
| 6: $x_m \leftarrow \text{SelectOptimalNode}(x_g, x_s, x_1, \dots, x_{i-1}, x_n, x_m)$ | |
| 7: end while | |
| 8: end if | |
| 9: $x_i \leftarrow x_m$ | |

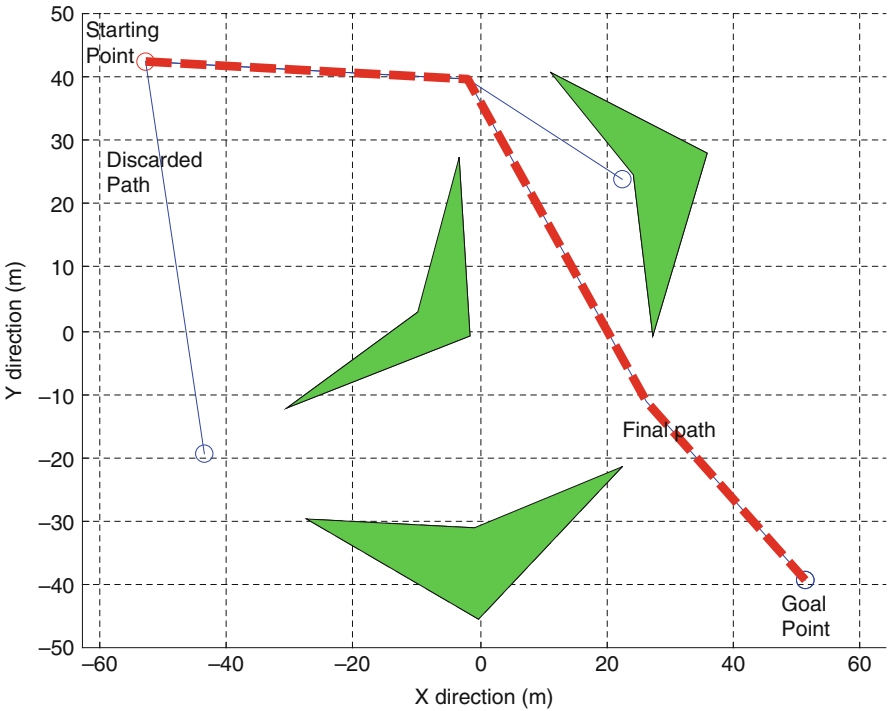


Fig. 63.5 Cul-de-sac filtering

after applying the *cul-de-sac* filtering. When the path on the left is chosen, the other blue path indicated as “Discarded Path” is selected first. However, after running the original algorithm with the *cul-de-sac* filtering, the dashed red line is generated. Figure 63.5 shows the result after applying the *cul-de-sac* filtering algorithm.

63.5.1.2 Recursive Filtering

As the intersection-based path planning considers only one obstacle at a time, if another obstacle is near the one being considered for path planning, the generated

Table 63.3 Pseudo code for recursive filtering**Algorithm 3** Recursive filtering

```

1: if CheckOverlapObstacle( $O_1, O_2, \dots, O_n, x_{i-1}, x_i$ )
2:    $P_s \leftarrow \text{InitializePath}(x_{i-1}, x_i)$ 
3:    $x_j \leftarrow x_{i-1}$ 
4:    $x_m \leftarrow \text{DrawLine}(x_i, x_j)$ 
5:   if  $x_m$  exist
6:      $x_i \leftarrow \text{SelectOptimalNode}(x_i, x_{i-1}, x_j)$ 
7:      $P_s \leftarrow \text{AddPath}(P_s, x_i)$ 
8:   end if
9:    $j \leftarrow j+1$ 
10:   $P \leftarrow \text{MergePath}(P, P_s)$ 
11: end if

```

path may conflict with other obstacles. So the recursive filtering is proposed, which checks if the path around one obstacle passes through any other obstacles. This filtering process checks the path between i -th and the previous node, which can be expressed as

$$\overline{\mathbf{x}_{i-1}\mathbf{x}_i} \cap \bigcup_{j=1}^N O_j \neq \emptyset \quad (63.5)$$

Equation (63.5) implies that the line segment between the previous node \mathbf{x}_{i-1} and the current node \mathbf{x}_i intersects with any obstacles along the line. In such cases, the intersection-based path-planning algorithm is called with \mathbf{x}_{i-1} and \mathbf{x}_i as the input argument for temporary starting and the goal points, respectively, find to a conflict-free path (see Table 63.3). Recursive filtering is performed after running the main planning algorithm and the *cul-de-sac* filtering introduced above. This filtering is frequently required when the given environment is filled with closely located obstacles and consumes a more computation time. Figure 63.6 shows an example of the recursive filtering. Since the line between the starting point and the goal point intersects with the obstacle on the right, the blue path is obtained by the intersection-based algorithm. Since it overlaps with the obstacle on the left, the recursive filtering is applied for the segment between the starting point and the first node, resulting in the red-dashed line, which is conflict-free.

63.5.1.3 Fibonacci Filtering

Fibonacci filtering is named after the well-known Fibonacci sequence from the fact that the current waypoint is computed using the two preceding points. It is developed to eliminate any extra nodes that increase overall cost (i.e., a longer path) when there is an admissible path with a lower cost. In Fibonacci sequence, each number is the sum of the previous two numbers, starting with 0 and 1. Similarly, Fibonacci filtering takes only the previous two nodes as the input argument.

If two nodes \mathbf{x}_{i-2} and \mathbf{x}_i satisfy (63.6), this filtering moves on to the next node. Therefore, Fibonacci filtering is the opposite concept of the recursive filtering described in the previous subsection:

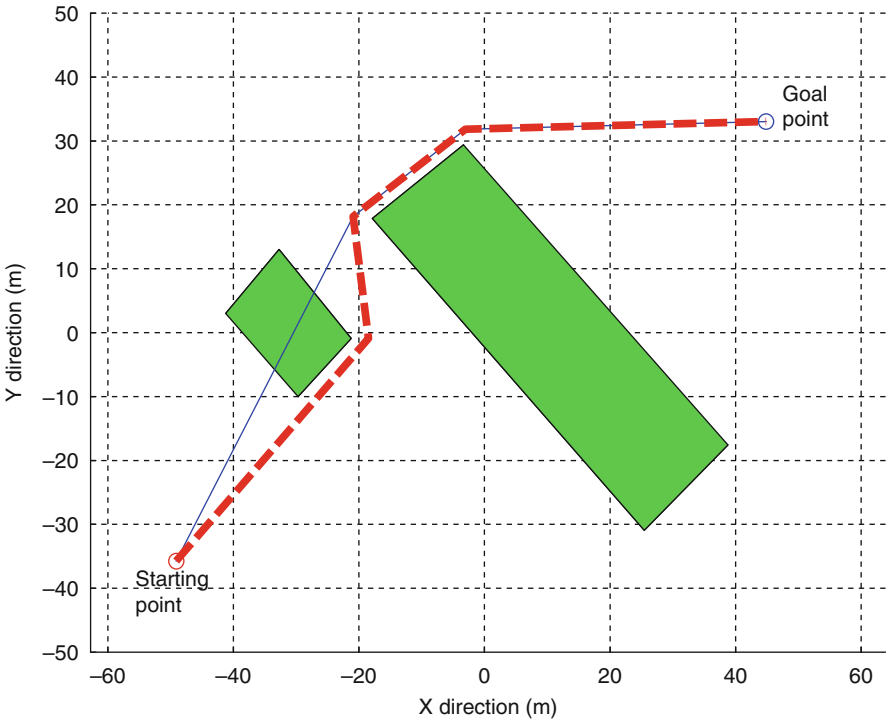


Fig. 63.6 Recursive filtering. *Solid lines (blue):* suboptimal paths before the filtering. *Dashed line (red):* optimal path obtained by the recursive filtering

Table 63.4 Pseudo code for Fibonacci filtering

Algorithm 4 Fibonacci filtering

- 1: if !CheckOverlapObstacle($O_1, O_2, \dots, O_n, x_{i-2}, x_i$)
- 2: $x_{j-1} \leftarrow x_i$
- 3: $i \leftarrow i-1$
- 4: end if

$$\overline{\mathbf{x}_{i-2}\mathbf{x}_i} \cap \bigcup_{j=1}^N O_j = \emptyset \quad (63.6)$$

In (63.6), the line that connects the point before the previous node \mathbf{x}_{i-2} and present node \mathbf{x}_i does not overlap with any obstacles, then the previous point \mathbf{x}_{i-1} should be cancelled out to straighten the generated path. This procedure is performed for all nodes that satisfy (63.6), and repeated until (63.6) is not satisfied for all nodes (see Table 63.4). As is the case with the additional filtering algorithms introduced above, this filtering is more frequently needed if the given environment is complex and more computation time is needed for this algorithm along with other additional filtering algorithms mentioned above. In Fig. 63.7, the first two segments

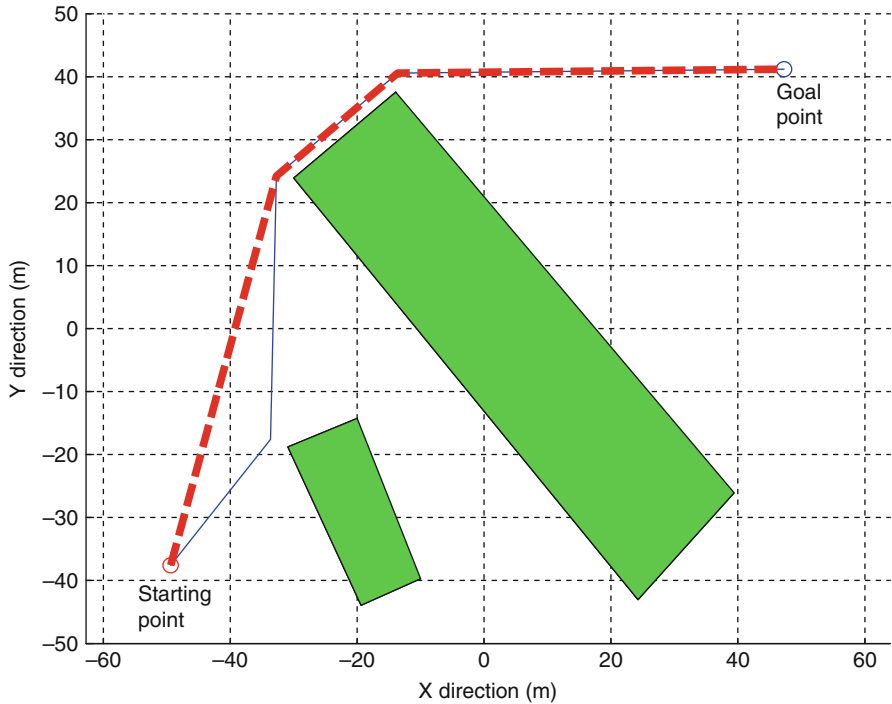


Fig. 63.7 Fibonacci filtering. *Solid line (blue):* suboptimal paths before the filtering. *Dashed line (red):* optimal path obtained by the Fibonacci filtering

from the starting point on the blue line are less optimal than the first line segment of the red-dashed line due to the extra node, which is eliminated by running the Fibonacci filtering.

63.5.2 Overall Procedure

Table 63.5 is the pseudo code for the all algorithms proposed so far. When the path planning is started, the intersection-based method is first applied and an initial candidate path is found. At this step, the cost function of the candidate paths are computed to run the *A** algorithm to find the path with the least cost. Here, the path may get stuck in *cul-de-sac*, which is avoided by the proposed filtering (recursive filtering). In this process, the path may have extra node, which is then eliminated by the Fibonacci filtering. The filtering procedures are performed after the intersection-based algorithm. When the path reaches to the goal point, this algorithm is terminated.

Figure 63.8 shows the path planning result with the recursive filtering and Fibonacci filtering in addition to the intersection-based method. Here, the recursive filtering process was responsible for finding the optimal path passing by obstacles 1

Table 63.5 Pseudo code for iterated algorithm

Algorithm 5 Intersection-based path planning (overall)

```

1:  $P \leftarrow \text{InitializePath}(x_s, x_g)$ 
2:  $x_i \leftarrow x_s$ 
3: while  $\text{Dist}(x_g, x_i) > d_{\text{admissible}}$  do
4:    $x_m \leftarrow \text{DrawLine}(x_g, x_i)$ 
5:   if  $x_m$  exist
6:      $x_i \leftarrow \text{SelectOptimalNode}(x_g, x_s, x_1, x_2, \dots, x_i)$ 
7:     if  $x_i = x_{i-1}$ 
8:        $x_i \leftarrow \text{CulDeSacFiltering}(x_g, x_s, x_1, x_2, \dots, x_i)$ 
9:     end if
10:    if  $\text{CheckOverlapObstacle}(O_1, \dots, O_n, x_{i-1}, x_i)$ 
11:       $P, x_i \leftarrow \text{RecursiveFiltering}(x_g, x_s, x_1, \dots, x_i)$ 
12:    end if
13:    if  $\neg \text{CheckOverlapObstacle}(O_1, \dots, O_n, x_{i-2}, x_i)$ 
14:       $x_{j-1} \leftarrow x_i$ 
15:       $i \leftarrow i-1$ 
16:    end if
17:     $P \leftarrow \text{AddPath}(P, x_i)$ 
18:  end if
19:   $i \leftarrow i+1$ 
20: end while

```

and 2. The Fibonacci filtering is responsible for finding the better (i.e., shorter) path among the three paths ending up near obstacle #3.

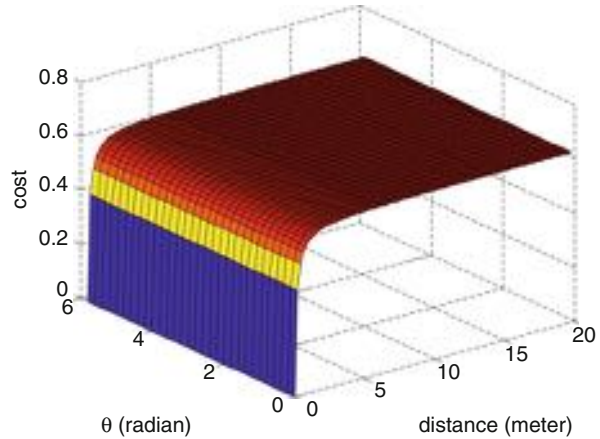
63.6 Potential Field-Based Collision Avoidance

Collision avoidance is a dynamic path generation technique, which finds an admissible path to avoid a collision among agents or with obstacles in the environment. The algorithm proposed here is based on the potential field approach with several improvements presented below (Moon et al. 2013). Most algorithms based on the potential field are based on the distance between the vehicle and target points at one time frame. However, in case of moving obstacles, it is desirable to consider the relative direction of motion as well. The proposed algorithm utilizes the cost function for the potential field as the function of the distance and direction of the obstacle using the normal vector as

$$J = fn(\mathbf{x}, \mathbf{v}, \mathbf{n}) \quad (63.7)$$

where \mathbf{x} and \mathbf{v} represent the position and velocity of the vehicle, respectively. \mathbf{n} is the set of normal vectors of the planes on the polyhedra representing the obstacles in the environment. Figure 63.9 illustrates the relationship among a UAV, an obstacle, and a waypoint along with the current and future position and velocity vectors of a UAV and a normal vector \mathbf{n}_i .

Fig. 63.10 Cost function for waypoint tracking



UAV moves in one step along the selected path. This procedure is iterated until the UAV reaches to the target point.

The cost function for the waypoints generated from the path planning can be expressed as (63.8), which can be visualized as shown in Fig. 63.10. Note that (63.8) is not dependent on the vehicle's heading angle. In order to prevent (63.8) from dominating other terms, a denominator that divides by the distance from the waypoint to the current position of an UAV is included:

$$J_{\text{waypoint}} = 1 - \exp\left(\frac{\|\mathbf{x}_{\text{waypoint}} - \mathbf{x}_{t+\Delta t}\|}{\|\mathbf{x}_{\text{waypoint}} - \mathbf{x}_t\|}\right) \quad (63.8)$$

The cost function for the obstacles and threats is expressed as (63.9), which is visualized as in Fig. 63.11, and again it is independent from the vehicle's heading. If the angle between the normal vector of the plane on an obstacle and the vehicle's velocity vector is very small, this indicates that the vehicle approaches to the object and therefore the cost function grows exponentially. However, if this angle is close to 180° , the cost function decreases rapidly because the UAV moves away from the object. The cost function in Eq. (63.9) also includes a denominator to scale the magnitude of the normal vector is included to divide the term by a magnitude vector from the object to the current position of an UAV:

$$J_{\text{obstacle},i} = (0.5 \cos \theta_{\text{obstacle}} + 0.5 + C_{\text{obstacle}}) \exp\left(-\frac{\|\mathbf{n}_i\|}{\|\mathbf{x}_t - \mathbf{x}_{\text{obstacle}}\|}\right) \quad (63.9)$$

where

$$\theta_{\text{obstacle}} = \cos^{-1}\left(\frac{\mathbf{n}_i \cdot \mathbf{v}_t + \Delta t}{\|\mathbf{n}_i\| \|\mathbf{v}_t + \Delta t\|}\right) - \pi$$

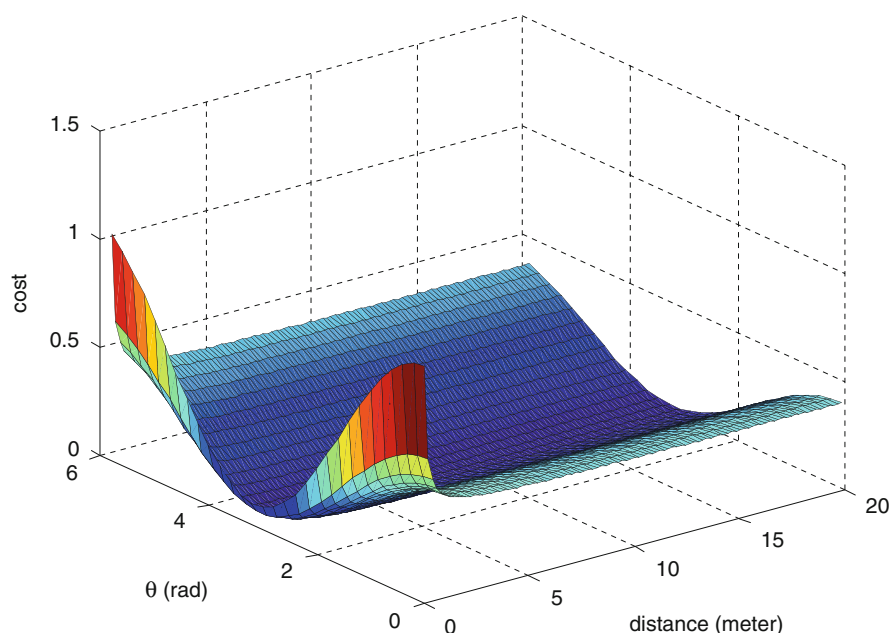


Fig. 63.11 Cost function for obstacles and threats

63.7 Simulation and Analysis

In order to evaluate the performance of the proposed algorithm, a series of simulations are conducted. In the proposed hierarchy, since the task-assignment layer cannot function without path planning, the intersection-based path planner is first implemented and validated.

63.7.1 Intersection-Based Path Planning

It is assumed that the UAV has an onboard sensor for detecting obstacles such as the laser scanner, which has finite detection range. In the simulation, 20 randomly shaped obstacles are placed where only ten of them are known a priori. The simulation was constructed in MATLAB running on a PC with Intel CPU Centrino 2 Duo 2.4 GHz.

Figure 63.12 shows the path-planning result when all of the obstacles are known a priori. UAV 1 flies to the goal point without any conflict with obstacles. Whenever obstacles unknown at the initial path-planning phase are discovered during the flight, the path-planning algorithm runs repeatedly in real time to find a conflict-free path (left side of Fig. 63.13). If none of the obstacles were known a priori, the path-planning algorithm runs in real time from the start until the end, building the obstacle

Fig. 63.12 Path planning in totally known environment

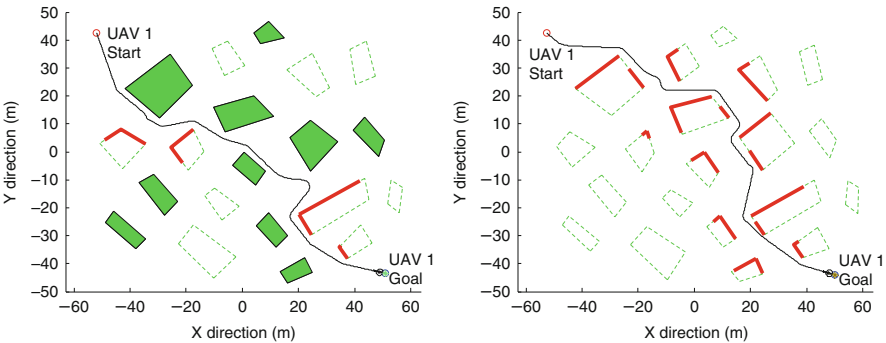
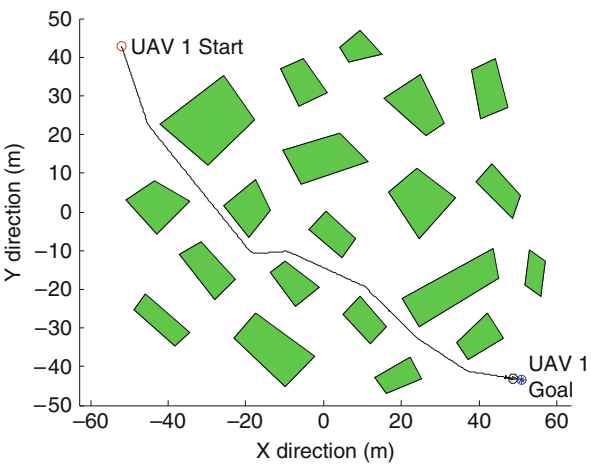


Fig. 63.13 Path planning in partially known environment (*left*) and totally unknown environment (*right*)

map *on the fly* using onboard sensors (right side of Fig. 63.13). Note that the obtained paths in the three cases with identical obstacles but with different sensing methods are different from one another.

It is generally agreed that finding a feasible path in a given environment is a hard problem and finding the optimal solution, whichever the definition of optimality is, is a difficult and there is not a general solution for such work. The proposed algorithm is viable in the sense that it can find an admissible path with a smaller computation load with all obstacles modeled as polygons. Such simplified obstacle is acceptable with path planning for UAVs, which needs to stay clear from obstacles for safety and to avoid turbulence caused by the airflow bouncing back from the obstacles. The proposed algorithm runs very fast in environments with fewer obstacles. However, if the environment is filled with many obstacles, the computation load can be quite heavy because of the additional filtering processes.

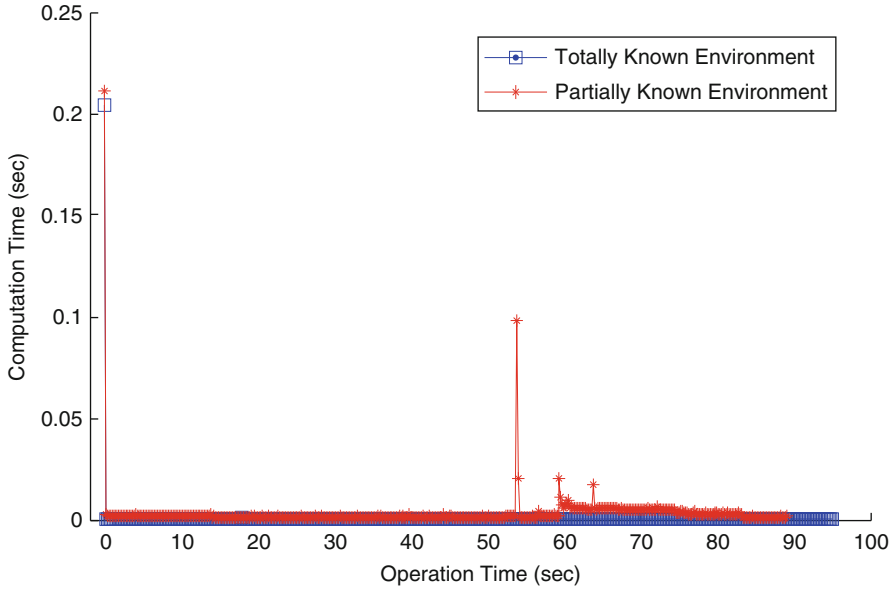


Fig. 63.14 Computation time versus elapsed time

Figure 63.14 shows the computation time during the simulation. The case with totally known environment is the blue lines with square marks, and the case with partially known environment is the red lines with plus marks. The totally known environment case does not require much computation except for the first path planning. However, the partially or totally unknown case requires more computation load during the simulation whenever new obstacles are detected ($t = 53$ s).

63.7.2 Negotiation-Based Task Assignment

In the simulation, three UAVs carry out a mission in an area with ten randomly shaped obstacles. Ten task points are arbitrarily placed in the field. As mentioned above, a task is defined as a visit to its associated waypoint. The margin for obstacle avoidance is assumed 2 m here. In this simulation, a simple kinematic model for UAVs were used such that

$$\begin{aligned} V_x &= V \cos \theta \\ V_y &= V \sin \theta \end{aligned} \quad (63.10)$$

where

$$\dot{\theta} \in [-\delta, \delta] \quad (63.11)$$

To evaluate the performance of the proposed algorithm, a set of simulation is conducted. A scenario with three UAVs carrying out a mission in an area

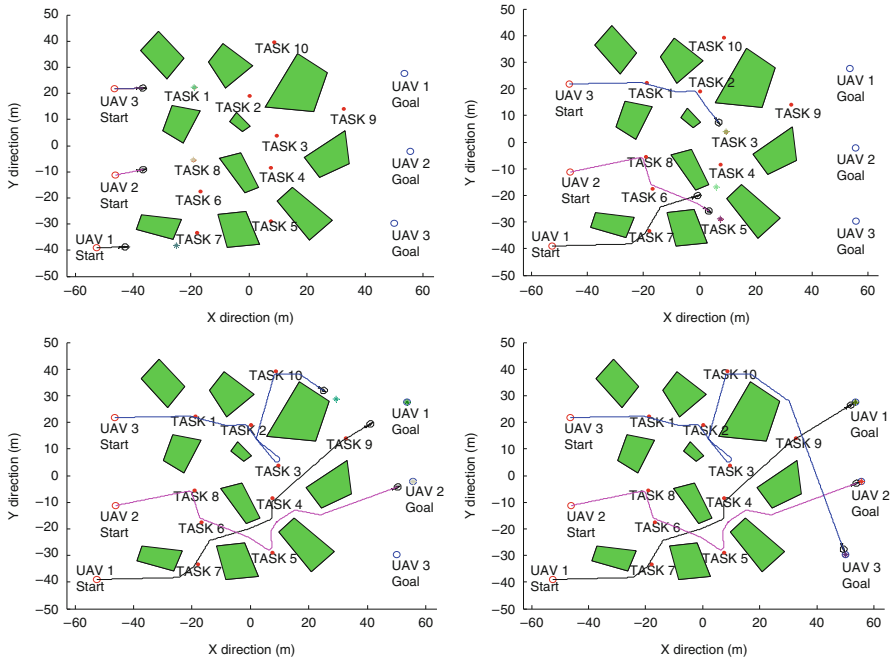


Fig. 63.15 Task assignment in a totally known environment ($t=10,60,120$, and 185 s from *left side of top to right side of bottom*)

with ten randomly shaped obstacles is considered. Ten task points are arbitrarily placed in the field. As mentioned, tasks are defined as a visit to a waypoint. The margin for obstacle avoidance is 2 m. Simple kinematic equations for UAVs were used in this simulation.

Figure 63.15 shows the simulation result. As the mission progresses, ten negotiations occur. UAV 1 is assigned with three tasks (task 4, task 7, and task 9) and participates in the whole negotiation process. UAV #2 is also assigned with three tasks (task 5, task 6, and task 8), and UAV #3 is assigned with four tasks (task 1, task 2, task 3, and task 10) during the mission. At fifth negotiation, UAV #1 sends the information of the determined task to UAV #2 and chooses to undertake task 4 because it is better that UAV #2 carries out task 5. Such a task swapping also occurs at the ninth negotiation process between UAV #1 and UAV #2 (see Table 63.6).

There are nine peaks in the plot of the elapsed time for negotiation process (Fig. 63.16). Although the maximum computation time ($=0.9$ s) of the first process is greater than the iterated time to run the proposed algorithm, it does not affect the real-time performance of the while mission because the task-assignment algorithm is executed offline prior to the mission. Since the computation time to perform each negotiation in the iterative algorithm is very short, the proposed task-assignment algorithm is a viable solution for real-time application. It is also noted that the

Table 63.6
Task-assignment procedure
for simulation with three
UAVs

| | UAV #1 | UAV #2 | UAV #3 |
|---------|---------------|---------------|----------------|
| Step 1 | Task 7 | Task 8 | Task 1 |
| Step 2 | Task 7 | Task 8 | Task 2 |
| Step 3 | Task 7 | Task 6 | Task 2 |
| Step 4 | Task 5 | Task 6 | Task 2 |
| Step 5 | Task 4 | Task 5 | Task 2 |
| Step 6 | Task 4 | Task 5 | Task 3 |
| Step 7 | Task 4 | Task 5 | Task 9 |
| Step 8 | Task 4 | Task 9 | Task 10 |
| Step 9 | Task 9 | Finish | Task 10 |
| Step 10 | Task 9 | Finish | Finish |

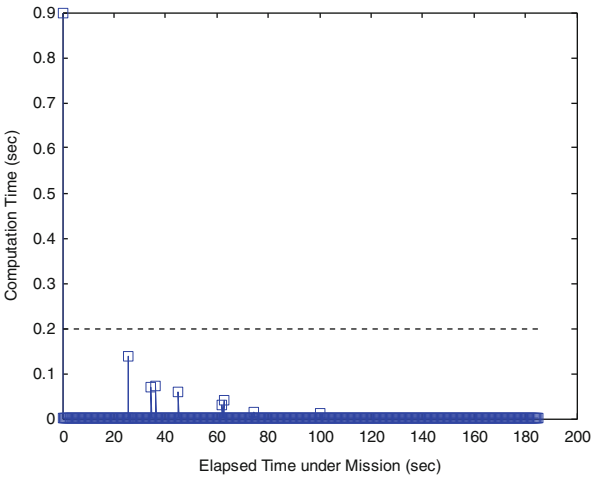


Fig. 63.16 Elapsed time
versus computation time

proposed negotiation-based algorithm exhibits a greedy behavior as each agent thrives to minimize its cost at each time frame in a fully decentralized manner so that the overall task-assignment algorithm generally produces suboptimal results than centralized, non-greedy algorithms.

63.8 Flight Experiments and Validations

As the final step, in order to validate the proposed algorithm in a realistic environment, actual flight tests were conducted using a fixed-wing UAV based on a blended wing body (BWB) airframe as shown in Fig. 63.17. The airplane is powered by an electrical motor and a lightweight in-house flight control computer is installed for automatic waypoint navigation. The test UAV’s specification is given in Table 63.7.

The experiment is conducted in the following steps. When the flight computer is initialized, each vehicle is launched using a bungee cord. After the vehicle climbs to

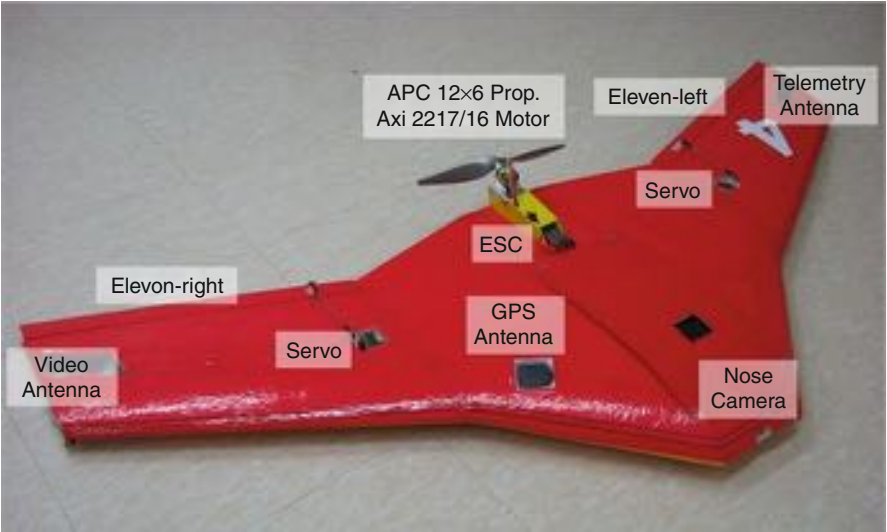


Fig. 63.17 Testbed UAV based on BWB airplane

Table 63.7 Testbed UAV specification

| | |
|----------------|---|
| Base platform | Blended wing body |
| Dimensions | Wing span: 1.52 m(W) Wing area: 0.52 m ² Aspect Ratio: 4.43 |
| Weight | 2.6 kg (fully instrumented) |
| Powerplant | Axi 2217/16 DC brushless motor Lithium-ion-polymer (11.1 V 5,300 mAH) |
| Operation time | <10 min |
| Avionics | Navigation: single GPS-aided INS GPS: U-Blox Supersense 5 IMU: 3DM-GX3 Flight computer: Gumstix Verdex 600 Communication: 900 MHz wireless Ethernet |
| Operation | Catapult launch, body landing |
| Autonomy | Speed, altitude, heading, altitude hold/command Waypoint navigation/automatic landing |

a prescribed altitude, it begins loitering until the experiment session begins. When all agents are airborne and ready to accept the commands for the mission, the ground station starts the task-assignment algorithm, which receives all agents’ position and heading and sends the task commands back to each agent over the wireless network. Each UAV communicates with the ground station through its own onboard modem.

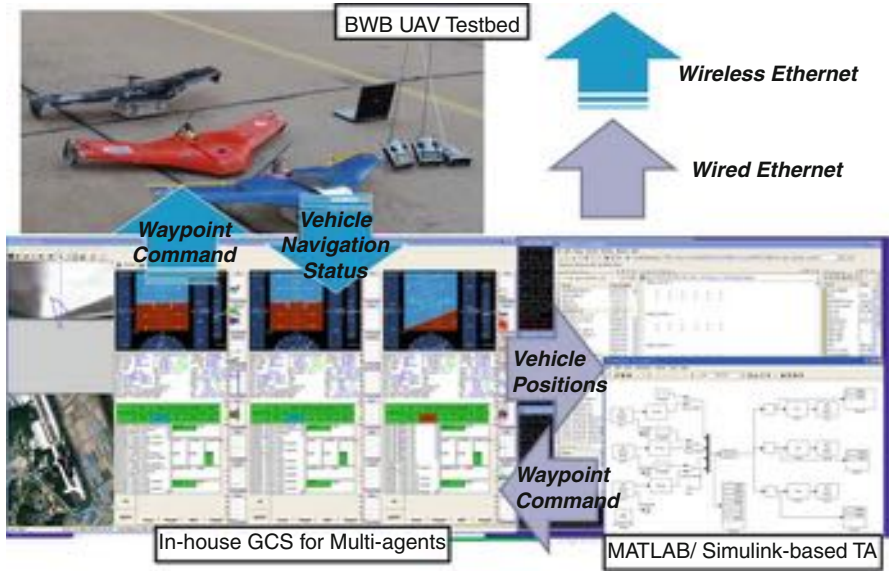
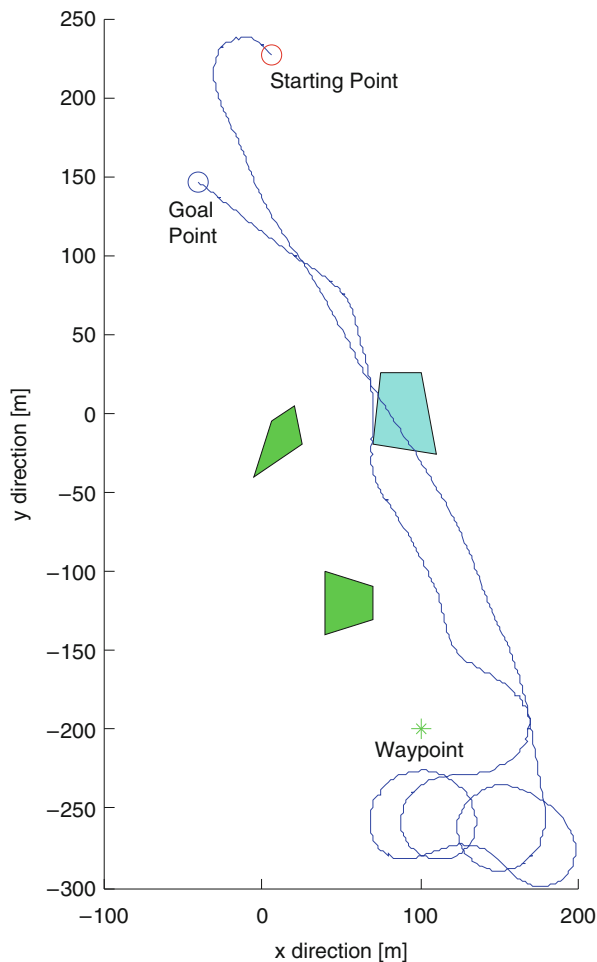


Fig. 63.18 Multi-agent experiment setup using fixed-wing UAVs

When the UAVs receive the task command from the ground, the onboard flight controller steers the vehicle to visit the commanded waypoint. A task is declared completed if the UAV passes the commanded location within a prescribed bound. When the UAV receives a new task request while flying to a task location previously assigned, the flight controller commands the vehicle to fly to the newly received task waypoint, simply ignoring the previous request.

Figure 63.18 illustrates how the entire flight experiment set up is constructed (Moon et al. 2013). The task-assignment and path-planning algorithms are implemented in MATLAB/Simulink, which communicates with the in-house ground station program (called *MVLOG*) capable of handling up to three (as of now) UAVs simultaneously. Each vehicle participating in the scenario sends their current status including the navigation information (position, velocity, attitude, and so on) over wireless communication. The ground station *MVLOG* collects and assembles all the information and sends them to the task-assignment and path-planning modules running on a separate computer connected on a same local network. Upon receiving the current vehicles' status, it executes the task-assignment/path-planning algorithms in real time and sends the updated waypoint commands for all vehicles. *MVLOG* relays this information to all the vehicles over the wireless network. This cycle repeats every second, which is a sufficient time for vehicles flying at 20 m/s performing the scenario considered here. The proposed algorithm is originally intended for fully decentralized scenario, where each agent runs the

Fig. 63.19 A path-planning experiment result



negotiation-based task assignment algorithm onboard and the ground station only functions as an arbiter. In this experiment, however, a fully centralized solver is employed for easier management of experiment.

The first scenario for flight experiment is to avoid two known obstacles represented as green-colored polygons and one unknown obstacle in cyan color as shown in Fig. 63.19. First, the UAV is launched and commanded to fly to a predefined altitude and then loiters until the first waypoint command is received. The launch location is marked with the red circle in Fig. 63.19. From there, the airplane flies to the initial loitering position in $-y$ direction in the graph and loiters along circular patterns. Then, the airplane is commanded to perform the obstacle avoidance by following along the waypoint command sent from the path planner on the ground

Fig. 63.20 Task-assignment scenario for three UAVs with a no-fly zone



as shown in Fig. 63.19, where the airplane flies in $+y$ direction while avoiding the green- and cyan-color obstacles. The cyan-color obstacle pops up when the airplane passes by the green asterisk, which triggers the path planner to plan for a new path that does not collide with the newly detected obstacle. The UAV flies around the obstacles without any collision until it reaches the goal point on the upper side of the graph.

Then, a scenario with three UAVs to visit six task points for reconnaissance with a no-fly zone (blue polygon in Fig. 63.20) is tested. The experiment result is shown in Figs. 63.21 and 63.22, where the trajectories of three UAVs performing the scenario as well as the task points and simulated obstacles are shown. Initially, the three agents are located at the lower right area of the map and then start flying to the upper-left direction upon receiving the task commands (Moon et al. 2013).

Table 63.8 shows how the tasks are assigned to three UAVs during the mission. Based on the initial positions of the agents, tasks are initially assigned to the participating agents as shown in step 1. When task 1 is completed by UAV #1, the remaining tasks are assigned to all UAVs as shown in step 2. This time, UAV #3 completes task 5, and again the remaining tasks are assigned to all other UAVs. This procedure is repeated until all tasks are completed at step 7.

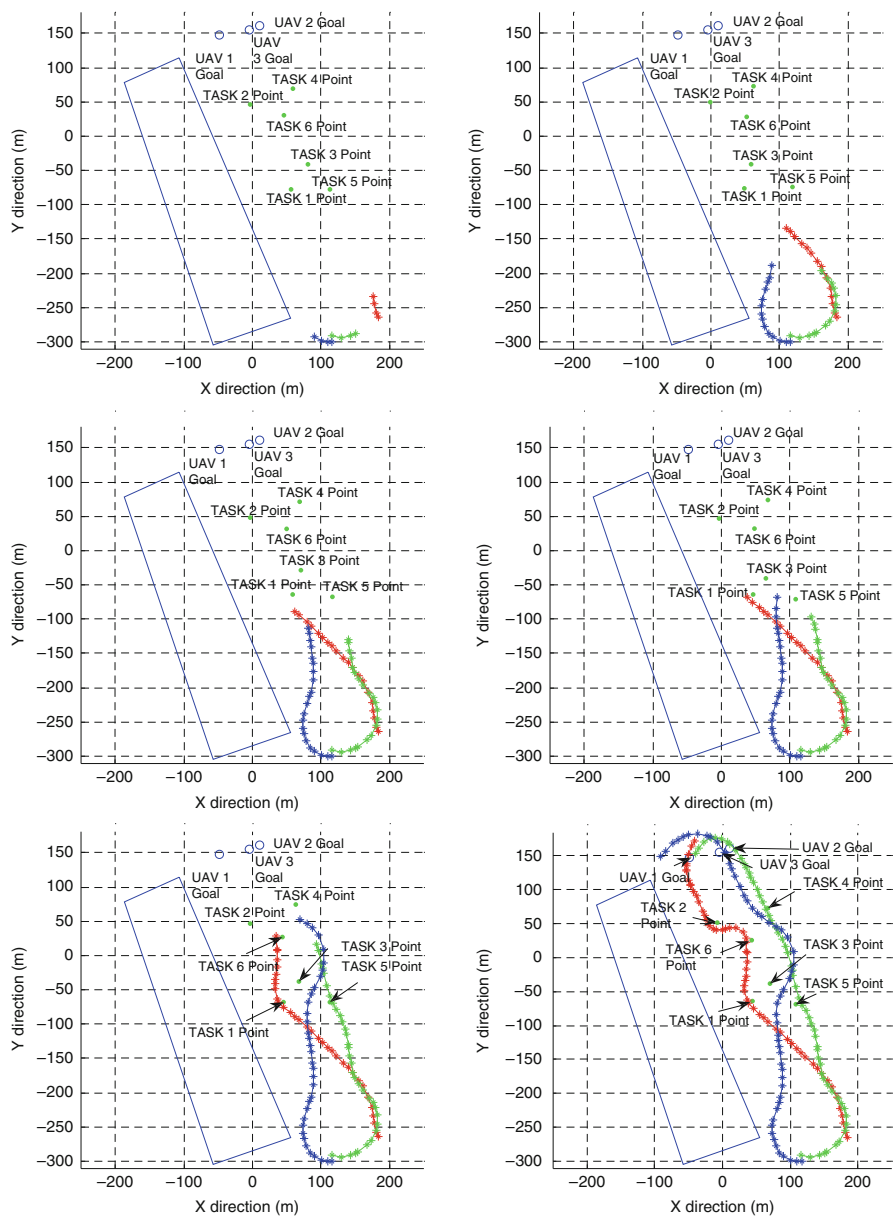


Fig. 63.21 A flight experiment result of a scenario for three UAVs with six tasks with a no-fly zone

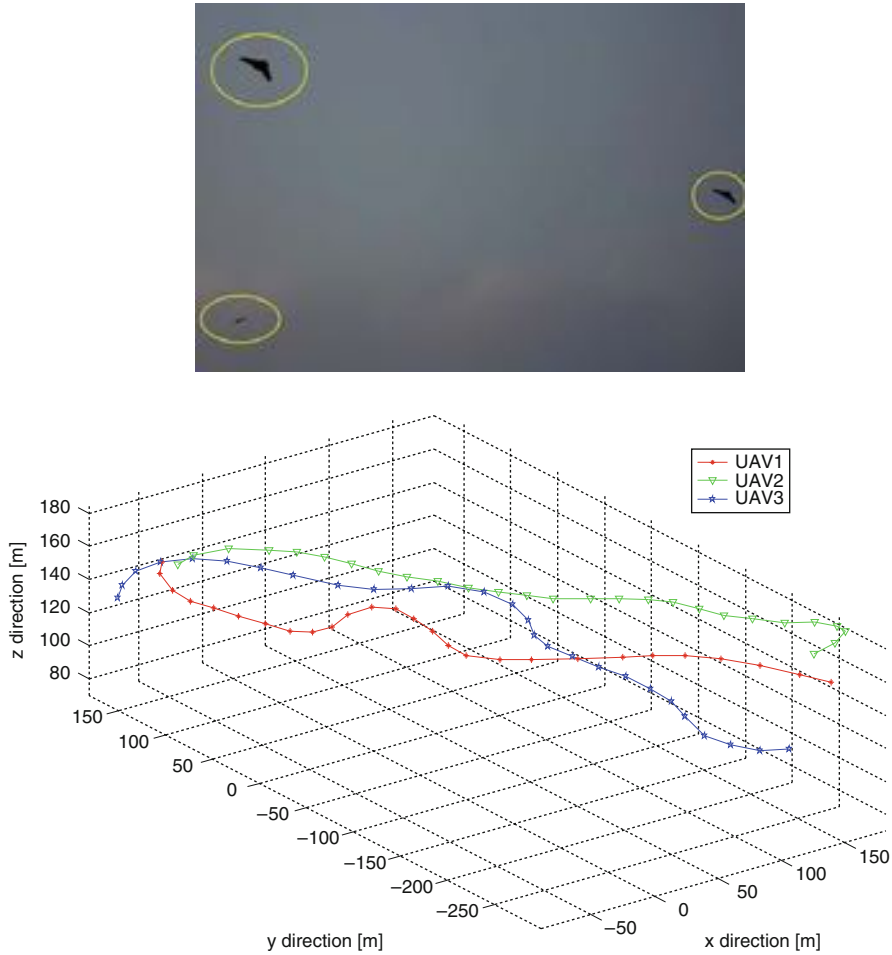


Fig. 63.22 A snapshot and a three-dimensional graph of the trajectories of three UAVs during the experiment in Fig.63.21

In the third scenario, the obstacle avoidance capability against pop-up threat in a multi-agent scenario was evaluated (Moon et al. 2013). Here, two UAVs fly into an area with four pop-up threats, where the UAVs are forbidden to fly over. The flight experiment was conducted following the procedure introduced above. The experiment result is shown in Fig. 63.23, where the three UAVs were commanded to perform the tasks given tasks while avoiding the pop-up threats using the potential field-based collision-avoidance algorithm. From this experiment, the proposed framework is shown to be capable of performing a dynamic task assignment in the presence of pop-up threats as well.

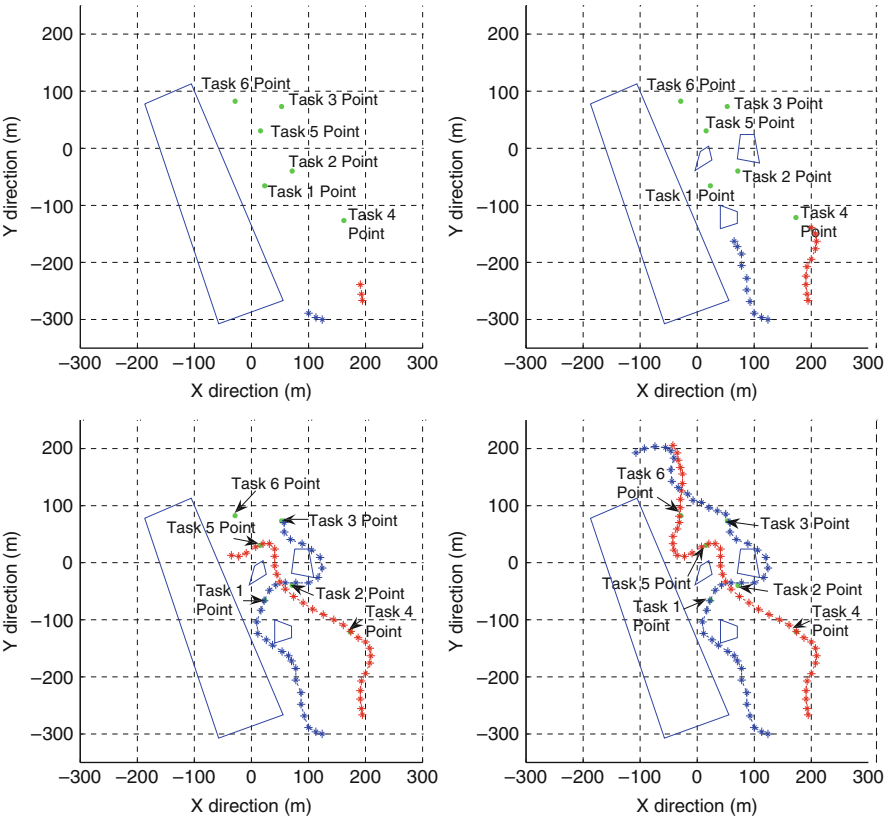


Fig. 63.23 A flight experiment result of a scenario for two UAVs with a no-fly zone and three pop-up threats

Table 63.8
Task-assignment procedure
for experiment with three
UAVs

| | UAV #1 | UAV #2 | UAV #3 |
|--------|---------------|--------|---------------|
| Step 1 | Task 1 | Task 5 | Task 3 |
| Step 2 | Task 3 | Task 6 | Task 5 |
| Step 3 | Task 6 | Task 4 | Task 3 |
| Step 4 | Task 6 | Task 4 | Task 2 |
| Step 5 | Task 2 | Finish | Task 4 |
| Step 6 | Task 2 | Finish | Finish |
| Step 7 | Finish | Finish | Finish |

Conclusion

In this chapter, a hierarchical framework for task assignment and path planning with real-time collision avoidance is proposed. The task assignment is based on a negotiation-based algorithm, and the path planner is based on the shortest-path principle combined with A^* search algorithm. Additional procedures to

improve the quality of the obtained paths are also introduced. The real-time collision-avoidance algorithm for pop-up threats or unmapped obstacles is based on potential field, which also considers the relative direction of motion. The proposed framework is first validated in simulations and then in a series of experiments using real fixed-wing UAVs in outdoor environment. The group of UAVs was able to accomplish the given missions by visiting all of the task points even in the presence of known/unknown obstacles and pop-up threats. The proposed algorithm is readily applicable to various multi-UAV scenarios in real environment unlike many algorithms that function only in sterile conditions.

References

- J. Bellingham, M. Tillerson, A. Richards, J.P. How, Multi-task allocation and path planning for cooperating UAVs, *Cooperative Control: Models, Applications and Algorithms* (Springer, 2001)
- D.P. Bertsekas, *SIAM J. Opt.* **1**(4), 425 (1991)
- D.E. Chang, S.C. Shadden, J.E. Marsden, R. Olfati-Saber, Collision avoidance for multiple agent systems, in *42nd IEEE Conference on Decision and Control*, Maui, 2003
- H.L. Choi, L. Brunet, J.P. How, *IEEE Trans. Robot.* **25**(4), 912 (2009)
- M. Earl, R. D'Andrea, *IEEE Trans. Robot.* **21**, 1158 (2005)
- P. Hart et al., *IEEE Trans. Syst. Sci. Cybern.* **4**(2), 100 (1968)
- V. Jain, I.E. Grossmann, *Inf. J. Comput.* **13**, 258 (2001)
- J. Latombe, *Robot Motion Planning* (Kluwer Academic, Boston, 1991)
- S. Lavalle, *Planning Algorithms* (Cambridge University Press, New York, 2006)
- I. Maza, F. Caballero, J. Capitan, J.R. Martinez-de-Dios, A. Ollero, *J. Intell. Robot. Syst.* **61**, 563 (2011)
- S. Moon, D.H. Shim, Development of an efficient path planning algorithm using UAVs in cluttered environment, in *Institute of Control, Robotics and Systems Conference*, Chuncheon, 2010
- S. Moon, E. Oh, and D.H. Shim, *J. Intell. Robot. Syst.* **70**, 303 (2013)
- P. Pettersson, P. Doherty, *J. Intell. Fuzzy Syst.* **17**(4), 395 (2006)
- J. Redding, J. Amin, J. Boskovic, Y. Kang, K. Hedrick, A real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters, in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, 2007
- W. Ren, R.W. Beard, E.M. Atkins, *IEEE Control Syst. Mag.* **27**, 71 (2007)
- A. Richards, Y. Kuwata, J. How, Experimental demonstrations of real-time MILP control, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Reno, 2003
- T. Schouwenaars, B.D. Moor, E. Feron, J. How, Mixed integer programming for multi-vehicle path planning, in *European Control Conference*, Porto, 2001
- D.H. Shim, A dynamic path generation method for a UAV swarm in the Urban environment, in *AIAA Guidance, Navigation, and Control Conference*, Honolulu, 2008
- P.B. Sujit, A. Sinha, D. Ghose, *Stud. Comput. Intell.* **70**, 39 (2007)
- A. Viguria, I. Maza, A. Ollero, *Adv. Robot.* **24**, 1 (2010)
- K. Yang, S. Gan, S. Sukkarieh, *J. Intell. Robot. Syst.* **57**, 101 (2010)

Michael J. Hirsch and Daniel Schroeder

Contents

64.1 Introduction 1578

64.2 Programming Formulation 1581

 64.2.1 Problem Definition 1581

 64.2.2 Parameters 1583

 64.2.3 Decision Variables 1584

 64.2.4 Constraints 1586

 64.2.5 Objective Function (for UAV *i*) 1589

 64.2.6 Discussion on Complexity 1590

64.3 Heuristic Algorithm 1590

64.4 Experimental Results 1591

 64.4.1 Test Environment 1591

 64.4.2 Experiments 1591

64.5 Conclusions and Future Research Areas 1597

References 1598

Abstract

This chapter is concerned with dynamically determining appropriate flight patterns for a set of autonomous UAVs in an urban environment, with multiple mission goals. The UAVs are tasked with searching the urban region for targets of interest and tracking those targets that have been detected. It is assumed that there are limited communication capabilities between the UAVs and that there exist possible line of sight constraints between the UAVs and the targets. Each UAV (*i*) operates its own dynamic feedback loop, in a receding-horizon

M.J. Hirsch (✉)
Intelligence, Information, and Services, Raytheon Company, Maitland, FL, USA
e-mail: mhirsch@iseatek.com; www.iseatek.com

D. Schroeder
Intelligence, Information, and Services, Raytheon Company, State College, PA, USA
e-mail: Daniel.E.Schroeder@raytheon.com

framework, incorporating local information (from UAV i perspective) as well as remote information (from the perspective of the “neighbor” UAVs) to determine the task to perform and the optimal flight path of UAV i over the planning horizon. This results in a decentralized and more realistic model of the real-world situation. As the coupled task assignment and flight route optimization formulation is NP-hard, a hybrid heuristic for continuous global optimization is developed to solve for the flight plan and tasking over the planning horizon. Experiments are considered as communication range between UAVs varies.

64.1 Introduction

Military forces face ever increasing challenges to provide timely and accurate information on targets of interest, especially those targets that are mobile and elusive in nature. These tasks expand significantly in complexity when the targets are operating in an urban environment. The use of multiple unmanned vehicles to provide this target information allows our military personnel to stay out of the line of fire. However, many remotely controlled UAVs (i.e., swarms) require as many skilled pilots as there are swarm members, and these pilots must be able to deconflict airspace demands, mission requirements, and situational changes in near real time (Bamburger et al. 2006). On the other hand, autonomous unmanned vehicles allow military personnel to focus on more important issues like interpreting the gathered information, as opposed to determining how to acquire the information (Hirsch et al. 2007). Hence, there is a need to build intelligent unmanned vehicles that can plan and adapt autonomously to the environment they perceive. The clear benefit is shortened mission-critical decision chains.

There has recently been much research done in the area of autonomous vehicle control for surveillance-type missions. Almost all of the research has dealt with centralized cooperative control, with little research concerned with the more realistic decentralized problem (Shima and Rasmussen 2009). Steinberg (2006) provided an overview on research and limitations of autonomous technologies for the control of heterogeneous unmanned naval vehicles. Experiments in this paper examined aspects such as multi-vehicle task allocation, dynamic replanning of vehicle tasks, as well as human-in-the-loop management. Constraints considered in the experiments included pop-up threats, adverse weather conditions, and communication issues between the autonomous vehicles, among others. Liang et al. (2010) present a decision support framework to accurately reconstruct the current picture of the battlespace, by correctly identifying target tracks (i.e., target location, movement, identity) as seen by multiple wide area search munitions. Ahmadzadeh et al. (2006) described their Time Critical Coverage planner as a component of the Office of Naval Research Autonomy program, ICARUS. Each autonomous vehicle was modeled as a Dubin’s vehicle (Dubins 1957), whereby the vehicles were assumed to be point masses with constant speed and a prescribed minimum turning radius. The vehicles also had prescribed starting and ending spatial-temporal locations, as well as polygonal obstacles to be avoided throughout flight. The objective was to

determine the flight path of the UAVs to maximize the total sensor footprint over the region of interest. The algorithm utilized to solve this problem was based on sampling a discretized search graph (LaValle 2006).

Ma and Miller (2005) implemented a receding-horizon approach to solve a mixed-integer linear program modeling a trajectory planning problem. They limited their focus to a single vehicle navigating through three-dimensional obstacles and terrain. They considered threat avoidance and made use of the commercial software CPLEX (ILOG CPLEX 2008) to solve their problem. Shetty et al. (2008) considered the strategic routing of multiple unmanned combat vehicles to service multiple potential targets in space. They formulated this as a mixed-integer linear program, and through a decomposition scheme looked at solving the target assignment problem (vehicles to targets) and then determining the tour that each vehicle should take to service their assigned targets (a classical vehicle routing problem). They implemented a tabu-search heuristic to find solutions to their problem. However, they assumed the vehicles were holonomic, which enabled the mixed-integer linear program formulation. Kenefic (2008) utilized techniques from particle swarm optimization (Kennedy and Eberhart 1995) to efficiently define tours for a Dubin's vehicle, visiting multiple ground targets in two-dimensional space. Schumacher and Shima (2009) considered the problem of wide area search munitions, which are capable of searching for, identifying, and attacking targets. Whenever a new target is found, or a new task needs to be assigned, a capacitated transshipment assignment problem is solved, to determine the optimal assignment of munitions to tasks. Note that from one solution to the next solution, the assignment can change significantly.

Sylvester et al. (2004) presented a path-planning algorithm for two non-holonomic vehicles engaged in a docking maneuver. A bijection search method was used to determine the trajectory of the vehicles, such that the minimum turning radius constraint of each vehicle was satisfied. Santilli et al. (1995) and Bicchi et al. (1995) considered planning a path for a non-holonomic autonomous vehicle in the presence of fixed obstacles. Jiang et al. (1999) also presented a path planning algorithm for a non-holonomic vehicle subject to obstacles. This algorithm employed a global and local strategy to find the shortest path in two dimensions that was free of collisions with the obstacles. Schouwenaars et al. (2001) considered fuel-optimal paths for multiple vehicles. They formulated the problem as a mixed-integer linear program. The vehicles needed to move from an initial to final state, while avoiding vehicular collisions, as well as stationary and moving obstacles. Obstacle positions were assumed known a priori.

Kingston and Beard (2007) presented an algorithm to keep moving UAVs equally spaced (angularly) about a stationary target. The UAVs adapted their spacing based on local communication with other UAVs. Velocity bounds were derived so that the UAVs satisfied their kinematic constraints. They extended this approach to the case when the target was moving, but the path and velocity of the target was assumed known throughout. Gu et al. (2006) discussed the problem of cooperative estimation using a team of UAVs. They considered the goal of organizing the team's configuration to achieve optimal position and velocity estimates of a moving ground target. Sinclair et al. (2007) and Doucette et al. (2009) considered a problem

of cooperative control of aerial munitions during the attack phase of a stationary ground target, in order to refine the estimates of the target location. A dynamic programming solution was implemented to determine the paths for the aerial munitions. McLain et al. (2000) presented a cooperative control problem for UAVs to rendezvous at given locations, at the same time. The UAVs considered threat and fuel constraints, in determining their trajectories. A decomposition approach was employed, whereby at the team level, an arrival time was determined, and at the individual UAV level, an optimal path to reach the specified location at the arrival time was determined.

Rafi et al. (2006) presented an approach to have an autonomous vehicle follow a moving ground target. Their technique specified positional updates for the vehicle, in terms of the next time-step turning rate as well as orientation updates for the gimballed camera mounted on the vehicle. They tested their approach using one vehicle and one moving ground target. Sinha et al. (2005) considered cooperative control for tracking of targets by multiple UAVs. An information theoretic criterion was used to determine the future path of each UAV, so that the total information gain (corresponding to the detected targets) was maximized. The UAVs were assumed to have fixed speed, and each UAV determined its angular turn rate to maximize the expected information gain for the next-time step. Akselrod et al. (2007) presented an approach for multi-target tracking of ground targets using UAVs via an hierarchical Markov decision process framework and dynamic programming. Their objective was to maximize the information gained by the UAVs. Sinha et al. (2004) presented a method to determine the location of sensors for tracking of moving ground targets, so as to maximize the resultant information gain. A genetic algorithm was implemented to determine the sensor positions. Of note here is that it was assumed that the targets were moving in the same direction as the sensors and their relative position remained constant over time.

Close-range maneuvering and following of a moving target continue to pose significant research challenges (Rafi et al. 2006). Close-range maneuvering requires real-time dynamic replanning and decision making as well as optimization of many parameters (taking into account the physical constraints of the vehicle). The problem becomes even more complex when the goal is to track a moving target for which the target dynamics is not known. In Hirsch et al. (2012, 2011) and Hirsch and Ortiz-Pena (2010), Hirsch et al. developed a decentralized approach for the autonomous cooperative control of UAVs, with the goal of persistent and accurate tracking of moving ground targets in an urban domain. The UAVs were able to share limited information with neighboring UAVs (other UAVs in their communication region) and had to dynamically replan their flight paths, incorporating predicted target movements and replanned flight paths of their neighbors into their own decision-making process. Line of sight also plays a big role in the flight plan of the UAVs and was incorporated into the problem through the use of Plücker coordinates (Semple and Kneebone 1952; Stolfi 1991). However, it was assumed that the targets were already known.

In this chapter, the approach of Hirsch et al. (2012, 2011) is extended to include, in a decentralized framework, the UAVs actively searching for and detecting

the targets, before tracking of the targets can commence. Since the UAVs have no knowledge of how many targets are present in the environment, at each decision-making step, the UAVs need to determine both the task they will perform (either searching for new targets or tracking those targets already detected) as well as determining the flight route to optimally accomplish their task. This results in a coupled task assignment/route planning mathematical formulation, which is highly nonlinear. A GRASP-SA heuristic is developed to efficiently solve the coupled problem. The solution quality is examined to determine its sensitivity to the communication capability of the UAVs. This chapter is organized as follows. Section 64.2 rigorously defines the problem of interest, discusses the dynamic feedback loop, and formulates the coupled task assignment/route planning problem as a nonlinear mathematical program. A hybrid heuristic for continuous global optimization is described in Sect. 64.3. This heuristic is utilized to solve the optimization problem at each time-step, for each UAV. Experimental results as the communication radius of the UAVs is varied are presented in Sect. 64.4. Section 64.5 provides some concluding remarks and future research directions.

64.2 Programming Formulation

64.2.1 Problem Definition

The overall problem addressed here is to provide (i) continual surveillance over a region of interest and (ii) accurately track all targets detected in this region. Hence, there are two tasks that a UAV can be performing at any given time: searching for targets or tracking those targets already detected. For this problem, there are N homogeneous autonomous vehicles flying at a fixed altitude. The vehicles are modeled as non-holonomic point masses on a two-dimensional plane with a minimum turning radius (i.e., a Dubin's vehicle (Dubins 1957)). The vehicles have minimum and maximum speed restrictions, as well as a maximum communication range, beyond which they cannot share information. Each vehicle has an independent internal representation of the surveillance region as a continuous search uncertainty map. For each location \vec{Y} in the region, the search uncertainty map for a particular UAV stores the uncertainty (from this UAV's perspective) of a target currently at location \vec{Y} . While in search mode, the UAV can update its search uncertainty map using information from its own onboard sensor as well as information from other UAVs currently in search mode (that can communicate with this UAV, i.e., its "neighbor" UAVs). While in tracking mode, the UAV can only update its search uncertainty map using information from neighbor UAVs that are in search mode. Similarly, each UAV has its own internal representation of the targets in the region of interest. When in search mode, the UAV can only update its target map from those neighbor UAVs in track mode. When in track mode, the UAV can update its target map based upon its local sensors as well as information provided from neighbor UAVs in track mode. The one exception to this is if in search mode, it is possible

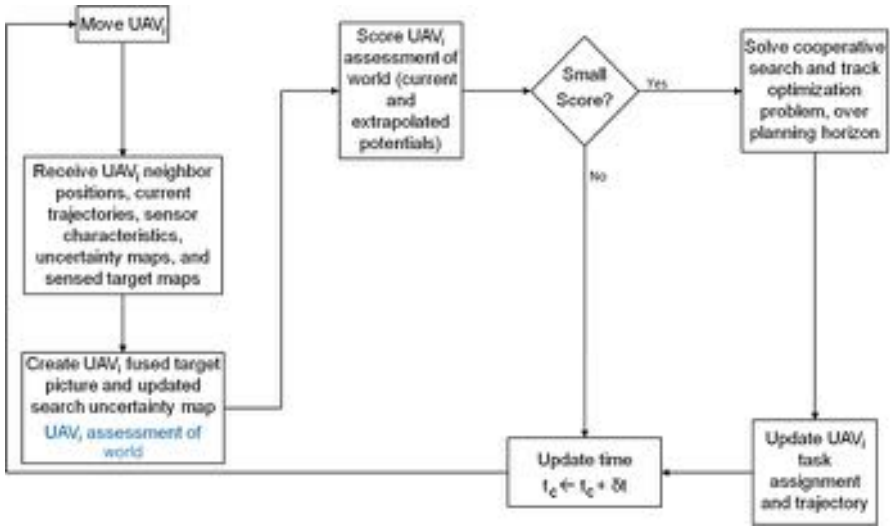


Fig. 64.1 Feedback control loop for UAV_{*i*}

(and hoped) that a UAV will detect a “new” target. When this happens, this target is added to the UAV’s stored target map, and shared with neighbor UAVs.

The vehicles operate in a decentralized manner; at each time-step, every UAV processes an equivalent version of the same dynamic feedback loop, as presented in Fig. 64.1. The feedback loop is discussed from the standpoint of UAV *i*. At the current time, t_c , the first step is for UAV *i* to move according to its planned route and continue to execute its current task (e.g., searching or tracking). Using onboard sensors, UAV *i* either searches its current field of view for new targets or attempts to sense the targets (known to UAV *i*) that are within its line of sight. The neighbors of UAV *i* send information relating to their (a) current positions, trajectories, and tasking; (b) sensor characteristics; (c) search uncertainty maps; and (d) sensed target maps. At this point, UAV *i* creates its assessment of the world, represented as a fused search uncertainty map and a fused target location map. This fused picture is scored in some qualitative manner, based upon the current positions of UAV *i* and its neighbor UAVs, their current planned trajectories and tasking, as well as the extrapolated search uncertainty map and target tracks, over the planning horizon. If the resultant score is small, or if the elapsed time since the last replan for UAV *i* is large, then UAV *i* solves a coupled tasking/route planning optimization problem for itself and its neighbor UAVs, over the planning horizon. The solution to this problem is a tasking for each UAV and route for each UAV to fly in order to accomplish their tasking. The mathematical formulation for this optimization problem is detailed in Sects. 64.2.2–64.2.5. It is worth noting that UAV *i* updates its own flight path and tasking, but does not share any of this computed information with its neighbor

UAVs. The reason for this is that the neighbors are going through this same feedback loop and may be incorporating additional knowledge not available to UAV i . The time is then incremented, and the dynamic loop continues.

64.2.2 Parameters

This section introduces the parameters incorporated into the dynamic tasking and flight plan optimization formulation. For the mathematical formulation to follow, the parameters, decision variables, and constraints are defined for all $\hat{i} \in \mathcal{N}_i(t)$, $t \in [t_c, t_c + \Delta_t]$, $\ell \in \{1, \dots, 5\}$, $h \in \{1, \dots, \mathcal{H}\}$, $j \in \{1, \dots, \mathcal{J}\}$, $u \in \{1, \dots, 4\}$, as appropriate:

t_c = The current time;

Δ_t = The current planning horizon;

C_d = The maximum distance at which any two UAVs can “communicate”, i.e., share information;

$\mathcal{N}_i(t)$ = The neighbors of UAV i , at time t , i.e., $\hat{i} \in \mathcal{N}_i(t) \implies \|P_i(t) - P_{\hat{i}}(t)\|_2 \leq C_d$. N.B.: $i \in \mathcal{N}_i(t)$;

The UAVs of interest (from UAV i ’s perspective) are indexed by $\hat{i} \in \mathcal{N}_i(t)$;

The moving ground targets are indexed by $j \in \{1, \dots, \mathcal{J}\}$;

The buildings (3D, rectangular, opaque obstacles) are indexed by $h \in \{1, \dots, \mathcal{H}\}$;

$\Omega_{\hat{i}}$ = The minimum turning radius of UAV \hat{i} ;

$P_{\hat{i}}^c$ = The current position of UAV \hat{i} (N.B.: $P_{\hat{i}}^c = \{x_{\hat{i}}^c, y_{\hat{i}}^c, z_{\hat{i}}^c\}$);

$S_{\hat{i}}^{\min}$ = The minimum speed of UAV \hat{i} ;

$S_{\hat{i}}^{\max}$ = The maximum speed of UAV \hat{i} ;

ϵ_1^{\min} = The minimum distance allowed between any two UAVs (to avoid UAV collisions);

ϵ_2^{\min} = The minimum distance allowed between any UAV and any known building. Assume this is set to 0, i.e., so that UAVs are constrained to not be within a building, but they can get infinitesimally close to the perimeter of each building.

N.B.: this assumption is valid, as one can always artificially increase the size of the buildings, as appropriate;

$Z_{\hat{i}}$ = The fixed altitude of UAV \hat{i} ;

$\mathcal{B}_h = \{\mathcal{P}_{h\ell}\}_{\ell=1}^5$ = The representation of building h , as a set of 5 rectangular regions (i.e., faces) in \mathbb{R}^3 , all oriented in the same counterclockwise fashion;

$\mathcal{P}_{h\ell} = \{\rho_{h\ell 1}, \dots, \rho_{h\ell 4}\}$ = The ℓ -th oriented rectangle (i.e., face) of building h ;

$\rho_{h\ell u} = \{\bar{x}_{h\ell u}, \bar{y}_{h\ell u}, \bar{z}_{h\ell u}\}$ = The u -th point of the ℓ -th oriented rectangle (i.e., face) of building h ;

$\xi_{h\ell} = \{v_{h\ell}; d_{h\ell}\}$ = The plane containing face $\mathcal{P}_{h\ell}$, represented as a normal vector $v_{h\ell}$ and distance to the origin $d_{h\ell}$ ($v_{h\ell} = [\rho_{h\ell 3} - \rho_{h\ell 2}] \times [\rho_{h\ell 1} - \rho_{h\ell 2}]$, $d_{h\ell} = v_{h\ell} \cdot \rho_{h\ell 1}$);

$\gamma(x, y)$ = A function representing the “importance” level of position (x, y) . The higher this value, the more important it is to provide accurate track of targets near (x, y) ;

$\Lambda_{\hat{i}j}(t)$ = The Plücker coordinate representation (Semple and Kneebone 1952; Stolfi 1991) of the oriented line from the position of UAV \hat{i} at time t , $P_{\hat{i}}(t)$, to the estimated position of target j , at time t , $X_j^i(t)$;

$\Lambda_{\hat{i}}(\vec{Y}, t)$ = The Plücker coordinate representation of the oriented line from the position of UAV \hat{i} at time t , $P_{\hat{i}}(t)$, to position \vec{Y} ;

$\hat{\Lambda}_{h\ell u, u+1}$ = The Plücker coordinate representation of the oriented line from $\rho_{h\ell u}$ to $\rho_{h\ell, u+1}$, with $u + 1 \equiv 1$ when $u = 4$;

$\mathcal{G}(\alpha, \hat{\alpha}) = \alpha_1 \hat{\alpha}_6 - \alpha_2 \hat{\alpha}_5 + \alpha_3 \hat{\alpha}_4 + \alpha_4 \hat{\alpha}_3 - \alpha_5 \hat{\alpha}_2 + \alpha_6 \hat{\alpha}_1$ where $\alpha = \{\alpha_1, \dots, \alpha_6\}$ and $\hat{\alpha} = \{\hat{\alpha}_1, \dots, \hat{\alpha}_6\}$ are the Plücker coordinate representations of two oriented lines;

$R_{\hat{i}}(t)$ = The measurement noise for the sensor onboard UAV \hat{i} , at time t . N.B.:

$R_{\hat{i}}(t) = \begin{pmatrix} \sigma_{r\hat{i}} & 0 \\ 0 & \sigma_{b\hat{i}} \end{pmatrix}$, where $\sigma_{r\hat{i}}$ = the range measurement error for the radar platform on UAV \hat{i} and $\sigma_{b\hat{i}}$ = the bearing measurement error for the radar platform on UAV \hat{i} ;

$F_{\hat{i}}(t)$ = The system dynamics matrix for the sensor onboard UAV \hat{i} , at time t ;

$\bar{Q}_{\hat{i}}(t)$ = The process noise for the sensor onboard UAV \hat{i} , at time t ;

$H_{\hat{i}}(t)$ = The observation matrix for the sensor onboard UAV \hat{i} , at time t ;

$\bar{d}_{\hat{i}}$ = The probability of detection for the sensor onboard UAV \hat{i} ;

χ = The factor by which to increase the search uncertainty map if there is no line of sight to a given position (in Constraint (64.9));

$\mathcal{M}, \bar{\mathcal{M}}$ = Large enough constants (needed for Constraints (64.7), (64.12), (64.13), (64.15), (64.16), (64.18), (64.19), (64.21), (64.22), (64.28), (64.29), (64.31), (64.32), (64.34), (64.35), (64.37), and (64.38)).

64.2.3 Decision Variables

This section defines the main decision variables for the mathematical formulation:

$P_{\hat{i}}(t)$ = The position of UAV \hat{i} , at time t (N.B.: $P_{\hat{i}}(t) = \{x_{\hat{i}}(t), y_{\hat{i}}(t), Z_{\hat{i}}\}$);

$S_{\hat{i}}(t)$ = The speed of UAV \hat{i} , at time t ;

$\theta_{\hat{i}}(t)$ = The heading of UAV \hat{i} , at time t ;

$\{X_{\hat{i}j}(t), \Sigma_{\hat{i}j}(t)\}$ = The estimated position and covariance of target j by UAV \hat{i} , at time t ;

$\{X_j^i(t), \Sigma_j^i(t)\}$ = The fused position and covariance of target j , from UAV i 's perspective, at time t ;

$\bar{\Phi}_{\hat{i}jh\ell u, u+1}(t) = \begin{cases} 1 & \text{if line segment } \Lambda_{\hat{i}j}(t) \text{ is oriented counterclockwise} \\ & \text{with respect to line segment } \hat{\Lambda}_{h\ell u, u+1}, \text{ at time } t \\ 0 & \text{o.w.} \end{cases}$;

$\bar{\Phi}_{\hat{i}jh\ell}(t)$ = The number of line segments representing the ℓ -th face of building h that line segment $\Lambda_{\hat{i}j}(t)$ is oriented counterclockwise with respect to, at time t ;

$\beta_{\hat{i}jh\ell}(t) = \begin{cases} 1 & \text{if line segment } \Lambda_{\hat{i}j}(t) \text{ does not intersect the } \ell\text{-th face} \\ & \text{of building } h \text{ at time } t \\ 0 & \text{o.w.} \end{cases}$;

$\bar{\beta}_{\hat{i}jh}(t)$ = The number of faces representing building h that line segment $\Lambda_{\hat{i}j}(t)$ does not intersect, at time t ;

$$\bar{\psi}_{\hat{i}jh}(t) = \begin{cases} 1 & \text{if there is line of sight from UAV } \hat{i} \text{ to target } j \text{ with respect to} \\ & \text{building } h, \text{ at time } t \\ 0 & \text{o.w.} \end{cases};$$

$\hat{\psi}_{\hat{i}j}(t)$ = The number of buildings that do not restrict line of sight from UAV \hat{i} to target j , at time t ;

$$\psi_{\hat{i}j}(t) = \begin{cases} 1 & \text{if UAV } \hat{i} \text{ has line of sight to target } j, \text{ at time } t; \\ 0 & \text{o.w.} \end{cases};$$

$$b_{\hat{i}} = \begin{cases} 1 & \text{if UAV } \hat{i} \text{ assigned to search mode over the planning horizon} \\ & [t_c, t_c + \delta_t] \\ 0 & \text{if UAV } \hat{i} \text{ assigned to track mode over the planning horizon} \\ & [t_c, t_c + \delta_t] \end{cases};$$

$\Gamma_i(\vec{Y}, t)$ = Search uncertainty map for UAV i at time t and position \vec{Y} ;

$$\bar{\Phi}_{\hat{i}h\ell u, u+1}(\vec{Y}, t) = \begin{cases} 1 & \text{if line segment } \Lambda_{\hat{i}}(\vec{Y}, t) \text{ is oriented counterclockwise} \\ & \text{with respect to line segment } \hat{\Lambda}_{h\ell u, u+1}, \text{ at time } t \\ 0 & \text{o.w.} \end{cases};$$

$\bar{\Phi}_{\hat{i}h\ell}(\vec{Y}, t)$ = The number of line segments representing the ℓ -th face of building h that line segment $\Lambda_{\hat{i}}(\vec{Y}, t)$ is oriented counterclockwise with respect to, at time t ;

$$\beta_{\hat{i}h\ell}(\vec{Y}, t) = \begin{cases} 1 & \text{if line segment } \Lambda_{\hat{i}}(\vec{Y}, t) \text{ does not intersect the } \ell\text{-th face} \\ & \text{of building } h, \text{ at time } t \\ 0 & \text{o.w.} \end{cases};$$

$\bar{\beta}_{\hat{i}h}(\vec{Y}, t)$ = The number of faces representing building h that line segment $\Lambda_{\hat{i}}(\vec{Y}, t)$ does not intersect, at time t ;

$$\bar{\psi}_{\hat{i}h}(\vec{Y}, t) = \begin{cases} 1 & \text{if there is line of sight from UAV } \hat{i} \text{ to position } \vec{Y} \text{ with respect} \\ & \text{to building } h, \text{ at time } t \\ 0 & \text{o.w.} \end{cases};$$

$\hat{\psi}_{\hat{i}}(\vec{Y}, t)$ = The number of buildings that do not restrict line of sight from UAV \hat{i} to position \vec{Y} , at time t ;

$$\psi_{\hat{i}}(\vec{Y}, t) = \begin{cases} 1 & \text{if UAV } \hat{i} \text{ has line of sight to position } \vec{Y}, \text{ at time } t; \\ 0 & \text{o.w.} \end{cases};$$

$$\psi_o(\vec{Y}, t) = \begin{cases} 1 & \text{if neither UAV } i \text{ nor any neighbors of UAV } i \text{ has line of sight} \\ & \text{to position } \vec{Y}, \text{ at time } t \\ 0 & \text{o.w.} \end{cases};$$

64.2.4 Constraints

$$P_{\hat{t}}(t_c) = P_{\hat{t}}^c \quad \forall \hat{t} \quad (64.1)$$

$$\frac{\partial x_{\hat{t}}(t)}{\partial t} = \mathcal{S}_{\hat{t}}(t) \cos [\theta_{\hat{t}}(t)] \quad \forall \hat{t}, t \quad (64.2)$$

$$\frac{\partial y_{\hat{t}}(t)}{\partial t} = \mathcal{S}_{\hat{t}}(t) \sin [\theta_{\hat{t}}(t)] \quad \forall \hat{t}, t \quad (64.3)$$

$$\left| \frac{\partial \theta_{\hat{t}}(t)}{\partial t} \right| \leq \Omega_{\hat{t}} \quad \forall \hat{t}, t \quad (64.4)$$

$$\mathcal{S}_{\hat{t}}^{\min} \leq \mathcal{S}_{\hat{t}}(t) \leq \mathcal{S}_{\hat{t}}^{\max} \quad \forall \hat{t}, t \quad (64.5)$$

$$\|P_{\hat{t}_1}(t) - P_{\hat{t}_2}(t)\|_2 \geq \varepsilon_1^{\min} \quad \forall \hat{t}_1, \hat{t}_2, t \quad (64.6)$$

$$P_{\hat{t}}(t) \cdot v_{h\ell} + d_{h\ell} \geq -\mu_{\hat{t}h\ell}(t)\mathcal{M} \quad \forall \hat{t}, h, \ell, t \quad (64.7)$$

$$\sum_{\ell=1}^5 \mu_{\hat{t}h\ell}(t) \leq 4 \quad \forall \hat{t}, h, \ell, t \quad (64.8)$$

$$\begin{aligned} \frac{\partial \Gamma_i(\vec{Y}, t)}{\partial t} = & \left[\prod_{\hat{t} \in \mathcal{N}_i(t)} (1 - \bar{d}_{\hat{t}})^{b_{\hat{t}}\psi_{\hat{t}}(\vec{Y}, t)} \right] \cdot \Gamma_i(\vec{Y}, t) \\ & + \psi_o(\vec{Y}, t) \cdot \chi \cdot \Gamma_i(\vec{Y}, t) \quad \forall t, \vec{Y} \end{aligned} \quad (64.9)$$

$$\psi_o(\vec{Y}, t) \leq 1 - b_{\hat{t}}\psi_{\hat{t}}(\vec{Y}, t) \quad \forall \hat{t}, t, \vec{Y} \quad (64.10)$$

$$\psi_o(\vec{Y}, t) + \sum_{\hat{t} \in \mathcal{N}_i(t)} b_{\hat{t}}\psi_{\hat{t}}(\vec{Y}, t) > 0 \quad \forall t, \vec{Y} \quad (64.11)$$

$$\mathcal{G}(\Lambda_{\hat{t}}(\vec{Y}, t), \hat{\Lambda}_{h\ell u, u+1}) - (1 - b_{\hat{t}})\bar{\mathcal{M}} \leq \bar{\phi}_{\hat{t}h\ell u, u+1}(\vec{Y}, t)\mathcal{M} \quad \forall \hat{t}, \vec{Y}, h, \ell, u, t \quad (64.12)$$

$$-\mathcal{G}(\Lambda_{\hat{t}}(\vec{Y}, t), \hat{\Lambda}_{h\ell u, u+1}) - (1 - b_{\hat{t}})\bar{\mathcal{M}} < \left(1 - \bar{\phi}_{\hat{t}h\ell u, u+1}(\vec{Y}, t)\right)\mathcal{M} \quad \forall \hat{t}, \vec{Y}, h, \ell, u, t \quad (64.13)$$

$$\bar{\phi}_{\hat{t}h\ell}(\vec{Y}, t) = \sum_{u=1}^4 \bar{\phi}_{\hat{t}h\ell u, u+1}(\vec{Y}, t) \quad \forall \hat{t}, \vec{Y}, h, \ell, t \quad (64.14)$$

$$\beta_{\hat{t}h\ell}(\vec{Y}, t) - (1 - b_{\hat{t}})\bar{\mathcal{M}} \leq 4 - \bar{\phi}_{\hat{t}h\ell}(\vec{Y}, t) \quad \forall \hat{t}, \vec{Y}, h, \ell, t \quad (64.15)$$

$$\beta_{\hat{t}h\ell}(\vec{Y}, t)\mathcal{M} + (1 - b_{\hat{t}})\bar{\mathcal{M}} \geq 4 - \bar{\phi}_{\hat{t}h\ell}(\vec{Y}, t) \quad \forall \hat{t}, \vec{Y}, h, \ell, t \quad (64.16)$$

$$\bar{\beta}_{\hat{t}h}(\vec{Y}, t) = \sum_{\ell=1}^5 \beta_{\hat{t}h\ell}(\vec{Y}, t) \quad \forall \hat{t}, \vec{Y}, h, t \quad (64.17)$$

$$\bar{\psi}_{\hat{t}h}(\vec{Y}, t) - (1 - b_{\hat{t}})\bar{\mathcal{M}} \leq \beta_{\hat{t}h\ell}(\vec{Y}, t) \quad \forall \hat{t}, \vec{Y}, h, t \quad (64.18)$$

$$1 - \bar{\psi}_{ih}(\vec{Y}, t) - (1 - b_i) \bar{\mathcal{M}} \leq 5 - \bar{\beta}_{ih}(\vec{Y}, t) \quad \forall \hat{i}, \vec{Y}, h, t \quad (64.19)$$

$$\hat{\psi}_{\hat{i}}(\vec{Y}, t) = \sum_{h=1}^{\mathcal{H}} \bar{\psi}_{ih}(\vec{Y}, t) \quad \forall \hat{i}, \vec{Y}, t \quad (64.20)$$

$$\psi_{\hat{i}}(\vec{Y}, t) - (1 - b_i) \bar{\mathcal{M}} \leq \bar{\psi}_{ih}(\vec{Y}, t) \quad \forall \hat{i}, \vec{Y}, t \quad (64.21)$$

$$1 - \psi_{\hat{i}}(\vec{Y}, t) - (1 - b_i) \bar{\mathcal{M}} \leq \mathcal{H} - \hat{\psi}_{\hat{i}}(\vec{Y}, t) \quad \forall \hat{i}, \vec{Y}, t \quad (64.22)$$

$$\Sigma_j^i(t) = \left[\sum_{\hat{i} \in \mathcal{N}_i(t)} (1 - b_{\hat{i}}) \psi_{\hat{i}j}(t) \Sigma_{\hat{i}j}^{-1}(t) \right]^{-1} \quad \forall j, t \quad (64.23)$$

$$X_j^i(t) = \Sigma_j^i(t) \cdot \left[\sum_{\hat{i} \in \mathcal{N}_i(t)} (1 - b_{\hat{i}}) \psi_{\hat{i}j}(t) \Sigma_{\hat{i}j}^{-1}(t) X_{\hat{i}j}(t) \right] \quad \forall j, t \quad (64.24)$$

$$\frac{\partial X_{\hat{i}j}(t)}{\partial t} = (1 - b_i) [F_i(t) X_{\hat{i}j}(t) + K_i(t) [\bar{z}_{ij}(t) - H_i(t) X_{\hat{i}j}(t)]] \quad \forall \hat{i}, j, t \quad (64.25)$$

$$\begin{aligned} \frac{\partial \Sigma_{\hat{i}j}(t)}{\partial t} = & (1 - b_i) [F_i(t) \Sigma_{\hat{i}j}(t) + \Sigma_{\hat{i}j}(t) F_i^T(t) + \bar{Q}_i(t) \\ & - K_i(t) R_i(t) K_i^T(t)] \quad \forall \hat{i}, j, t \end{aligned} \quad (64.26)$$

$$K_i(t) = \Sigma_{\hat{i}j}(t) H_i^T(t) R_i^{-1}(t) \quad \forall \hat{i}, j, t \quad (64.27)$$

$$\mathcal{G}(\Lambda_{\hat{i}j}(t), \hat{\Lambda}_{h\ell u, u+1}) - b_i \bar{\mathcal{M}} \leq \bar{\phi}_{ijh\ell u, u+1}(t) \mathcal{M} \quad \forall \hat{i}, j, h, \ell, u, t \quad (64.28)$$

$$-\mathcal{G}(\Lambda_{\hat{i}j}(t), \hat{\Lambda}_{h\ell u, u+1}) - b_i \bar{\mathcal{M}} < (1 - \bar{\phi}_{ijh\ell u, u+1}(t)) \mathcal{M} \quad \forall \hat{i}, j, h, \ell, u, t \quad (64.29)$$

$$\bar{\phi}_{ijh\ell}(t) = \sum_{u=1}^4 \bar{\phi}_{ijh\ell u, u+1}(t) \quad \forall \hat{i}, j, h, \ell, t \quad (64.30)$$

$$\beta_{ijh\ell}(t) - b_i \bar{\mathcal{M}} \leq 4 - \bar{\phi}_{ijh\ell}(t) \quad \forall \hat{i}, j, h, \ell, t \quad (64.31)$$

$$\beta_{ijh\ell}(t) \mathcal{M} + b_i \bar{\mathcal{M}} \geq 4 - \bar{\phi}_{ijh\ell}(t) \quad \forall \hat{i}, j, h, \ell, t \quad (64.32)$$

$$\bar{\beta}_{ijh}(t) = \sum_{\ell=1}^5 \beta_{ijh\ell}(t) \quad \forall \hat{i}, j, h, t \quad (64.33)$$

$$\bar{\psi}_{ijh}(t) - b_i \bar{\mathcal{M}} \leq \beta_{ijh\ell}(t) \quad \forall \hat{i}, j, h, t \quad (64.34)$$

$$1 - \bar{\psi}_{ijh}(t) - b_i \bar{\mathcal{M}} \leq 5 - \bar{\beta}_{ijh}(t) \quad \forall \hat{i}, j, h, t \quad (64.35)$$

$$\hat{\psi}_{\hat{i}j}(t) = \sum_{h=1}^{\mathcal{H}} \bar{\psi}_{ijh}(t) \quad \forall \hat{i}, j, t \quad (64.36)$$

$$\psi_{\hat{i}j}(t) - b_i \bar{\mathcal{M}} \leq \bar{\psi}_{ijh}(t) \quad \forall \hat{i}, j, t \quad (64.37)$$

$$1 - \psi_{\hat{i}j}(t) - b_{\hat{i}}\bar{\mathcal{M}} \leq \mathcal{H} - \hat{\psi}_{\hat{i}j}(t) \quad \forall \hat{i}, j, t \quad (64.38)$$

$$P_{\hat{i}}(t) \in \mathbb{R}^3 \quad \forall \hat{i}, t \quad (64.39)$$

$$S_{\hat{i}}(t) \in \mathbb{R}_+ \quad \forall \hat{i}, t \quad (64.40)$$

$$\theta_{\hat{i}}(t) \in [-\pi, \pi] \quad \forall \hat{i}, t \quad (64.41)$$

$$\bar{\phi}_{\hat{i}h\ell}(\vec{Y}, t), \hat{\psi}_{\hat{i}}(\vec{Y}, t), \bar{\beta}_{\hat{i}h}(\vec{Y}, t) \in \mathbb{Z}_+ \quad \forall \hat{i}, \vec{Y}, h, \ell, t \quad (64.42)$$

$$\bar{\phi}_{\hat{i}j h\ell}(t), \hat{\psi}_{\hat{i}j}(t), \bar{\beta}_{\hat{i}j h}(t) \in \mathbb{Z}_+ \quad \forall \hat{i}, j, h, \ell, t \quad (64.43)$$

$$\bar{\phi}_{\hat{i}h\ell u, u+1}(\vec{Y}, t), \psi_{\hat{i}}(\vec{Y}, t), \psi_o(\vec{Y}, t), \beta_{\hat{i}h\ell}(\vec{Y}, t) \in \{0, 1\} \quad \forall \hat{i}, \vec{Y}, h, \ell, u, t \quad (64.44)$$

$$\bar{\phi}_{\hat{i}j h\ell u, u+1}(t), \psi_{ij}(t), \beta_{ij h\ell}(t), \bar{\psi}_{ij h}(t) \in \{0, 1\} \quad \forall \hat{i}, j, h, \ell, u, t \quad (64.45)$$

$$\mu_{ihl}(t) \in \{0, 1\} \quad \forall \hat{i}, j, h, \ell, t \quad (64.46)$$

$$b_{\hat{i}} \in \{0, 1\} \quad \forall \hat{i} \quad (64.47)$$

Interpretation of Constraints

Constraints (64.1) prescribe initial positions for each of the UAVs in $\mathcal{N}_i(t)$;

Constraints (64.2) and (64.3) define the motion of UAV \hat{i} , as a function of the speed and heading of the UAV;

Constraints (64.4) restrict the instantaneous change in heading of UAV \hat{i} to be less than $\Omega_{\hat{i}}$ (modeling the UAVs as Dubin's vehicles);

Constraints (64.5) define minimum and maximum speed restrictions UAV \hat{i} ;

Constraints (64.6) enforce no two UAVs being closer than ε_1^{\min} ;

Constraints (64.7) and (64.8) prescribe that no UAV trajectory can intersect with the boundary of any building;

Constraints (64.9)–(64.11) determine the updated search uncertainty map for UAV i at time t ;

Constraints (64.12)–(64.22) are only enforced if UAV \hat{i} is assigned to a search task over the planning horizon $[t_c, t_c + \delta_t]$:

Constraints (64.12) and (64.13) force $\bar{\phi}_{\hat{i}h\ell u, u+1}(\vec{Y}, t)$ to be 1 when line segment $\Lambda_{\hat{i}}(\vec{Y}, t)$ is oriented counterclockwise with respect to the line segment $\hat{\Lambda}_{h\ell u, u+1}$, at time t , and to be 0 otherwise;

Constraints (64.14) count the number of line segments representing the ℓ -th face of building h that line segment $\Lambda_{\hat{i}}(\vec{Y}, t)$ is oriented counterclockwise with respect to, at time t ;

Constraints (64.15) and (64.16) force $\beta_{\hat{i}h\ell}(\vec{Y}, t)$ to be 1 when line segment $\Lambda_{\hat{i}}(\vec{Y}, t)$ does not intersect the ℓ -th face of building h , at time t , and to be 0 otherwise;

Constraints (64.17) count the number of faces of building h that do not inhibit line of sight from UAV \hat{i} to position \vec{Y} , at time t ;

Constraints (64.18) and (64.19) force $\bar{\psi}_{\hat{i}h}(\vec{Y}, t)$ to be 1 when there is line of sight from UAV \hat{i} to position \vec{Y} , at time t , with respect to building h , and to be 0 otherwise;

Constraints (64.20) count the number of buildings that do not inhibit line of sight from UAV \hat{i} to position \vec{Y} , at time t ;

Constraints (64.21) and (64.22) force $\psi_{\hat{i}}(t)$ to be 1 when UAV \hat{i} has line of sight to position \bar{Y} , at time t , and to be 0 otherwise;

Constraints (64.23) and (64.24) determine the fused covariance and state for target j at time t , from UAV i 's perspective, based on those neighbors of UAV i that have line of sight to target j ;

Constraints (64.25)–(64.27) are the continuous time Kalman filter (Kalman-Bucy filter) equations for the state of target j , as seen from UAV \hat{i} , where $\bar{z}_{ij}(t)$ represents the measurement of target j by UAV \hat{i} , at time t (N.B.: a detailed explanation of the Kalman filter equations can be found in Blackman and Popoli (1999), Gelb (1974), and Maybeck (1994));

Constraints (64.28)–(64.38) are only enforced if UAV \hat{i} is assigned to a track task over the planning horizon $[t_c, t_c + \delta_t]$:

Constraints (64.28) and (64.29) force $\bar{\phi}_{ijhlu,u+1}(t)$ to be 1 when line segment $\Lambda_{\hat{i}j}(t)$ is oriented counterclockwise with respect to the line segment $\hat{\Lambda}_{hlu,u+1}$, at time t , and to be 0 otherwise;

Constraints (64.30) count the number of line segments representing the ℓ -th face of building h that line segment $\Lambda_{\hat{i}j}(t)$ is oriented counterclockwise with respect to, at time t ;

Constraints (64.31) and (64.32) force $\beta_{ijht}(t)$ to be 1 when line segment $\Lambda_{\hat{i}j}(t)$ does not intersect the ℓ -th face of building h , at time t , and to be 0 otherwise;

Constraints (64.33) count the number of faces of building h that do not inhibit line of sight from UAV \hat{i} to target j , at time t ;

Constraints (64.34) and (64.35) force $\bar{\psi}_{ijh}(t)$ to be 1 when there is line of sight from UAV \hat{i} to target j , at time t , with respect to building h , and to be 0 otherwise;

Constraints (64.36) count the number of buildings that do not inhibit line of sight from UAV \hat{i} to target j , at time t ;

Constraints (64.37) and (64.38) force $\psi_{ij}(t)$ to be 1 when UAV \hat{i} has line of sight to target j , at time t , and to be 0 otherwise;

Constraints (64.23)–(64.27) prescribe domain restrictions on all of the decision variables.

64.2.5 Objective Function (for UAV i)

The objective function, F , as defined in Eq. (64.48), minimizes the sum of two components. The first component computes the expectant volume of UAV i 's search uncertainty map over the planning horizon, $[t_c, t_c + \delta_t]$. The second component computes the weighted sum of the determinants of the fused covariances for the targets that are known to UAV i , over the planning horizon.

$$F = \min \left[a_s \int_{t_c}^{t_c + \Delta_t} \int_{\mathbb{R}^2} \Gamma_i(\mathcal{Y}, \tau) d\mathcal{Y} d\tau + a_t \int_{t_c}^{t_c + \Delta_t} \sum_{j=1}^{\mathcal{J}} \left[\gamma(X_j^i(\tau)) \left| \Sigma_j^i(\tau) \right| d\tau \right] \right] \quad (64.48)$$

64.2.6 Discussion on Complexity

It is easy to see that our problem of determining the flight plan for a set of autonomous UAVs is a generalization of the standard vehicle routing problem (VRP). While there are many variants to the standard VRP, the general form considers m vehicles, starting at a depot, tasked with delivering goods or services to a set of n customers. Each customer has a certain required demand, and each vehicle has a capacity. In addition, each vehicle must start and end their route at the depot (a tour).

A solution to this problem is a set of tours for which each customer is visited only once while respecting capacity and demand constraints. The objective can be to minimize the total distance traveled by all vehicles, to minimize the maximum distance traveled by any one vehicle, or to minimize the number of vehicles required to service the customers. The VRP is known to be NP-hard (Garey and Johnson 1979), and thus our problem is NP-hard as well.

64.3 Heuristic Algorithm

There are many papers dealing with UAV routing (Shetty et al. 2008; Ma and Miller 2005; Schouwenaars et al. 2001; Sinha et al. 2005) that formulate the problem statement as a linear or mixed-integer linear program (or a formulation that can be linearized in the standard way). This allows the authors to use various proven heuristics to find good routes for the UAVs. However, for UAV routing, incorporating more realistic constraints on the movement of the UAVs necessitates the introduction of nonlinear constraints that cannot be linearized without approximation of the nonlinear constraints. Hence, the need to utilize more generalized heuristics (or meta-heuristics) in the search for near-optimal solutions.

To efficiently solve the optimization problem presented in Sects. 64.2.2–64.2.5, a hybrid heuristic was developed, combining Greedy Randomized Adaptive Search Procedures (GRASP) and Simulated Annealing (SA). GRASP is a multi-start local search procedure, where each iteration consists of two phases, a construction phase and a local search phase (Feo and Resende 1989, 1995; Festa and Resende 2002). In the construction phase, interactions between greediness and randomization generate a diverse set of quality solutions. The local search phase improves upon the solutions found in the construction phase. In this GRASP implementation, only the construction portion is utilized. Simulated Annealing is a heuristic to find good-quality solutions to optimization problems by approximating the cooling process of metals (Kirkpatrick et al. 1983). At each step, a current solution is perturbed. If the perturbation results in a better solution, then the current solution is replaced. If the perturbed solution is worse than the current solution, it still might replace the current solution; replacement will occur with a probability based on the distance between the current and perturbed solution values and the current temperature in the annealing process. As the heuristic progresses, the temperature is lowered, making it more and more unlikely to replace the current solution with a worse perturbed solution.

For our hybrid GRASP-SA heuristic, the GRASP heuristic determines the task assignment for the UAVs, interacting directly with a Simulated Annealing heuristic which determines the routes that the UAVs should fly to accomplish their tasks. The best solution over all of the GRASP multi-start iterations is retained as the final solution. It is important to note that a solution, for a given UAV, is the tasking and flight path for the UAV, as well as the neighbors of the UAV, over the planning horizon. Figure 64.2 provides pseudocode for the GRASP-SA hybrid heuristic, with the function Ξ representing the SA heuristic function.

64.4 Experimental Results

64.4.1 Test Environment

The experiments to follow were carried out on a Hewlett-Packard with an Intel(R) Core(TM) 2 Duo E8400 CPU at 3.00 GHz and 3.71 GB of RAM running Microsoft Windows 7 Enterprise (64-bit). The implementation of the GRASP and Simulated Annealing heuristics as written in the C++ programming language and compiled with Microsoft Visual C++ 2008. The algorithm used for random-number generation is an implementation of the Mersenne Twister algorithm described in Matsumoto and Nishimura (1998).

64.4.2 Experiments

While no specific UAV was modeled, the experiments utilized characteristics of the Raven UAV. The simulations in this study model the UAV sensors as omnidirectional radars measuring range and bearing to the targets. Conversions of the measurement and measurement uncertainty from spherical to cartesian coordinates were done utilizing the standard transformations (Lerro and Bar-Shalom 1993). Table 64.1 displays the parameter values used during the simulation. Thirty-two buildings were modeled in the simulation, each with a height of 100 m, while the altitude of the 4 UAVs was fixed to 10 m. Eight moving ground targets were simulated, with minimum and maximum speeds of 0 and 30 miles h^{-1} , respectively. Three different experiments were performed, varying the communication range of the UAVs. In the “no-comms” case, each UAV had an effective communication radius of 0 m, making them truly independent of all the other UAVs. For the “half-comms” case, each UAVs communication radius was set to 979.5 m, half the length of the simulated region of interest. And in the “unlimited comms” case, each UAV had a communication radius larger than the simulated region of interest, meaning as long as there were no obstructions (e.g., buildings) between two UAVs, they could effectively communicate.

To highlight a small sample of the simulation, Figs. 64.3–64.5 show the results for time steps 44–46 of the unlimited communications case. This is a top-down view,


```

procedure GRASP-SA Heuristic (Problem Instance)
1  UAVsAssigned  $\leftarrow \emptyset$ ;
2  UAVAssignment  $\leftarrow \emptyset$ ;
3  UAVRoutes  $\leftarrow \emptyset$ ;
4   $\alpha \leftarrow \text{UnifRand}[0,1]$ ;
5  while |UAVsAssigned| < NumUAVs do
6      min  $\leftarrow +\infty$ ;
7      max  $\leftarrow -\infty$ ;
8      for  $\hat{i} = 1, \dots, \text{NumUAVs}$  do
9          if  $\hat{i} \notin \text{UAVsAssigned}$  then
10              $[g_{\hat{i}}, z_{\hat{i}}] \leftarrow \Xi(\hat{i}, \text{UAVAssignment}, \text{UAVRoutes}, \text{Search})$ ;
11             if min >  $g_{\hat{i}}$  then min  $\leftarrow g_{\hat{i}}$ ;
12             if max <  $g_{\hat{i}}$  then max  $\leftarrow g_{\hat{i}}$ ;
13              $[\bar{g}_{\hat{i}}, \bar{z}_{\hat{i}}] \leftarrow \Xi(\hat{i}, \text{UAVAssignment}, \text{UAVRoutes}, \text{Track})$ ;
14             if min >  $\bar{g}_{\hat{i}}$  then min  $\leftarrow \bar{g}_{\hat{i}}$ ;
15             if max <  $\bar{g}_{\hat{i}}$  then max  $\leftarrow \bar{g}_{\hat{i}}$ ;
16         end if
17     end for
18     RCL  $\leftarrow \emptyset$ ;
19     Threshold  $\leftarrow \text{min} + \alpha * (\text{max} - \text{min})$ ;
20     for  $\hat{i} = 1, \dots, \text{NumUAVs}$  do
21         if  $\hat{i} \notin \text{UAVsAssigned}$  then
22             if  $g_{\hat{i}} \leq \text{Threshold}$  then
23                 RCL  $\leftarrow \text{RCL} \cup \{[\hat{i}, z_{\hat{i}}, \text{Search}]\}$ ;
24             if  $\bar{g}_{\hat{i}} \leq \text{Threshold}$  then
25                 RCL  $\leftarrow \text{RCL} \cup \{[\hat{i}, \bar{z}_{\hat{i}}, \text{Track}]\}$ ;
26         end if
27     end for
28      $j \leftarrow \text{RandomlySelectElement}(\text{RCL})$ ;
29     UAVsAssigned  $\leftarrow \text{UAVsAssigned} \cup \{j_1\}$ ;
30     UAVAssignment  $\leftarrow \text{UAVAssignment} \cup \{j_3\}$ ;
31     UAVRoutes  $\leftarrow \text{UAVRoutes} \cup \{j_2\}$ ;
32 return(UAVAssignment, UAVRoutes);
end GRASP-SA Heuristic;

```

Fig. 64.2 Pseudo-code for the hybrid GRASP-SA heuristic

Table 64.1 Parameter settings

| Name | Value |
|---------------------|-----------------|
| Δ_t | 10 |
| T_s | 0 |
| T_e | 120 |
| q | 7 |
| N | 4 |
| Ω_i | $\frac{\pi}{6}$ |
| S_i^{\min} | 30 |
| S_i^{\max} | 170 |
| ϵ_1^{\min} | 15 |
| \mathcal{H} | 32 |
| z | 10 |
| \tilde{d}_i | 0.9 |

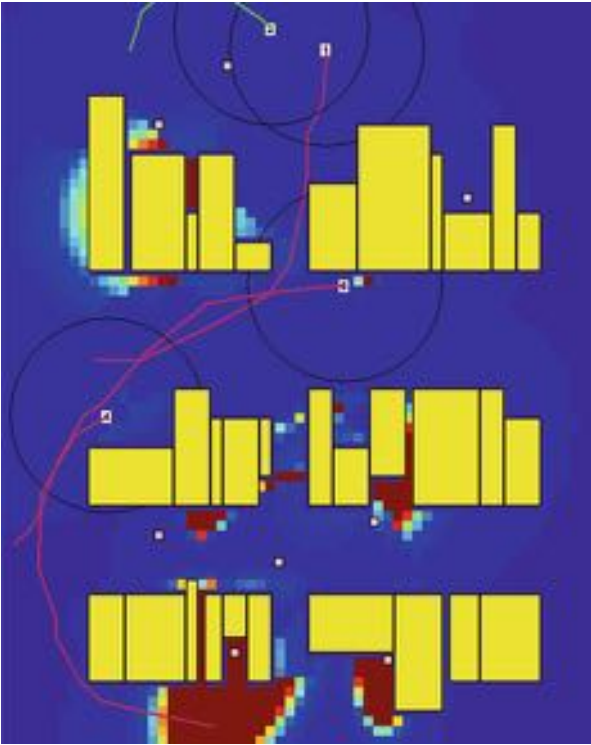


Fig. 64.3 Simulation results for unlimited communications case, at time-step 44. UAVs 1, 3, and 4 are engaged in the search task, while UAV 2 is tracking the target within its sensor coverage (*black circle around UAV*)

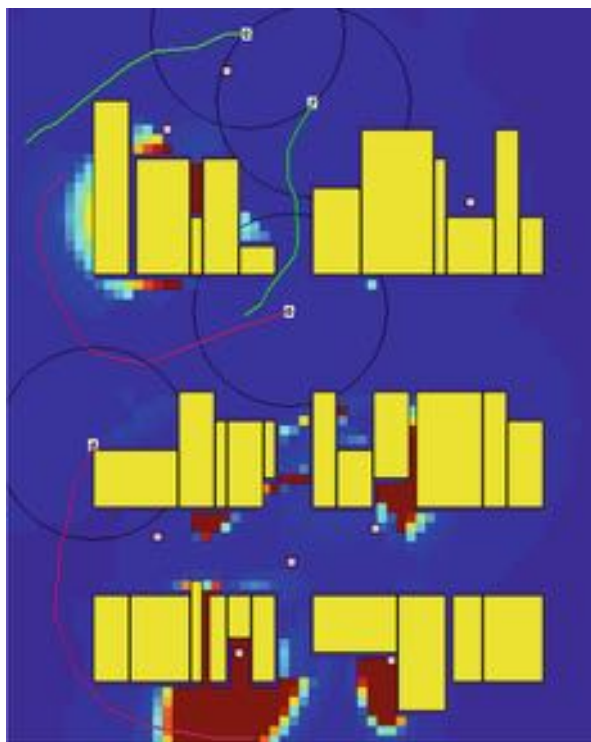


Fig. 64.4 Simulation results for unlimited communications case, at time-step 45. UAVs 3 and 4 are engaged in the search task, while UAVs 1 and 2 are tracking the target within their sensor coverage

with the yellow rectangles representing the buildings, the small white rectangles representing the targets, the UAVs numbered, and the planned routes of the UAVs emanating from each UAV. UAVs engaged in a search task have routes colored red, while those engaged in a tracking task have routes colored green. The background represents the search uncertainty map, with blue corresponding to a low level of uncertainty and red a high level of uncertainty. The sensor coverage for each UAV is represented by a black circle about each UAV. At time-step 44, UAVs 1, 3, and 4 are engaged in a search task, while UAV 2 is engaged in tracking the target within its sensor coverage. At this time-step, UAV 1 and 2 are close enough together to share information, so UAV 1 knows about the target UAV 2 is currently tracking. At time-step 45, UAVs 3 and 4 continue searching, while UAV 1 switches to a tracking task, to help track the target UAV 2 has been tracking. Time-step 46 sees UAV 1 with a new trajectory, to better keep the target it is tracking within its sensor coverage. UAV 2 is still engaged in tracking, and UAVs 3 and 4 are still engaged in a search task. Note also that UAV 4 has its route planned to reduce the uncertainty shown in the bottom left region of the scenario.

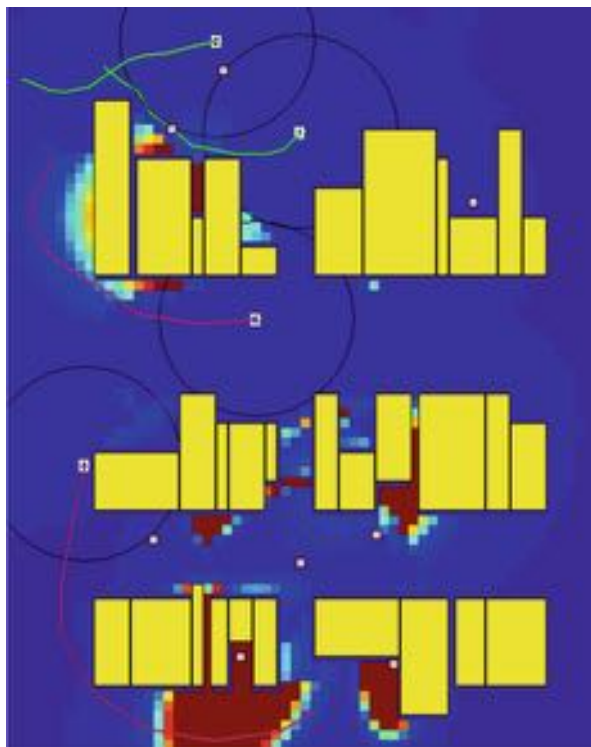


Fig. 64.5 Simulation results for unlimited communications case, at time-step 46. UAVs 3 and 4 are engaged in the search task, while UAV 1 and 2 are continuing to track the target within its sensor coverage. In addition, UAV 1 has changed its trajectory to keep the target within its coverage

One method to measure how the UAV communication range affects the searching algorithm is to observe the minimum search uncertainty value across all UAVs at each location for the simulated region of interest. Figure 64.6 displays the resultant mean minimum search uncertainty map value of the UAVs for the simulated region of interest, at each time-step. The mean search plots tend to show spike and dips. What happens in these cases is that a UAV's search uncertainty map has a group of values that become large in one particular area of the search region. When enough of these values become large, and as long as the UAV is not tasked to track a target, the UAV will eventually plan a route that leads it to search that area of interest. This sudden decrease in many search uncertainty values drives the mean sharply downward. The zero-comms case does not have as many sharp spikes because it has fewer opportunities to switch to tracking (since those UAVs cannot share their state with each other). However, they must conduct their search on their own – this leads to the general upward trend in the zero-comms case in the mean search plot, since they cannot leverage the search information of their neighbors.

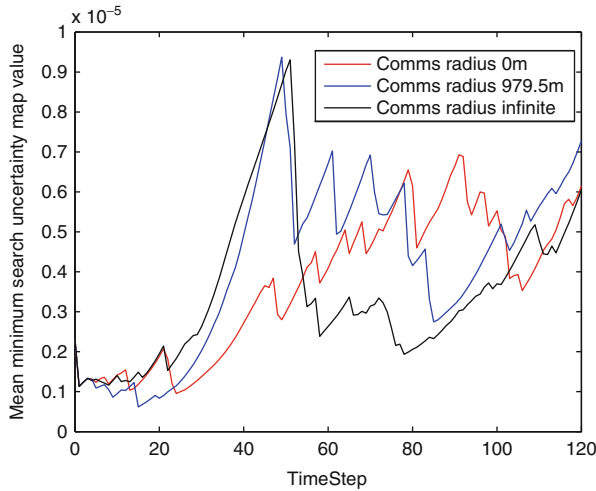


Fig. 64.6 The mean minimum search uncertainty map value over all UAVs at each time-step, for the three UAV communication cases

Figure 64.7 displays the number of UAVs engaged in tracking each target, at each time-step. Of note here is that the mean weighted percentage of targets tracked increases from 48.854 when the communications radius is 0m to 70.208 when the communication radius is 979.5m and finally to 80 when the communication radius is infinite. Figure 64.8 displays the mean Mahalanobis distance (Eq. 64.49), between the fused and truthful positions, for each target, for each time-step. As can be seen here, the mean Mahalanobis distance decreases as the communication radius increases. This is because as communication among the UAVs increases, more information on the target kinematics can be shared among the UAVs (and hence fused into each UAVs target map):

$$M_d(x, \bar{x}, \Sigma, \bar{\Sigma}) = (x - \bar{x}) (\Sigma + \bar{\Sigma})^{-1} (x - \bar{x})^T \quad (64.49)$$

As can be seen in the result plots, a push-and-pull effect is occurring. During the search phase, the UAVs want to fly to areas of high uncertainty (and hence reduce the uncertainty there), but as the UAVs detect targets in these high uncertainty regions, they switch their tasking to tracking (because the initial uncertainty of a target contributes more to the objective function than does the search uncertainty). This can be partially alleviated by modifying the decision variable b_i to include a time dimension, $b_i(t)$. This would have the effect of allowing each UAV to switch between tasks over the planning horizon, rather than forcing each UAV to make a decision on a task for the whole planning horizon. This can also be achieved by artificially reducing the size of the planning horizon. Both approaches will increase the complexity and computational throughput and are out of the current scope.

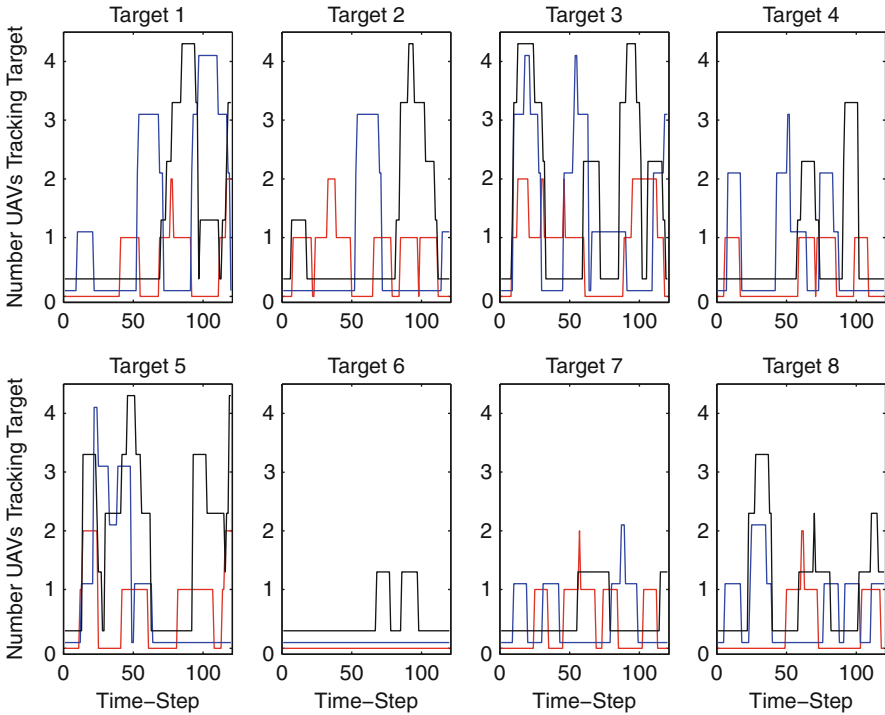


Fig. 64.7 Number of UAVs engaged in tracking each target, at each time-step. The *red* curve denotes the communications radius of 0 m, the *blue* curve denotes the communications radius of 979.5 m, and the *black* curve denotes the infinite communications radius

64.5 Conclusions and Future Research Areas

This chapter considered dynamically updating the tasking and flight paths for UAVs in an urban domain. The UAVs were assumed to be Dubins-like vehicles (i.e., point masses with kinematic constraints) with limited communication capabilities. This resulted in a decentralized approach where each UAV dynamically updated flight plans for itself and its neighbors. As it was an urban domain, the UAVs needed to incorporate obstacle avoidance constraints into the flight plan generation as well as taking into account anticipated future line of sight obstructions to the targets and search region. A dynamic feedback control loop was implemented for each UAV, and a rigorous mathematical formulation was presented for the coupled task assignment/flight path optimization problem each UAV had to solve at each time-step.

This approach was successfully applied to a simulated scenario utilizing a hybrid GRASP-SA heuristic. The influence of communication radius among the UAVs on the resultant objective function value was studied. While it is in general expected

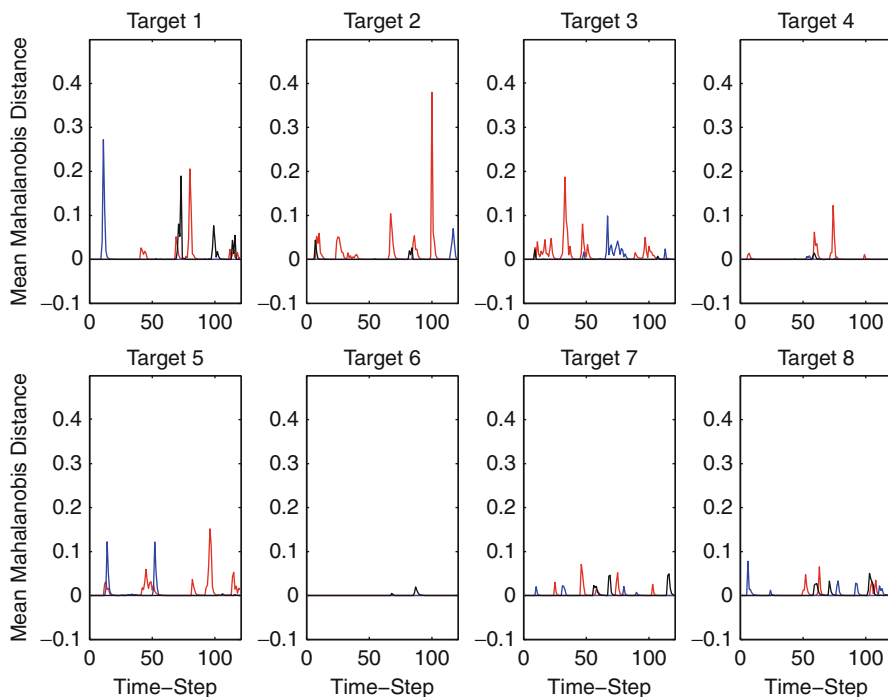


Fig. 64.8 Mean Mahalanobis distance (between the fused position and the truthful position) for each target, for each time-step. The *red curve* denotes the communications radius of 0 m, the *blue curve* denotes the communications radius of 979.5 m, and the *black curve* denotes the infinite communications radius

that the larger the communication radius, the better the performance of the UAVs, as seen from the results, many other factors also influence performance.

It is expected that UAV utilization will continue to increase in scope, both for military and nonmilitary operations. Hence, there is the need to continue to advance the state of the art in autonomous UAV research. Future research directions include applying our general approach to UAVs with heterogeneous sensors (e.g., directional radars, passive sensors) as well as to heterogeneous vehicles, from different domains (e.g., UAVs and UGVs working synergistically). A comparison of the efficiency of our approach, from a computational and functional perspective, to other centralized and decentralized approaches is currently underway. The end goal is geared toward testing these algorithms on real UAVs.

References

- A. Ahmadzadeh, G. Buchman, P. Cheng, A. Jadbabaie, J. Keller, V. Kumar, G. Pappas, Cooperative control of UAVs for search and coverage, in *Proceedings of the AUVSI Conference on Unmanned Systems*, (Orlando, Florida, 2006)

- D. Akselrod, A. Sinha, T. Kirubarajan, Collaborative distributed sensor management to multitarget tracking using hierarchical Markov decision processes, in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, San Diego, vol. 6699, ed. by O.E. Drummond, R.D. Teichgraeber (SPIE, Bellingham, 2007), pp. 1–14
- R.J. Bamburger, D.P. Watson, D.H. Scheidt, K.L. Moore, Flight demonstrations of unmanned aerial vehicle swarming concepts. Johns Hopkins APL Tech. Dig. **27**(1), 41–55 (2006)
- A. Bicchi, G. Casalino, C. Santilli, Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, 1995, pp. 1349–1354
- S. Blackman, R. Popoli, *Design and Analysis of Modern Tracking Systems* (Artech House, Boston, 1999)
- E.A. Doucette, A.J. Sinclair, D.E. Jeffcoat, Simultaneous localization and planning for cooperative air munitions via dynamic programming, in *Optimization and Cooperative Control Strategies*, ed. by M.J. Hirsch, C.W. Commander, P.M. Pardalos, R. Murphey (Springer, Berlin/London, 2009), pp. 69–79
- L.E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. Am. J. Math. **79**(3), 497–516 (1957)
- T.A. Feo, M.G.C. Resende, A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. **8**, 67–71 (1989)
- T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures. J. Global Optim. **6**, 109–133 (1995)
- P. Festa, M.G.C. Resende, GRASP: an annotated bibliography, in *Essays and Surveys in Meta-heuristics*, ed. by C.C. Ribeiro, P. Hansen (Kluwer, Boston, 2002), pp. 325–367
- M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W.H. Freeman, San Francisco, 1979)
- A. Gelb, *Applied Optimal Estimation* (MIT, Cambridge, 1974)
- G. Gu, P.R. Chandler, C.J. Schumacher, A. Sparks, M. Pachter, Optimal cooperative sensing using a team of UAVs. IEEE Trans. Aerosp. Electron. Syst. **42**(4), 1446–1458 (2006)
- M.J. Hirsch, H. Ortiz-Pena, Autonomous network connectivity and cooperative control for multiple target tracking, in *Proceedings of the 27th Army Science Conference*, Orlando, 2010
- M.J. Hirsch, H. Ortiz-Pena, N. Sapankevych, R. Neese, Efficient flight formation for tracking of a ground target, in *Proceedings of the National Fire Control Symposium*, San Diego, 2007, pp. 1–16
- M.J. Hirsch, H. Ortiz-Pena, M. Sudit, Decentralized cooperative urban tracking of multiple ground targets by a team of autonomous UAVs, in *Proceedings of the 14th International Conference on Information Fusion*, Chicago, 2011, pp. 1196–1202
- M.J. Hirsch, H. Ortiz-Pena, C. Eck, Cooperative tracking of multiple targets by a team of autonomous UAVs. Int. J. Oper. Res. Inf. Syst. **3**(1), 53–73 (2012)
- ILOG CPLEX, <http://www.ilog.com/products/cplex>. Accessed Oct 2008
- K. Jiang, L.D. Seneviratne, S.W.E. Earles, A shortest path based path planning algorithm for nonholonomic mobile robots. J. Intell. Rob. Syst. **24**, 347–366 (1999)
- R.J. Kenefic, Finding good Dubins tours for UAVs using particle swarm optimization. J. Aerosp. Comput. Inf. Commun. **5**, 47–56 (2008)
- J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, Australia, 1995, pp. 1942–1948
- D. Kingston, R. Beard, UAV splay state configuration for moving targets in wind, in *Advances in Cooperative Control and Optimization*, ed. by M.J. Hirsch, P.M. Pardalos, R. Murphey, D. Grundel (Springer, Berlin, 2007), pp. 109–129
- S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
- S.M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, 2006)
- D. Lerro, Y. Bar-Shalom, Tracking with debiased consistent converted measurements versus EKF. IEEE Trans. Aerosp. Electron. Syst. **29**(3), 1015–1022 (1993)

- Z. Liang, W.A. Chaovalitwongse, A.D. Rodriguez, D.E. Jeffcoat, D.A. Grundel, J.K. O'Neal, Optimization of spatiotemporal clustering for target tracking from multi-sensor data. *IEEE Trans. Syst. Man Cybern. (C)* **40**(2), 176–188 (2010)
- C.S. Ma, R.H. Miller, Mixed integer linear programming trajectory generation for autonomous nap-of-the-earth flight in a threat environment, in *Proceedings of the IEEE Aerospace Conference*, Big Sky, 2005, pp. 1–9
- M. Matsumoto, T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **8**(1), 3–30 (1998)
- P.S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. I (Navtech Press, Arlington, 1994)
- T.W. McLain, P.R. Chandler, M. Pachter, A decomposition strategy for optimal coordination of unmanned air vehicles, in *Proceedings of American Control Conference*, Chicago, 2000, pp. 369–373
- F. Rafi, S. Khan, K. Shafiq, M. Shah, Autonomous target following by unmanned aerial vehicles, in *Proceedings of SPIE: Unmanned Systems Technology VIII*, Orlando, vol. 6230, ed. by G.R. Gerhart, C.M. Shoemaker, D.W. Gage (SPIE, Bellingham, 2006), pp. 1–8
- C. Santilli, A. Bicchi, G. Casalino, A. Balestrino, Nonholonomic, bounded curvature path planning in cluttered environments, in *Proceedings of the Conference on Emerging Technologies and Factory Automation*, Paris, vol. 2, 1995, pp. 363–372
- T. Schouwenaars, B. De Moor, E. Feron, J. How, Mixed-integer programming for multi-vehicle path planning, in *Proceedings of the European Control Conference*, Porto, 2001, pp. 2603–2608
- C. Schumacher, T. Shima, Single-task tours, in *UAV Cooperative Decision and Control*, ed. by T. Shima, S. Rasmussen (SIAM, Philadelphia, 2009), pp. 15–36
- J.G. Semple, G.T. Kneebone, *Algebraic Projective Geometry* (Oxford University Press, New York, 1952)
- V.K. Shetty, M. Sudit, R. Nagi, Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. *Comput. Oper. Res.* **35**(6), 1813–1828 (2008)
- T. Shima, S. Rasmussen, *UAV Cooperative Decision and Control* (SIAM, Philadelphia, 2009)
- A.J. Sinclair, R.J. Prazenica, D.E. Jeffcoat, Simultaneous localization and planning for cooperative air munitions, in *Advances in Cooperative Control and Optimization*, ed. by M.J. Hirsch, P.M. Pardalos, R. Murphey, D. Grundel (Springer, Berlin, 2007), pp. 81–94
- A. Sinha, T. Kirubarajan, Y. Bar-Shalom, Optimal cooperative placement of UAVs for ground target tracking with doppler radar, in *Proceedings of SPIE: Signal processing, Sensor Fusion, and Target Recognition XIII*, Orlando, vol. 5429, ed. by I. Kadar (SPIE, Bellingham, 2004), pp. 95–104
- A. Sinha, T. Kirubarajan, Y. Bar-Shalom, Autonomous ground target tracking by multiple cooperative UAVs. *Proceedings of the IEEE Aerospace Conference*, Big Sky, 2005, pp. 1–9
- M. Steinberg, Intelligent autonomy for unmanned naval vehicles, in *Proceedings of SPIE: Unmanned Systems Technology VIII*, Kissimmee, vol. 230, ed. by G.R. Gerhart, C.M. Shoemaker, D.W. Gage (SPIE, Bellingham, 2006), pp. 1–12
- J. Stolfi, *Oriented Projective Geometry: A Framework for Geometric Computations* (Elsevier, Palo Alto, California, 1991)
- C.A. Sylvester, G.J. Wiens, N.G. Fitz-Coy, Control of collaborative mobile robots subject to nonholonomic constraints, in *Proceedings of the International Symposium on Collaborative Technologies and Systems*, San Diego, 2004

Sertac Karaman, Emre Koyuncu, and Gokhan Inalhan

Contents

| | | |
|------|--|------|
| 65.1 | Introduction | 1602 |
| 65.2 | Task Assignment Problem | 1605 |
| 65.3 | Algorithmic Implementation for the Shortest Fleet-Path Problem | 1609 |
| 65.4 | Computational Experiments | 1612 |
| 65.5 | Conclusion | 1615 |
| | References | 1616 |

Abstract

Decades of advances in electronics and communication technologies have enabled large-scale, flexible, reconfigurable multiple-UAV systems with a variety of applications ranging from large-scale military surveillance and reconnaissance to environmental monitoring and disaster response. Most of these applications require multiple tasks and service requests to be handled by a large group of UAVs in an efficient manner. Motivated by these problems, this chapter is devoted to a large-scale distributed task/target assignment problem for a fleet of autonomous UAVs. Central to this chapter is an algorithm that uses the delayed column generation approach on a non-convex supply-demand formulation for the problem. The algorithm exploits a computationally tractable distributed

S. Karaman (✉)

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

e-mail: sertac@MIT.EDU

E. Koyuncu

Controls and Avionics Laboratory, Istanbul Technical University, Istanbul, Turkey

e-mail: emre.koyuncu@itu.edu.tr

G. Inalhan

Faculty of Aeronautics and Astronautics, Istanbul Technical University, Istanbul, Turkey

e-mail: inalhan@itu.edu.tr

coordination structure, i.e., a market for tasks and targets created by the UAV fleet. The resulting structure is solved via a fleet-optimal dual simplex ascent in which each UAV updates its respective flight plan costs with a linear update of waypoint task values as evaluated by the market. Synchronized and asynchronous distributed implementations of this approximation algorithm is demonstrated in a thorough experimental study involving dynamically changing scenarios with random pop-up targets. Results from the tests performed on an in-house built network mission simulator are also provided for the numerical verification of the algorithm on (a) bounded polynomial-time computational complexity and (b) hard real-time performance for problem sizes on the order of 100 waypoints per UAV.

65.1 Introduction

In the last three decades, the advances in electronics and communications technology provided faster and cheaper computational power which can be seamlessly networked over wide geographies. These advances resulted in a new paradigm in large-scale dynamic systems (Lasdon 1970). In this paradigm, groups of independent dynamic systems, such as unmanned air vehicles, act as a cooperative unit for a diverse set of applications in remote sensing, exploration, and imaging. These systems have been envisioned to provide highly flexible and reconfigurable structures that use individual autonomy to respond to changing environments and operations.

Of particular importance to military applications is the replacement of traditional complex aircraft fighter systems with an array of simpler and lower-cost unmanned air vehicles (UAVs). Figure 65.1 illustrates a dynamic conflict scenario envisioned by the Defense Advanced Research Project Agency (DARPA). In this scenario, unmanned air vehicle fleets are a part of a global coordination network that includes manned aircraft and satellites used together for information gathering, data relay, and target elimination. This network is controlled by a series of distributed logistics centers that identify geographic locations of interest and generate high-level assignments. These locations are transmitted to each vehicle in the airborne platform as waypoints which are described via a series of geographic positions and mission times. Once this information is received, the aim of each unmanned air vehicle is to handle the lower-level operations which include flying through its regions of interest, performing the necessary tasks such as imaging or data collection, and identifying new targets either autonomously or through the logistics centers to which these images and the data are transmitted back to in real time. The efficient operation of this airborne platform relies not only on the each vehicle's ability to autonomously fly through its designated way-points but also on each vehicle's ability to negotiate or coordinate the assignment of newly identified or transmitted targets. With the ever-growing involvement of UAVs in complex

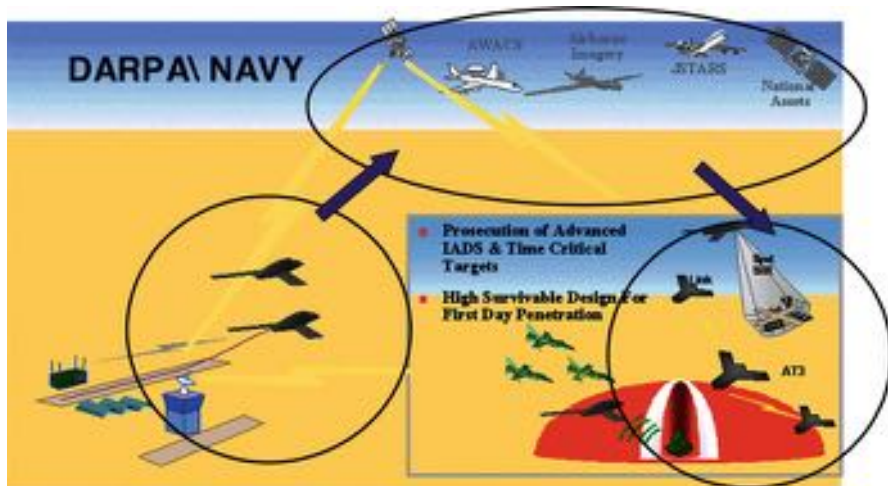


Fig. 65.1 Reconfigurable coordination network envisioned by DARPA (2012). In this network, unmanned air vehicle fleets improve flexibility and offer a high degree of redundancy and reconfigurability in the event of a vehicle failure

application areas (such as dynamically changing urban rescue operations), the types and the number of tasks easily outgrow one vehicle's (or a set of UAV operators' command and control) limited capabilities and thus require a fleet of UAVs working autonomously in a collaborative fashion to achieve desired operational goals. In remote locations, where a minimum amount of ground support (centralized control) is available, it is crucial that the vehicles coordinate their target and task assignments (Toth and Vigo 2002) in dynamically changing scenarios.

In this chapter, a distributed optimization method that allows collaborative task/target assignment across such autonomous UAV fleets is proposed. Specifically, the method considers the cases in which the number of tasks/targets is very large (i.e., on the order of tens to hundreds per UAV) and dynamically change.

For large-scale assignment problems with limited resources (Melhorn and Ziegelmann 2000; Minoux 1987), the optimal/near-optimal allocation of such resources becomes a crucial deciding factor for the success of a mission. The problem of task assignment in optimization framework is inherently a binary integer linear program (Garfinkel and Nemhauser 1972) (a special case of mixed integer linear programming (MILP) (Brickman 1989; Ralphs and Galati 2006; Richards et al. 2003)) and has been used widely for formulating UAV task assignment problems (Schouwenaars et al. 2001; Bellingham et al. 2003). In this work, a structural decomposition of the most general form of this task assignment formulation through column generation is exploited. Essentially, the column generation approach provides a method for breaking up column-dense large-scale optimization problems into subcomponents that are integrated to a growing master problem

which is optimized one step at a time. Together with Desaulniers et al. (2005) a good survey of column generation in integer programming is presented in Barnhart et al. (1998). The reader is referred to Ehrgott and Tind (2007) and Lubbecke and Desrosiers (2005) for the use of column generation technique in the context of integer programming for many problems outlined in these references.

There has been considerable effort in literature to use column generation to solve shortest-path problems (Irnich and Desaulniers 2005), vehicle routing problems with time windows (Kallehauge and Larsden 2005; Donna and Pape 2005; Chabrier 2006), and split-delivery vehicle routing problem (Desaulniers 2010; Archetti et al. 2011). However, these formulations are aimed at solving the centralized assignment problem in a step-by-step, progressive fashion. In comparison, this chapter outlines a distributed implementation structure that is amenable to onboard real-time computation on the UAVs.

Specifically, Sect. 65.2 provides a primitive non-convex supply-demand form of the task assignment problem with extreme point solutions that conveniently correspond to the waypoint selection for each UAV. This column-dense extreme point formulation is used for employing a delayed column generation optimization scheme which involves (a restricted form of) a master and subproblem formulations. Both of these steps are shown to be implementable at the local computer of any of the UAVs requiring only communication for passing flight path information (i.e., specific waypoint selections) and the costs associated with these flight paths. This two-step solution corresponds to a fleet-optimal dual simplex ascent method in which each UAV updates its respective flight plan costs with a linear update of waypoint task values as evaluated in the market created by the UAVs for the targets.

Market architecture that seeks to maximize benefit while minimizing travel distance costs has been used for solving task allocation problem as well. In Zlot et al. (2002), exploration has been performed in a distributed fashion. Specifically, new tasks that can be generated by each of the robots are inserted into their respective plans or subsequently traded across the group. This approach has been further extended to handle task failures, partial malfunctions, and robot death (Dias et al. 2004). Pongpunwattana et al. (2003) combined evaluation computing-based techniques and the distributed structure of market-based algorithm using Vickrey auction where the sale price is the value of the second-highest bid to motivate truthful bids from the participants. In Lemaire et al. (2004), distributed architecture for multi-robot tasks allocation problem is considered, and the Contract-Net protocol has been enhanced with a token-based system which equilibrates the workload between the agents. By considering both single- and multi-task assignment problem, Choi et al. (2009) presented two decentralized algorithms that utilize a market-based decision strategy and a consensus routine based on local communication as the conflict resolution mechanism to achieve agreement on the winning bid values. This work has been further extended to account for tasks with time windows of validity and heterogeneous agent capabilities and constraints (Ponda et al. 2010). Similarly, in the application context of multi-agent task allocation, Sheng et al. (2004) considered disruptions due to limited-range communication, and the

proposed method incorporated a term into utility function to limit the amount of communication loss. Bhattacharya et al. (2010) also presented a distributed implementation of a separable optimization problem with nonlinear constraints arising from coupling between pairs of robots and showed that the algorithm converges to an optimal solution under certain conditions. In Cheng and Kumar (2008), the decentralized algorithm used a local sensing-based control law and time synchronization to distribute all vehicles at regular intervals along a common curve in finite time without needing any communication between neighboring vehicles. Authors also refer the readers to Dias et al. (2006) in which a comprehensive literature survey and their analysis (up to 2005) are given on market-based approaches to multi-agent coordination. Similarly, in this chapter, Sect. 65.3 is devoted to algorithmic implementation details of the proposed method and Sect. 65.4 to the computational results of the standard fleet-optimal shortest-path target assignment problem.

The main contributions of this chapter can be listed as follows. We provide a novel delayed column generation relaxation on a non-convex supply-demand formulation for target-task assignment problem. The algorithm which solves this relaxation exploits a computationally tractable distributed coordination structure, i.e., a market for tasks and targets created by the UAV fleet. The resulting structure is solved via a fleet-optimal dual simplex ascent in which each UAV updates its respective flight plan costs with a linear update of waypoint task values as evaluated by the market. The solution method provides the flexibility of synchronized and asynchronous distributed implementations across dynamically changing scenarios with random pop-up targets. In comparison to existing solution techniques, the method as proposed provides (i) bounded polynomial-time computational complexity and (ii) hard real-time performance for problem sizes on the order of 100 waypoints per UAV. Algorithmic structure and numerical implementation of the results presented in this chapter have previously appeared in Karaman and Inalhan (2008).

65.2 Task Assignment Problem

Consider the task assignment problem of m vehicles to n targets, defined as follows. Let y_{ij} be a binary decision variable that evaluates to one if the i th vehicle is assigned to service target j and evaluates to zero otherwise. For notational convenience, let y_i denote the vector $y_i := [y_{i1} \ y_{i2} \ \cdots \ y_{in}]$. This vector of decision variables simply denotes the set of all targets that are assigned to the i th vehicle. Let $c_i(\cdot)$ denote the cost function that associates a cost to each such vector y_i . Then, the task assignment problem can be written as follows:

Problem A:

$$\begin{aligned} & \text{minimize} && c_1(y_1) + c_2(y_2) + \cdots + c_m(y_m) && (65.1) \\ & \text{subject to} && I \ y_1 + I \ y_2 + \cdots + I \ y_m = 1_{n \times 1} \end{aligned}$$

$$\begin{aligned}
\sum_j y_{1j} &\geq 1 \\
\sum_j y_{2j} &\geq 1 \\
&\vdots \\
\sum_j y_{mj} &\geq 1 \\
y_{ij} &\in \{0, 1\}
\end{aligned}$$

where I denotes the identity matrix of suitable size and $1_{n \times 1}$ denotes the vector of ones of size n . In this formulation, the first set of constraints ensure that every target is serviced exactly once, and the rest ensure that each vehicle services at least one target.

A common choice for the cost function $c_i(y_i)$ is the length of the shortest path through the targets serviced in the assignment y_i . In that case, the objective function in Eq. (65.1) becomes the “shortest fleet path.” However, throughout this chapter, no specific assumptions are placed on the structure of the cost function. In particular, the cost function can be nonlinear (in subscript j that indexes the targets) as in the shortest-path example.

The LP relaxation of the problem described above is obtained by posing the constraint $0 \leq y_{ij} \leq 1$, where $y_{ij} \in \mathbb{R}$, to replace the constraint $y_{ij} \in \{0, 1\}$. Note that the upper bound on y_{ij} is implied by the interconnecting assignment constraint.

The optimization problem described in Problem A lends itself to a distribution solution through the well-known Dantzig-Wolfe decomposition (Vanderback and Savelsbergh 2006). For each vehicle i , define the corresponding polyhedral region $P_i := \{y_{ij} \in \mathbb{R} : \sum_j y_{ij} \geq 1, 0 \leq y_{ij} \leq 1\}$. Then, using the Minkowski resolution theorem, the assignment vector y_i can be described as a convex combination of the extreme points of P_i . More precisely, let y_i^q , indexed by $q \in Q_i$, denote the extreme points of P_i . Then, each y_i can be represented in terms of a new set $\{x_i^q : q \in Q_i\}$ of variables as follows:

$$y_i = \sum_{q \in Q_i} x_i^q y_i^q$$

subject to the convexity constraints

$$\sum_{q \in Q_i} x_i^q = 1; \quad x_i^q \geq 0.$$

This formulation, in particular, leads to a representation of the objective function that is linear in the new decision variables. More precisely, $c_i(y_i) = c_i(x_i^q y_i^q) = x_i^q c(y_i^q) = x_i^q c_i^q$, where c_i^q is a constant independent of the decision variables x_i^q , for all $q \in Q_i$. Define also the constants $d_i^q := I y_i^q$. This transformation to use the new decision variables leads to a new mathematical problem, which, in the terminology of Dantzig-Wolfe decomposition, is known as the *master problem*.

$$\begin{aligned}
& \text{minimize} && \sum_{q \in Q_1} c_1^q x_1^q + \cdots + \sum_{q \in Q_m} c_m^q x_m^q && (65.2) \\
& \text{subject to} && \sum_{q \in Q_1} d_1^q x_1^q + \cdots + \sum_{q \in Q_m} d_m^q x_m^q = b_0 = 1_{n \times 1} \\
& && \sum_{q \in Q_1} x_1^q = 1 \\
& && \vdots \\
& && \sum_{q \in Q_m} x_m^q = 1 \\
& && x_i^q \geq 0 \quad \forall q \in Q_i, \quad i = 1, \dots, m.
\end{aligned}$$

After introducing some new notation, this mathematical program can be stated in more compact form as follows:

Master Problem:

$$\begin{aligned}
& \text{minimize} && c'_1 x_1 + \cdots + c'_m x_m && (65.3) \\
& \text{subject to} && D_1 x_1 + \cdots + D_m x_m = b_0 \\
& && f'_1 x_1 = 1 \\
& && \vdots \\
& && f'_m x_m = 1 \\
& && x_i^q \geq 0 \quad \forall q \in Q_i, \quad i = 1, \dots, m,
\end{aligned}$$

where $f_i = [1 \cdots 1]'$ has suitable length, and $D_i = [d_i^1 \mid d_i^2 \mid \cdots \mid d_i^q \mid \cdots]$, $q \in Q_i$.

Usually, the number of variables in the master problem is much larger than the number of constraints. In such circumstances, one can consider solving the master problem by keeping only a small fraction of the variables that are collectively adequate for forming the basis for a simplex formulation. This problem with a small number of variables is called the *restricted master problem* (RMP), defined as follows:

Restricted Master Problem:

$$\begin{aligned}
& \text{minimize} && \sum_{q \in \bar{Q}_1} c_1^q x_1^q + \cdots + \sum_{q \in \bar{Q}_m} c_m^q x_m^q && (65.4) \\
& \text{subject to} && \sum_{q \in \bar{Q}_1} d_1^q x_1^q + \cdots + \sum_{q \in \bar{Q}_m} d_m^q x_m^q = b_0 \\
& && \sum_{q \in \bar{Q}_1} x_1^q = 1 \\
& && \vdots \\
& && \sum_{q \in \bar{Q}_m} x_m^q = 1 \\
& && x_i^q \geq 0 \quad \forall q \in Q_i, \quad i = 1, \dots, m,
\end{aligned}$$

where the index set \bar{Q}_i is a subset of Q_i .

Let π_j denote the dual variables associated with the n coupling constraints and μ_i denote the dual variables of associated with the convexity constraints. Thus, define a dual variable, π_j , associated with each target and a dual variable, μ_i , associated with each vehicle. And define

$$B := \begin{bmatrix} \bar{D}_1 & \bar{D}_2 & \cdots & \bar{D}_m \\ [1 \cdots 1] & [0 \cdots 0] & \cdots & [0 \cdots 0] \\ [0 \cdots 0] & [1 \cdots 1] & \cdots & [0 \cdots 0] \\ \vdots & \vdots & \ddots & \vdots \\ [0 \cdots 0] & [0 \cdots 0] & \cdots & [1 \cdots 1] \end{bmatrix}.$$

Then, for the restricted master problem, the primal and the dual solutions are given as

$$x_B = B^{-1} b$$

$$[\pi \mu]' = [\bar{c}'_1 \cdots \bar{c}'_m]' B^{-1}.$$

Moreover, the dual formulation of the problem can be defined uniquely as

Dual of the Restricted Master Problem:

$$\begin{aligned} & \text{minimize} && \sum_j \pi_j + \sum_i \mu_i \\ & \text{subject to} && (d_1^q)' \pi + \mu_1 \leq c_1^q, \quad q \in \bar{Q}_1 \\ & && \vdots \\ & && (d_m^q)' \pi + \mu_1 \leq c_m^q, \quad q \in \bar{Q}_m. \end{aligned}$$

To solve the dual of the RMP, consider finding new columns to enter the basis in the simplex method. The *reduced cost* for each column is calculated as follows:

$$c_i^q - [\pi \mu] \begin{bmatrix} d_1^q \\ 0 \\ \vdots \\ 0 \\ 1_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} = c_i^q - \pi d_1^q - \mu_i.$$

When the pivoting rule is such that the column with the least (negative) reduced cost is added to the basis, the problem of finding a column to enter the basis can be posed as an optimization problem:

$$\min_{q \in Q_i} c_i^q - \pi d_i^q - \mu_i.$$

This optimization problem is also called the *subproblem* (SP). Alternatively, using our original notation, the SP can be stated as follows:

$$\min_{y_i} c_i(y_i) - \pi y_i - \mu_i.$$

Now, consider solving the dual of the RMP to determine the optimal task assignment as posed in Problem A. Proceed using a revised simplex method, choosing the columns that forms the basis by solving the SPs, for each i , as described above. Keep in the restricted master problem only those columns that are chosen for the basis using the SPs. This technique roughly described above is usually called *column generation* (see, e.g., Bertsimas and Tsiklis 1997; Desrosiers and Lubbecke 2005; Vanderback 2005; and Desaulniers et al. 2005).

Note that each SP is a binary integer program, a class of problems that is known to be NP hard, hence considered unlikely to admit efficient algorithms. Yet, the special structures of the SPs allow them to be interpreted as a simple search problem, which leads to a polynomial-time (in fact, a linear-time) algorithm that solves the SPs. In contrast, binary integer programs are usually solved using branch-and-bound methods, which require an exponential running time in the worst case.

Thus far, the column generation technique is described in a rather general setting. The next section focusses on a particular task assignment problem instance and describes a distributed implementation in detail. At this point, let us stress that, in that particular setting, the SPs provide an elegant way to update the flight plan: linear updates maintain the “values” of the waypoints formed in a virtual market created by the UAVs. In fact, the dual variable for a waypoint represents its market value. This market interpretation is discussed later in the text.

65.3 Algorithmic Implementation for the Shortest Fleet-Path Problem

To simplify the presentation, let us concentrate on a particular problem where the objective is to minimize the total path length traveled by all UAVs. Let us examine this problem in two phases.

Phase I: Initialization Prior to the initialization of the algorithm, all UAVs communicate their states, i.e., their position and the location of all the targets. Thus, the state of the world and that of the UAVs is known to all UAVs. In the initialization of the phase, each UAV:

- Calculates the distances between the waypoints, which is encoded in an $n \times n$ matrix denoted by V such that V_{ij} is the distance between waypoints i and j .

In the sequel, it is assumed that the necessary conditions are satisfied so that the matrix V becomes symmetric and all its diagonal elements are zero.

- Computes the distance from every UAV to every point, encoded in an $n \times m$ matrix r such that r_{ij} is the distance from the i th UAV to the j th waypoint. With a slight abuse of notation, denote by r_i the $n \times 1$ vector distances from the i th UAV to all the waypoints.
- Forms its flight plan matrix, denoted by D_i , which encodes all possible combination of way-points that it can travel.
- Finds the cost for each waypoint combination in D_i . This can be done, for instance, by computing the shortest path through all waypoints visited in each waypoint combination, using V_i and r_i . The cost of this path is stored in a matrix denoted by c_i for the i th vehicle.
- Creates the restricted master problem matrix B that contains the columns forming the initial basis. For particular choices of m and n , one can always find such a matrix, which can be stored in memory to simplify computations. See, for example, Bertsimas and Tsiklis (1997) for details.
- Communicates the associated flight cost \bar{c}_i for all columns in B to the other UAVs in the fleet.

Note that this initialization scheme is applicable in a class of problems much larger than that described above. In particular, the cost function need not be the shortest path, but may also involve sophisticated measures of fuel consumption, efficiency, risk, as well as mission completion time. In addition, efficient approximation algorithms with rigorous bounds on optimality can often be employed when computing such quantities.

Phase II: Dual Simplex Ascent The second phase of the algorithm is a distributed implementation of the dual simplex ascent over the extreme points. In each iteration, each UAV selects a cost-minimizing waypoint combination for itself via solving its SP. The column corresponding to this waypoint combination is, then, inserted into the master problem to find out whether this column is better for the whole fleet, in terms of cost, when compared to its previous alternative.

Each UAV is engaged in solving its own SP. These calculations are executed simultaneously on every UAV. The results are synchronized afterwards at the end of each iteration. The synchronization ensures that each UAV updates the solution to the master problem identically. Since each UAV executes exactly the same (deterministic) calculations, there is no need to exchange the results for the solution of the master problem. By design, the master problem contains only a few constraints; hence, its solution is obtained relatively fast. More precisely, it amounts to inverting and multiplying matrices with dimensions $(n + m) \times (n + m)$ only a constant number of times.

The procedure described in this section is summarized in Algorithm 1, where both phases of the algorithm are presented in detail. At this point, let us emphasize that the communication requirement for the synchronization step is small for many practical applications. In fact, each UAV must communicate a waypoint combination it would like to visit along with the cost of the selected waypoint combination.

Algorithm 1: Distributed synchronized implementation for i th UAV

```

// PHASE I: Initialization
1 Compute the distances between the way-points to form the  $n \times n$  matrix  $V$ ;
2 Compute the distance to the way-points to form the  $n \times 1$  vector  $r_i$ ;
3 for each way-point combination that the  $i$ th UAV can execute do
4   Compute the column encoding the way-point combination and add to the path matrix
      $D_i$ ;
5   Compute the cost of this path and add to the cost vector  $c_i$ 
6 end
7 Read the initial  $B$  matrix from the memory;
8 Communicate the cost of the corresponding paths in the  $B$  matrix to start the master
   iteration;
// PHASE II: Dual Simplex Ascent
9 repeat
10  Compute the solution to the restricted master iteration and find the dual variables
     vector:  $[\pi \ \mu]' \leftarrow [\tilde{c}'_1 \ \dots \ \tilde{c}'_m]' B^{-1}$ ;
11  Remove the ineffective way-point combinations, if any, with maximum cost;
12  Compute a solution to the SP corresponding to the  $i$ th UAV by updating flight costs:
      $c_i^q \leftarrow c_i^q - \pi d_1^q - \mu_i$ ;
13  Communicate the new way-point combination and its cost. Insert the minimum-cost
     feasible way-point combination to the restricted master problem.
14 until There is no path with negative sub-problem cost;

```

Arguably, this data can be communicated fast and reliably on most modern wireless communication hardware, including even those with stringent power and computation constraints.

The approach presented above is furnished with many other properties suitable for the application at hand. Firstly, solving the SPs require much smaller amount of memory than that required by integer programming techniques, such as the branch-and-bound algorithm tailored to solve the original optimization problem. This property allows implementation on platforms with limited computational capabilities. Secondly, the solution to the master problem is often a feasible solution which can be implemented immediately. Thus, the algorithm provides a feasible solution right after the initialization phase is completed, and it improves this solution if provided more computation time. This anytime flavor of the algorithm makes the algorithm suitable for real-time implementations. Admittedly, the solution to the master problem is not necessarily an integer solution at all times. Yet, any non-integer solution can easily be rounded, for example, with the help of effective randomization (see, e.g., Raghavan and Thompson 1987). Although this may lead to a suboptimal solution, rigorous bounds on optimality can be obtained in some cases (Bertsimas et al. 1999).

In Figs. 65.2 and 65.3, the algorithm is illustrated on an example scenario involving nine targets. As seen in Fig. 65.2, the aggregate cost of the targets serviced by each UAV varies across iterations, as the UAVs decide on which targets to service in a virtual market. Notice, however, that the total cost is a nonincreasing function of the number of iterations; in fact, the total cost is substantially reduced after

Fig. 65.2 Aggregate cost of visiting all targets (*cyan*) and the costs barred by each UAV across iterations of the algorithm

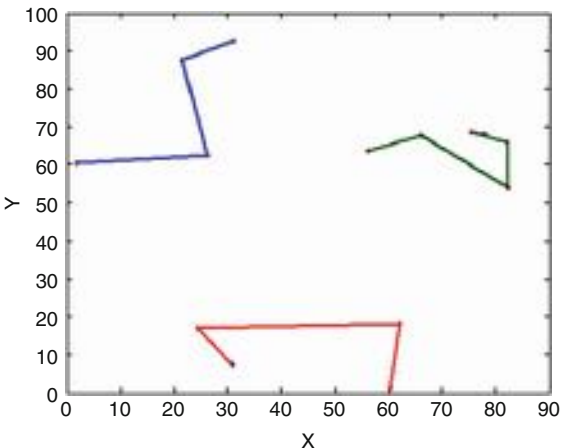
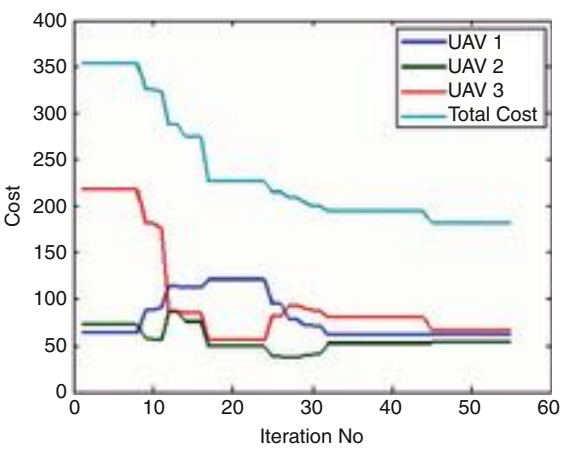


Fig. 65.3 Final waypoint assignment for three UAVs

50 iterations when compared to the first feasible solution. Figure 65.3 shows the final waypoint assignments decide to implement.

The algorithm is also tested on scenarios involving several targets. The results of a Monte Carlo experiment involving over 600 scenarios are shown in Fig. 65.4. As seen from the figure, a scenario involving 250 targets can be solved across 5 UAVs within 50 s.

65.4 Computational Experiments

This section is devoted a thorough computational study where the algorithm described in the previous section is evaluated in terms of its amenability to real-time implementations, using a realistic multi-vehicle simulation system, which is

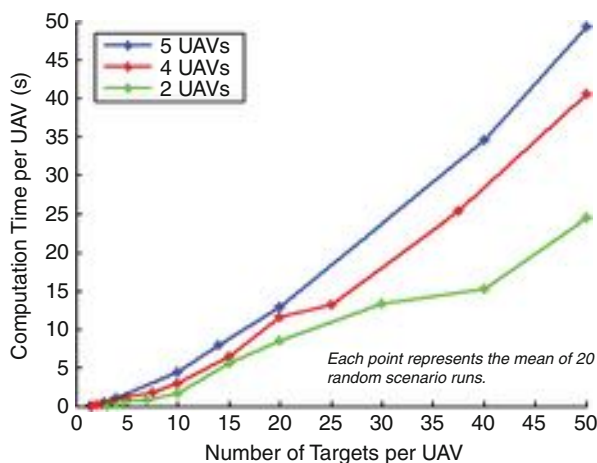


Fig. 65.4 Computation time per UAV versus the number of targets per UAV obtained from Monte Carlo computational experiments. This plot shows how the computational effort (per UAV) scales with respect to the number of targets (per UAV)

illustrated in Fig. 65.5. The simulation system, described in detail in Cetinkaya et al. (2007), simulates the dynamics governing each vehicle and its control systems separately, using one xPC Target platform for each UAV (Mathworks 2012a). The xPC Target platform can run, in real time, a simulation of the dynamics and the control systems developed using Matlab/Simulink (Mathworks 2012b). The task assignment algorithm is run in a distributed manner, as described in the previous section, on individual computers, which are called the mission coordination computers. The xPC Target simulation platforms, the mission coordination computers, as well as a computer running the world model are connected through a fast TCP/IP network. This network is devoted solely to physics-specific communication. Sending messages between mission coordination computers through this network is not allowed. Rather, the mission coordination computers are connected to one another through a wireless network that simulates the inter-vehicle communication process. The simulator system is particularly designed for evaluating the performance of real-time task assignment algorithms.

To be evaluated on this platform, consider the following real-time implementation of the distributed algorithm described in the previous section. Given the large set of targets obtained from the world model, the UAVs agree on a subset of the targets to execute the algorithmic procedure. Let us choose this subset to be the closest set of k target for some fixed k , which can be chosen depending on the computational resources available to the fleet. Each time one of the targets is serviced, the assignment algorithm is restarted by adding a new target to the input. In this manner, the number of targets in the optimization problem is fixed to k , even

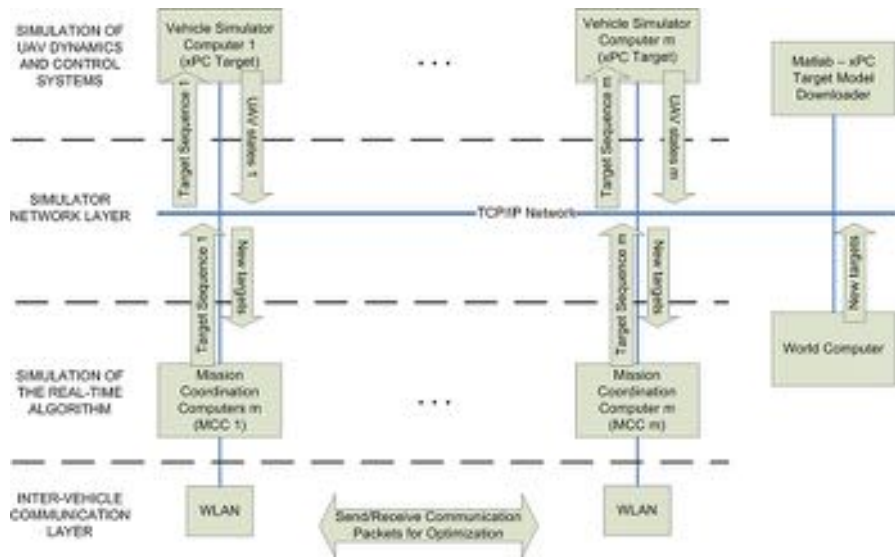


Fig. 65.5 The architecture of the simulation system

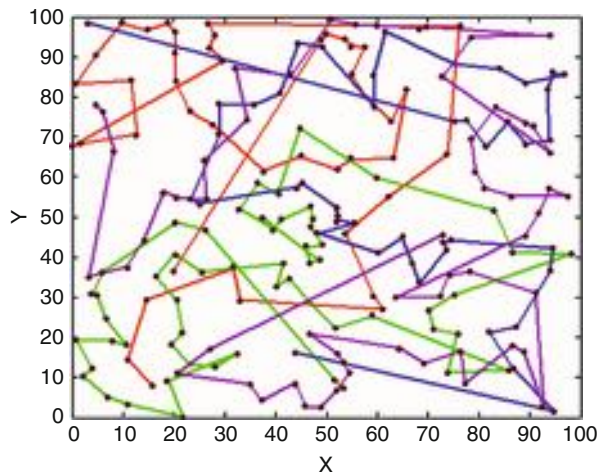


Fig. 65.6 The waypoint routes for a random pop-up task-target assignment for four vehicles. The algorithm is implemented in the receding horizon mode for 200 waypoints

if the outstanding number of targets may be very large and may change dynamically, for example, when the operators issue more targets to be serviced.

In Figs. 65.6 and 65.7, the evolution of the mission is shown for 200 and 500 targets, respectively. The computational requirements for this restricted-horizon planning is shown in Fig. 65.8, where a polynomial-time growth is observed.

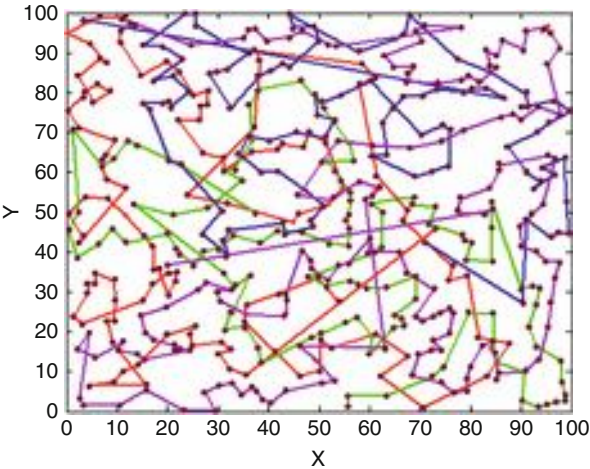


Fig. 65.7 The waypoint routes for a random pop-up task-target assignment for four vehicles. The algorithm is implemented further in the receding horizon mode for 500 waypoints

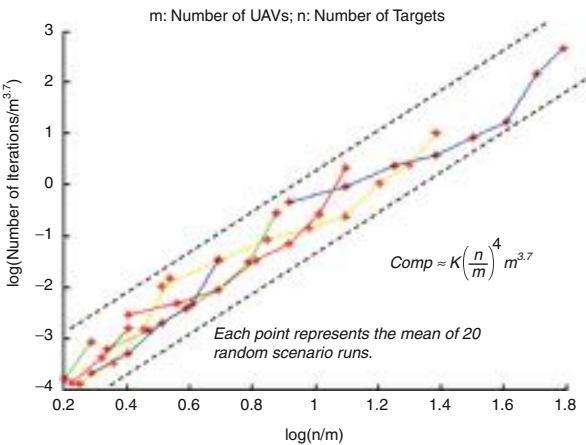


Fig. 65.8 The numeric computational complexity analysis suggests polynomial-time growth for restricted time horizons

65.5 Conclusion

A distributed task allocation algorithm is designed and implemented for real-time operation in this work. The performance of the distributed algorithm is seen to be well better than the central one. This is attributed to two main reasons, namely, the distributed nature of the algorithm and the polynomial-time subproblem structure. The former reason was expected since the algorithm contains very small communication messages to be exchanged between the decision-making agents and the distribution uses more than one processing units. The latter reason is a good advantage of the decomposition algorithm used; an NP-hard subproblem structure is seen to be polynomial-time solvable for restricted horizons after decomposition.

A test-bed for real-time distributed task allocation algorithms was also presented. This simulator is a general purpose simulator for testing UAV-dependent algorithms. The tests performed on an in-house built network mission simulator provide numerical verification of the algorithm on (a) bounded polynomial-time computational complexity and (b) hard real-time performance for problem sizes on the order of 100 waypoints per UAV.

References

- C. Archetti, M. Bouchard, G. Desaulniers, Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. *Transp. Sci.* **45**(3), 285–298 (2011)
- C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance, Branch-and-price: column generation for solving huge integer programs. *Oper. Res.* **46**(3), 316–329 (1998)
- J. Bellingham, M. Tillerson, A. Richards, J.P. How, Multi-task allocation and path planning for cooperating UAVs, in *Cooperative Control: Models, Applications and Algorithms at the Conference on Coordination, Control and Optimization*, vol. 1, (Kluwer Academic, 2003), pp. 23–41
- D. Bertsimas, J.N. Tsitsiklis, *Introduction to Linear Optimization*, 1st edn. (Athena Scientific, Belmont, 1997)
- D. Bertsimas, C. Teo, R. Vohra, On dependent randomized rounding algorithms. *Oper. Res. Lett.* **24**(3), 105–114 (1999)
- S. Bhattacharya, V. Kumar, M. Likhachev, Distributed optimization with pairwise constraints and its application to multi-robot path planning, in *Proceedings of Robotics: Science and Systems*, Zaragoza, 2010
- L. Brickman, *Mathematical Introduction to Linear Programming and Game Theory*, 2nd edn. (Springer, New York, 1989)
- A. Cetinkaya, S. Karaman, O. Arslan, M. Aksugur, G. Inalhan, Design of a distributed C2 architecture for interoperable manned-unmanned fleets, in *Conference on Cooperative Control and Optimization*, Gainesville, 2007
- A. Chabrier, Vehicle routing problem with elementary shortest path based column generation. *Comput. Oper. Res.* **33**(10), 2972–2990 (2006)
- P. Cheng, V. Kumar, An almost communication-less approach to task allocation for multiple unmanned aerial vehicles, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, 2008, pp. 1384–1389
- H. Choi, L. Brunet, J.P. How, Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robot.* **25**(4), 912–926 (2009)
- DARPA, The defense advanced research projects agency (2012), <http://www.darpa.mil>. Accessed 5 Oct 2012
- G. Desaulniers, Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Oper. Res.* **58**(1), 179–192 (2010)
- G. Desaulniers, J. Desrosiers, M.M. Solomon, *Column Generation* (Springer, New York, 2005)
- J. Desrosiers, M.E. Lubbecke, A primer in column generation, in *Column Generation*, ed. by G. Desaulniers, J. Desrosiers, M.M. Solomon (Springer, New York, 2005), pp. 1–32
- M.B. Dias, M.B. Zinck, R.M. Zlot, A. Stentz, Robust multirobot coordination in dynamic environments, in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, 2004, pp. 3435–3442
- M.B. Dias, R. Zlot, N. Kalra, A. Stentz, Market-based multirobot coordination: a survey and analysis. *Proc. IEEE* **94**(7), 1257–1270 (2006)
- E. Donna, C.L. Pape, Branch-and-price heuristics: a case study on the vehicle routing problem with time windows, in *Column Generation*, ed. by G. Desaulniers, J. Desrosiers, M.M. Solomon (Springer, New York, 2005), pp. 99–130

- M. Ehrgott, J. Tind, Column generation in integer programming with applications to multicriteria optimization. Report, University of Auckland School of Engineering, no. 651, 2007
- R.S. Garfinkel, G.L. Nemhauser, *Integer Programming* (Wiley, New York, 1972)
- S. Irnich, G. Desaulniers, Shortest path problems with time windows, in *Column Generation*, ed. by G. Desaulniers, J. Desrosiers, M.M. Solomon (Springer, New York, 2005), pp. 33–66
- B. Kallehauge, J. Larsden, O.B.G. Madsen, M.M. Solomon, Vehicle routing problem with time windows, in *Column Generation*, ed. by G. Desaulniers, J. Desrosiers, M.M. Solomon (Springer, New York, 2005), pp. 67–98
- S. Karaman, G. Inalhan, Large-scale task/target assignment for UAV fleets using a distributed branch and price optimization scheme, in *Proceedings of the 17th IFAC World Congress*, Seoul, 2008, pp. 13310–13317
- L. Lasdon, *Optimization Theory for Large-Scale Systems* (Macmillan, New York, 1970)
- T. Lemaire, R. Alami, S. Lacroix, A distributed tasks allocation scheme in multi-UAV context, in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, vol. 4, 2004, pp. 3622–3627
- M.E. Lubbecke, J. Desrosiers, Selected topics in column generation. *Oper. Res.* **53**(6), 1007–1023 (2005)
- Mathworks, xPC target user's guide, mathworks (2012a), http://www.mathworks.com/help/pdf_doc/xpc/xpc_target_ug.pdf. Accessed 5 Oct 2012
- Mathworks, Matlab external interfaces manual (2012b), http://www.mathworks.com/help/pdf_doc/matlab/apiext.pdf. Accessed 5 Oct 2012
- K. Melhorn, M. Ziegelmann, Resource constrained shortest paths, in *Proceedings of the 8th Annual European Symposium on Algorithms*, Saarbrücken. Lecture notes in computer science, vol. 1879 (Springer, New York, 2000), pp. 326–337
- M. Minoux, A class of combinatorial problems with polynomially solvable large scale set covering/partitioning relaxations. *RAIRO Oper. Res. Recherche Opérationnelle* **21**(2), 105–136 (1987)
- S. Ponda, J. Redding, H. Choi, J.P. How, M. Vavrina, J. Vian, Decentralized planning for complex missions with dynamic communication constraints, in *American Control Conference (ACC'10)*, Baltimore, 2010, pp. 3998–4003
- A. Pongpunwattana, R. Rysdyk, J. Vagners, D. Rathbun, Market-based co-evolution planning for multiple autonomous vehicles, in *Proceedings of the AIAA Unmanned Unlimited Conference*, Chicago, 2003
- P. Raghavan, C.D. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* **7**(4), 365–374 (1987)
- T.K. Ralphs, M.V. Galati, Decomposition and dynamic cut generation in integer linear programming. *Math. Program. A* **106**(2), 261–285 (2006)
- A. Richards, Y. Kutawa, J. How, Experimental demonstrations of real-time MILP control, in *GNC'03: AIAA Control Navigation and Guidance Conference and Exhibit*, Austin, 2003
- T. Schouwenaars, E. Feron, B. de Moor, J.P. How, Mixed integer programming for multi-vehicle path planning, in *ECC'01: European Control Conference*, Porto, Sept 2001, pp. 2603–2608
- W. Sheng, Q. Yang, S. Ci, N. Xi, Multi-robot area exploration with limited-range communications, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, 2004
- P. Toth, D. Vigo, *The Vehicle Routing Problem* (Society for Industrial and Applied Mathematics, Philadelphia, 2002)
- F. Vanderbeck, Implementing mixed integer column generation, in *Column Generation*, ed. by G. Desaulniers, J. Desrosiers, M.M. Solomon (Springer, New York, 2005), pp. 331–358
- F. Vanderbeck, M.W.P. Savelsbergh, A generic view of Dantzig-Wolfe decomposition in mixed integer programming. *Oper. Res. Lett.* **34**(3), 296–306 (2006)
- R. Zlot, A. Stentz, M.B. Dias, S. Thayer, Multi-robot exploration controlled by a market economy, in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'02*, Washington, vol. 3, 2002, pp. 3016–3023

Andrew Kopeikin, Sameera S. Ponda, and Jonathan P. How

Contents

| | | |
|--------|---|------|
| 66.1 | Overview | 1620 |
| 66.2 | Design Challenges | 1622 |
| 66.3 | Communications and Networking..... | 1625 |
| 66.3.1 | System and Network Architectures..... | 1625 |
| 66.3.2 | Wireless Channels | 1626 |
| 66.3.3 | Network Topology and Consensus | 1629 |
| 66.3.4 | Information Routing..... | 1630 |
| 66.4 | Coordination Under Communication Constraints | 1632 |
| 66.4.1 | Information Consensus..... | 1632 |
| 66.4.2 | Effects on Cooperative Control | 1634 |
| 66.4.3 | Effects on Cooperative Decision Making | 1636 |
| 66.5 | Motion Planning for Communications | 1638 |
| 66.5.1 | Graph Theoretic Methods and Potential Fields | 1638 |
| 66.5.2 | Reactive and Adaptive Techniques | 1640 |
| 66.5.3 | Implementation Examples | 1641 |
| 66.6 | Communication Relays for Network Support | 1643 |
| 66.6.1 | Controlling Relay Chains | 1643 |
| 66.6.2 | Deploying Relays to Support a Network | 1645 |
| 66.6.3 | Controlling Communications Through Task Allocation..... | 1647 |
| 66.7 | Concluding Remarks and Remaining Challenges | 1648 |
| | References | 1650 |

Abstract

The communication network is a fundamental component of a multi-UAV system. It enables exchanges in command and control messages and allows remotely sensed mission data to be sent to processing centers. Proper control of the network, the topic of this chapter, is critical for the system to function

A. Kopeikin (✉) • S.S. Ponda • J.P. How
Department of Aeronautics and Astronautics, Aerospace Controls Laboratory Massachusetts
Institute of Technology, Cambridge, MA, USA
e-mail: kopeikin@MIT.EDU; sponda@alum.mit.edu; jhow@mit.edu

properly. Following a brief overview, Sect. 66.2 lists some of the main challenges in network communication control. Next, Sect. 66.3 provides background discussion in wireless networking including system architecture, wireless channel performance, topology models, and information routing. Bounds and limitations in cooperative control and decision making because of network limitations are provided in Sect. 66.4. Next, the chapter provides several network control strategies for multi-UAV systems. First, Sect. 66.5 presents motion planning methods to control the topology, and second, Sect. 66.6 describes relay deployment techniques to extend the performance of the network. Finally, Sect. 66.7 provides some closing remarks and directs the reader to further literature on this topic.

66.1 Overview

Network and communication systems such as the Internet, social media, cellphones, and countless other wireless systems have revolutionized human way of life. These same technologies enable teams of multiple human and unmanned agents to share situational awareness and planning strategies to cooperate and improve their performance in executing a set of mission tasks. As such, the communication network is a fundamental component of a multiple unmanned aerial vehicle (UAV) system. Controlling it is often just as important as controlling the vehicles because inadequate communications can significantly degrade team performance. Even though ad hoc networks have been actively studied in the past decades for military (Leiner et al. 1996), home (Negus et al. 2000), and metropolitan (Ritter et al. 2001) network applications, they have only recently started to generate significant interest for cooperative control of multiple robotic air, ground, and sea vehicles.

There are several general architectures to deploy multiple UAVs as a team. UAVs may be connected only to a central node, such as a human-operated control station, which receives information, computes a plan of action, and sends instructions specific to each vehicle (centralized control). Alternately, UAVs may be connected directly to each other, exchange information, and each formulate their own plan without a designated leader (decentralized control) (Stipanović et al. 2004; Choi et al. 2009). Hybrid centralized and decentralized solutions also exist, and UAVs may also cooperate with other manned and unmanned space, air, ground, and sea agents over the network in a joint effort (Chandler et al. 2001; Chandler and Pachter 2001; Marshall et al. 2011) as illustrated in Fig. 66.1. In all cases, team coordination requires agents to exchange their state information, observations of the world, and control decisions such as task allocation or motion planning. Furthermore, if the UAVs are being used to gather information, as is often the case, the collected data may need to be communicated to a designated point for analysis, possibly in real time (Office of the Secretary of Defense 2009).

Because UAVs are highly mobile vehicles, information is most commonly exchanged across the network using wireless communication. Signals containing encoded messages travel between transmitting and receiving radio modules over wireless channels. The quality of the channel is fundamentally based on the strength

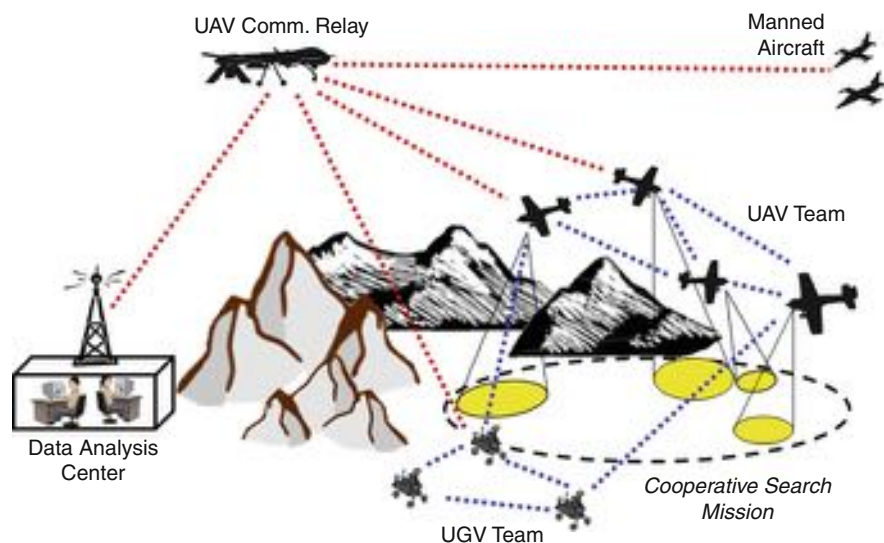


Fig. 66.1 A networked multi-UAV team performing a cooperative search mission. UAVs are seen cooperating with ground vehicles, and a communication relay UAV connects the team to the base as well as to other manned aircraft involved in the mission

of the signal at the receiver compared to noise and interference in the environment, a value named the signal to interference and noise ratio (SINR) (Goldsmith 2005). The quality of the link drives the probability that information transmitted will successfully be received, and affects the rate at which information can be exchanged over that channel.

One typical objective in controlling a multi-UAV team is to ensure that the system remains in a state where the communication network can support cooperative planning (or coordination) and data collection requirements. The network itself can be controlled through its topology, which describes the set of interconnections between nodes and how strong those connections are. There are several general methods to control the topology in a team of mobile agents:

1. Wireless channels generally degrade with increasing distance and due to obstacles in the line of sight. Cooperative motion planning can position the vehicles in locations which improve data transfers (Goldsmith 2005).
2. The received signal can be improved by increasing transmission power (Santi 2005), changing the orientation or directing the transmitter (Ahmed et al. 2011), or choosing the proper time and location to transmit (Yan and Mostofi 2010).
3. The topology can be controlled by actively choosing which links should be active and which exchanges are better suited for multiple hop node-to-node transfers (Zavlanos and Pappas 2008).
4. Additional mobile agents can be deployed as communication relays to extend the performance of the underlying network (Zhu et al. 2009).

These control methods can be employed individually or in tandem to achieve the performance required in the communication network. The design trade must

properly consider the benefits, costs, and implications in computation, communication, and vehicle resources associated with each of these methods.

The concept of multi-UAV teams is still fairly new operationally. As such, the methods to control communication in these systems are actively evolving through research studies (discussed in this chapter) and lessons learned from other types of networks. Aerial networking itself has been employed for years in commercial and military aircraft operations (Mulkerin and Zernic 2003). Modern generation strike aircraft and aerial intelligence platforms commonly share mission parameters among themselves via Tactical Data Links for real-time cooperation. These systems however are limited in range, throughput, and interoperability and exhibit delays. Because information sharing has become so crucial in warfare, large-scale UAVs are being deployed as persistent communication relays to better connect aircraft and ground nodes (Jonson et al. 2008). Similar concepts are under active investigation to rapidly provide communications for large-scale disaster recovery scenarios similar to Hurricane Katrina (Federal Communications Commission 2011) or first response in challenging environments such as wild land fire fighting (Zajkowski et al. 2006). It is paramount to understand how best to employ these systems to maximize their network support. Many other multi-unmanned vehicle concepts are being considered for military (U.S. Air Force Chief Scientist AF/ST 2010), exploration (Anderson et al. 2003), and surveying (Leonard et al. 2010) applications which will also implicitly require control of communications to operate.

The purpose of this chapter is to provide an overview of the different elements of network communication control for teams of UAVs. It describes the challenges involved in controlling communications in such multi-agent systems and provides technical fundamentals in wireless communications theory as support. The chapter analyzes recent literature studies about the effects of limited interagent communications on the performance of team cooperation. Furthermore, it reviews recently developed techniques to control network communications in a team using motion control, communication relays, and cooperative task allocation strategies. The objective is to familiarize the reader with this relatively new and rapidly evolving topic.

66.2 Design Challenges

There are numerous and often competing challenges in controlling network communications in a multi-UAV system. The design must properly balance between ensuring that the communication requirements are met without over-constraining the system design and operation. Furthermore, even if designed and controlled properly, network communications still exhibit limitations and uncertainties in performance which impact the system.

The control of network communications follows the same feedback control principles used for many other types of systems. The state, or topology of the network, is sensed and used as feedback, with a model of its dynamics, to predict the outcome of control inputs. Modeling and accurately predicting the performance of

wireless channels is a significant challenge in this process. In general, links degrade with increasing internode distance due to path loss. Signal shadowing occurs due to obstacles in the environment such as buildings, mountains, or even the platform vehicle itself. Furthermore, multipath replicas of the signal reflecting and scattering off these obstacles constructively or destructively interfere with each other at the receiver (see Fig. 66.3). Path loss, shadowing, and multipath fading effects are increasingly detrimental and difficult to predict in more congested operating environments (Santi 2005; Goldsmith 2005; Mostofi et al. 2009). Modeling techniques for these dynamics are provided in Sect. 66.3. Fortunately, large- and medium-scale UAVs often operate in open skies, which mitigates many of these detrimental effects. However, actively studied concepts of mini/micro-UAV teams deployed below the tree line, inside buildings, or in urban settings will be significantly affected by these dynamics (Warwick 2009). Similarly, noise and interference from other emitting sources in the environment can be difficult to predict and can change both in time and by location Wagle and Frew (2011). In some cases, another source may actively seek to interfere or “jam” communications to degrade the system performance as a countermeasure (U.S. Air Force Chief Scientist AF/ST 2010). For similar reasons, measurements of the signal strength, while accurate, may also fluctuate significantly at time scales of only tens or hundredths of a second. The uncertainties in the model and the rapid variation in the measurements pose a significant challenge in the network feedback control loop and lead to limited performance guarantees of the system (Goldsmith 2005).

Given the ability to control the network topology, the problem of efficiently routing information between nodes is a challenge in itself (Goldsmith 2005). Different types of data have different communication requirements, but in general the typical objective is to route data to its destination with minimal delay, over links with sufficient data rate capacity and with minimal number of packets dropped due to error. A wireless network typically operates on one or more communication channels in a specific range of the electromagnetic spectrum. As such, nodes transmitting on the same or neighboring channels interfere with each other, which generates a design trade (Xue and Kumar 2004) best illustrated through a simple example. Assume that a UAV needs to transmit time sensitive data to a distant base. One option is to increase its transmitter power enough to send the data directly to the base, as UAV2 does in Fig. 66.2a. However, this higher transmit power now interferes more with other UAVs also trying to communicate. Furthermore, the long link may be strong enough to reach the base under nominal conditions, but may not be robust to channel fading. Another option then is for the UAV to transmit over a more robust link and at a lower power to another less distant UAV (as UAV2 now does in Fig. 66.2b), which can then relay the message, possibly through additional hops, to the base. This reduces interference from that UAV, but may cause delays in end-to-end delivery of the data, and cost overhead in the network to establish the multi-hop route. The challenge of selecting the best route and power settings is further exacerbated as the number of agents increases, as wireless link qualities continuously fluctuate under fading, and because the network topology is dynamic as agents move around to execute the mission (Goldsmith 2005). This is especially

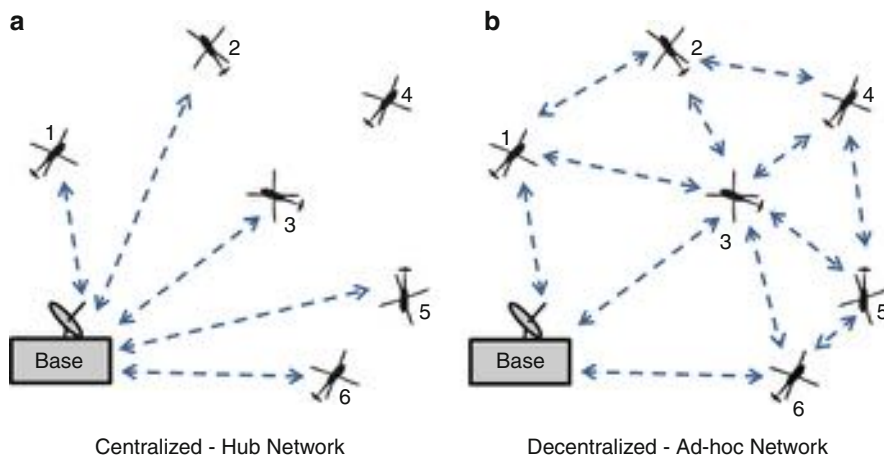


Fig. 66.2 Network architectures. In (a) all UAVs communicate only with the base, and UAV4 is out of range and disconnected. In (b) connections can be shorter, and UAVs interconnect and can route data via multi-hop

relevant for multi-UAV systems, which can involve vehicles traveling rapidly over large distances.

Multi-agent networks for all types of systems face the communication control challenges described above. The degree to which these challenges affect the system changes based on the domain of operation. For instance, unmanned ground vehicles (UGVs) typically operate in environments with obstacles to line of sight, resulting in channel fading, but they have the ability to stop when a suitable location to transmit is encountered (Zhang and Quilling 2011). Autonomous underwater vehicles (AUVs) have limited underwater communication capabilities and may only be able to exchange coordinated control information once every few hours upon surfacing (Leonard et al. 2010). Unattended ground sensors (UGS) forming a fixed wireless sensor network (WSN) may be less susceptible to dynamic changes in topology but need to be power conscious when transmitting since their battery lifetime is limited (Ibrahim et al. 2009). There are a number of challenges especially relevant for UAVs. First, the size, weight, and power (SWaP) of the UAV radio device can be limited by design constraints of the vehicle and compete with requirements imposed by other onboard systems. Next, unless a vertical flight aircraft is used, the antenna position and orientation, which affects channel quality (Ahmed et al. 2011), is subject to vehicle dynamics. To mitigate this problem, some designs have included multiple receivers on the vehicle meeting SWaP constraints (Kung et al. 2010; Yanmaz et al. 2011). Similarly, a fixed wing vehicle is unable to hold a fixed position considered to be “optimal” for communications. Also interestingly, some lightweight materials commonly used for UAVs, such as carbon fiber, have conductive properties which shield radio signals and can cause loss of link (Krueger 2002). Finally, SWaP constraints in micro-UAVs may limit onboard

computing capability and therefore limit the complexity of the algorithm used for motion control, relay positioning, or task assignment.

Even with proper system design and implementation, the communication network will still have limitations that can significantly degrade the performance of the multi-UAV system. Algorithms to control vehicle motion, transmission power, information routing, and task allocation must rapidly converge to a good solution and require minimal message exchanges to mitigate delays. Command delays, however short, may cause a formation of UAVs to perform inefficiently or *churn* (effectively go unstable), which can have disastrous effects such as vehicle collisions (Schwager et al. 2011). Delayed and dropped messages sent to planning agents, whether centralized or decentralized, can cause inconsistencies in situational awareness and, as a result, flawed planning (Ponda et al. 2010). Similarly, message delays in decentralized planning may prevent agents from reaching consensus on a plan (Johnson et al. 2011). These limitations, therefore, have implications on the system architecture, choice of algorithms, and performance guarantees for multi-UAV systems.

66.3 Communications and Networking

This section provides the technical background in wireless communications and networking that will be utilized by many of the multi-agent network control methods in this chapter.

66.3.1 System and Network Architectures

The design and control of the communication network depends on the system architecture and data transfer requirements. For instance, traditional cellular phone networks form a hub-and-spoke model between a fixed base station and mobile users in its area. Telephone voice data requires minimal delay to be effective, but does not need high data rates and has some error tolerance. Conversely, data networks for Internet downloads require high and bursty (sporadic) data rates and dynamic bandwidth allocation between a gateway, routers, and end users. These networks can tolerate some delays but should be relatively error-free (Modiano 2012).

A multi-UAV system network typically needs to support two types of data. The first type consists of command and control messages including state information (telemetry), observations of the world (e.g., estimated target location), and control data (e.g., waypoints or task allocation). These messages have low bandwidth requirements but must be exchanged with minimal delay and error for effective team coordination. Second, information gathered from sensors such as video, still images, atmospheric samples, and other types of remote sensing data may also need to be transmitted to designated processing, exploitation, and dissemination (PED) centers (Christmann and Johnson 2007). Again, communication requirements depend on

the data type, but live video, for instance, requires minimal delay and high data rates and has some fault tolerance (Modiano 2012).

A multi-UAV system architecture built on centralized control, where vehicles remain within transmission range of the command and control node, can operate using a hub-and-spoke network model, similar to a cellular system (Fig. 66.2a). Some implementations can even extend command and control range by using a low data rate satellite communication system (Marshall et al. 2011). However, many of the applications envisioned for multi-UAV teams require greater topology flexibility, with internode information exchanges and relaying (Fig. 66.2b). This second architecture (Fig. 66.2b) closely follows concepts of ad hoc networks, which have received significant interest in many disciplines (Santi 2005), and applies to most of the systems studied in this chapter.

Network architectures are generally abstracted to a multilayer design, where each layer is individually responsible for a different role in communicating information. Modern ad hoc networks typically consist of (1) a *physical layer* responsible for sending bits across a wireless channel, (2) an *access layer* to deconflict channel usage from multiple users, (3) a *network layer* to route information from node to node, (4) a *transport layer* to regulate flow and retransmissions, and (5) the *application layer* which uses and creates data (Modiano 2012). While understanding these layers is important in network design, many of the applications discussed in this chapter, along with latest research in ad hoc networks, push for cross layer design optimization (Goldsmith 2005; Xiao et al. 2004; Kaabi and Filali 2010; Hiertz et al. 2010). As such, further discussion of these layers is limited in this chapter, and the reader is referred to Bertsekas and Gallager (1992) and Kawadia and Kumar (2005) for more details.

66.3.2 Wireless Channels

In wireless networks, data is exchanged over communication channels by sending information encoded and modulated from a transmitter, over the air using a signal, to a receiver which demodulates and decodes to process the data. The strength of the received signal power compared to surrounding noise, or signal-to-noise ratio (SNR), γ , is a critical parameter in assessing the quality of the link. This value is commonly modeled with the simplified relationship:

$$\text{SNR: } \gamma = \frac{gP}{N_o W} \quad (66.1)$$

$$g = K \left(\frac{d_o}{d} \right)^\alpha \quad (66.2)$$

where P is the transmission power, $\frac{N_o}{2}$ is the power spectral density of the environment noise, and W is the bandwidth of the signal. The channel gain g is generally modeled as Eq. (66.2) in its most simple form, in which K is a gain based

on equipment characteristics, d is the Euclidean distance between the transmitter and receiver, and α is the path loss exponent which equals 2 in free space and up to 6 in environments congested with obstacles (Goldsmith 2005). This relationship represents the path loss dynamics experienced in wireless channels, in which γ decreases with increasing distance.

The next effect to consider is shadowing of the signal due to obstacles in the line of sight. A practical way to model shadowing in uncertain environments is to vary γ according to a lognormal distribution with variance σ^2 which depends on the density of obstacles in the environment (Goldsmith 2005). A convenient method to express this is to convert the channel gain in Eq. (66.2) into decibels (g_{dB}), where shadowing now becomes a Gaussian-distributed random variable centered on the path loss gain:

$$g_{dB} = K_{dB} - 10\alpha \log \left(\frac{d}{d_o} \right) - \mathcal{N}(0, \sigma_{dB}^2) \quad (66.3)$$

The distribution has been shown to be spatially correlated to X_c , the typical size of obstacles in the area. The spatial covariance $A(x)$ in the distribution between two points separated by distance x can be expressed as

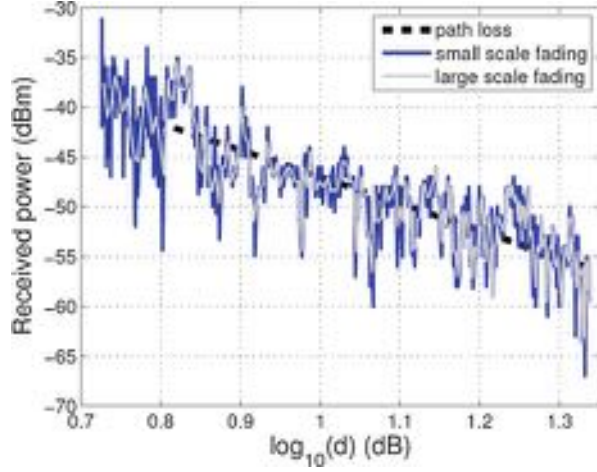
$$A(x) = \sigma_{dB}^2 e^{-x/X_c} \quad (66.4)$$

This expression can be used to predict the amount of time a moving vehicle may undergo deep shadowing fades, or determine how to reposition the vehicle to exit a shadowed area (Mostofi et al. 2009; Mostofi 2009).

Multipath fading, which occurs because of constructive and destructive interference from scatterings and reflections of the signal by the obstacles, can be modeled using a nonstationary random process centered on channel gain g subject to path loss and shadowing in Eq. (66.3). Models using Rayleigh, Rician, or Nakagami distributions with a tunable fading parameter have been shown to follow empirical measurements of this dynamic (Mostofi et al. 2009; Goldsmith 2005). Multipath fading values can be added to the g_{dB} expression in Eq. (66.3) and then used to predict the overall SNR using Eq. (66.1). This dynamic can vary significantly over small distances and is often modeled without spatial correlation for simplicity (Mostofi et al. 2009). The three dynamics can be seen using experimentally measured wireless channel signal strength measurements in Fig. 66.3 obtained from (Mostofi et al. 2009).

Many field experiments have studied the dynamics of link performance in greater detail. Research in Wagle and Frew (2011) used data collected from aerial links to measure an additional temporal correlation factor in the above dynamics. A study in Ahmed et al. (2011) observed that link qualities vary depending on the relative orientations between the transmitter and receiver, even when using omnidirectional radio modules. Multiple antenna configurations were similarly used in Kung et al. (2010) and Yanmaz et al. (2011) to improve the overall reception for the vehicle through diversity.

Fig. 66.3 Measured dynamics of a wireless channel. The *solid blue curve* represents the signal strength, which varies at a small scale due to multipath fading. Spatially correlated shadowing can be seen by averaging locally (*light gray curve*). The average of the large-scale variations then follows distance-dependent path loss (*dashed line*) (Figure and caption copyright Mostofi et al. (2009))



Relationships in Eqs. (66.1)–(66.4) model the dynamics of a single wireless channel. In a multi-user network, multiple agents simultaneously communicating on the same or neighboring channels interfere with each other. In this case, SNR is often rewritten as the signal to interference and noise ratio (SINR) represented by γ_{ij} for the channel from node i to j (Ribeiro 2011)

$$\text{SINR: } \gamma_{ij} = \frac{g_{ij} P_{ij}}{(N_o W)_j + \frac{1}{\rho} \sum_{(m,n) \neq (i,j)} g_{mj} P_{mn}} \quad (66.5)$$

Here, the interference caused by other nodes m transmitting to n adds to the noise at receiver j in the denominator. While some of these effects can be mitigated through signal code processing ($\frac{1}{\rho}$), a common method of deconfliction is to divide channel usage by time (TDMA) where $\rho = 1$ (Ribeiro 2011).

The SNR in Eq. (66.1) or SINR in Eq. (66.5) is a fundamental indicator of wireless link performance and can be used to predict operationally relevant data transmission metrics. First, the theoretical *capacity* of the channel can be evaluated using Shannon's equation:

$$c_{ij} = W_{ij} \log_2(1 + \gamma_{ij}) \quad (66.6)$$

which represents its data rate in *bits/sec*. While this value is an upper bound, and actual data transmissions are often set to significantly lower rates due to fading (Goldsmith 2005), it is a useful indicator in applications where information throughput must be considered. Another important performance metric is the *bit error rate* (BER) or the probability an information bit will be dropped, which can require packet retransmissions and inefficient data flow. BER decreases with increasing γ , and the relationship depends on the specific modulation scheme.

Equation (66.7) represents the relationship for a common modulation method, where Q is defined in Goldsmith (2005):

$$\text{BER}_{ij} = Q(\sqrt{2\gamma}) \quad (66.7)$$

Many of the communication network control methods discussed in this chapter have an objective to optimize or meet some threshold values for capacity, BER_{ij} , or even information delay which relates to c_{ij} (Xiao et al. 2004). The models listed in this section highlight the challenges associated with predicting and controlling wireless channel performance due to dynamic uncertainties in fading and coupling with other agents using the network.

66.3.3 Network Topology and Consensus

Another relevant challenge, especially in ad hoc networks, involves optimizing the network topology (i.e., determining which nodes need to form interconnections), to achieve the desired connectivity for information flow. The network topology can be represented as a graph $G = (V, E)$ with a set of n nodes $V = \{1, 2, \dots, n\}$ and edges $E \subseteq V \times V$ indicating connectivity between nodes. The graph can be represented using an *adjacency matrix*, A , with $n \times n$ elements a_{ij} , where $a_{ij} = 1$ if i can communicate with j and $a_{ij} = 0$ otherwise. Each row of the adjacency matrix can be read directly to identify the set of neighbors for every agent. In some formulations, $a_{ij} = w$, where $w \in [0, 1]$, depending on the quality of the link and topology properties. The *neighborhood* of agent i , $N_i = \{j \in V : a_{ij} \neq 0\}$, is defined as the set of agents that agent i can communicate with directly (Olfati-Saber et al. 2007). For instance in the network in Fig. 66.2b, the neighbors of UAV4 are $N_4 = \{2, 3, 5\}$. Theoretical formulations have established bounds on how many agents each node should be connected to for the network to become asymptotically disconnected or connected (Xue and Kumar 2004). For this reason, many multi-vehicle systems control the network so that each agent remains connected to a specific number of neighbors (Atay and Bayazit 2008; Zhu et al. 2009; Goddemeier et al. 2011; Razafindralambo and Simplot-Ryl 2011; Reich et al. 2008; Zavlanos et al. 2011). Many of these schemes are discussed in Sects. 66.5 and 66.6.

Analyzing the network topology provides useful insights on how information propagates throughout the network and how the team of agents reaches consensus on quantities of interest such as plans, parameters, and situational awareness. The stability and convergence properties of most consensus algorithms can be predicted by analyzing the spectral properties of the network graph *Laplacian*, \mathcal{L} , for a given network structure (Olfati-Saber et al. 2007). The Laplacian is another related matrix representation of the network topology and is formulated as

$$\mathcal{L} = D - A \quad (66.8)$$

using the adjacency matrix A and the *degree matrix*, $D = \text{diag}(d_1, \dots, d_n)$, with elements $d_i = \sum_{j \in N_i} a_{ij}$, representing the outgoing connections of each agent (sum over its neighbors). The maximum degree of the graph, $\Delta = \max_i d_i$, is useful in determining spectral properties of the network Laplacian. In particular, all of the eigenvalues of the Laplacian \mathcal{L} in the complex plane are bounded by a closed disk of radius Δ centered at $\Delta + 0j$. Assuming the graph is undirected, where $a_{ij} = a_{ji}$, the Laplacian \mathcal{L} is symmetric, and therefore its eigenvalues lie on the real line within the set $[0, 2\Delta]$. These can be ordered sequentially as

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2\Delta. \quad (66.9)$$

The first (smallest) eigenvalue of \mathcal{L} is always zero ($\lambda_1 = 0$) since every row sum is $\sum_j l_{ij} = 0$, and is known as the trivial eigenvalue. The second eigenvalue, λ_2 , is known as the *algebraic connectivity* or *Fiedler value* of the graph and is always positive for *strongly connected* graphs where a single- or multi-hop path exists from any agent to every other agent. The algebraic connectivity λ_2 determines the speed of convergence for most consensus algorithms, and as such, many multi-vehicle applications attempt to maximize this value by controlling the network topology (Ibrahim et al. 2007, 2009; Yan and Mostofi 2010; Michael et al. 2009; Zavlanos et al. 2011) (see Sects. 66.5 and 66.6). The last eigenvalue, λ_n , is related to stability guarantees for reaching an equilibrium in time-delayed networks, providing associated bounds on the maximum allowable delays (Olfati-Saber et al. 2007). Further details on consensus algorithms are provided in Sect. 66.4.1.

It must be noted that graph theoretic methods are simplifications of multi-UAV network representations, since connectivity is often assumed to be binary and links are considered active if agents are within a threshold distance of each other (Santi 2005). These methods usually do not consider realistic wireless channel dynamics as described in the previous subsection or some of the complexities in information routing discussed next. Nevertheless, they do provide helpful mathematical interpretations of the potential performance of the network and are useful in guiding the system design to achieve configurations more suitable for information exchange (Michael et al. 2009).

66.3.4 Information Routing

Since envisioned multi-UAV team applications require similar flexibility to ad hoc wireless network architectures, network routing becomes a significant challenge that must be considered in controlling the system. Given the wireless channel properties and network topology, information must be routed along links with sufficient capacities, and arrive at its destination with minimum error (BER) and delay, as dictated by data requirements. Routing is an active area of research in the wireless networking community (Al-Karaki and Kamal 2004). If the network involves multiple channels on different frequencies, as is often the case, the problem extends to finding the optimal route and channel allocation (Kaabi and Filali 2010). Strategies range from decentralized, highly adaptable methods to more centralized

optimized schemes (Goldsmith 2005). Understanding how a particular network system routes information is necessary to adequately control communications in a team of UAVs, as it may affect where agents are positioned to provide a suitable topology.

Several industry standards (802.11s, 802.15, 802.16) drive the design and operation of mesh networking modules commonly used in ad hoc networks (Kaabi and Filali 2010; Hiertz et al. 2010). These standards rely on decentralized routing mechanisms. A commonly used protocol is Ad hoc On-Demand Distance Vector (AODV) (Perkins and Belding-Royer 1999). In AODV, a node which needs to transmit broadcasts a “request to send” message which identifies the destination node. Neighbors receive the message, compute a cost metric (discussed later), and rebroadcast the message. The process continues, where cost is computed at every hop, until the destination is reached. Each node stores its lowest cost originator. Once a designated amount of time has elapsed, the destination node sends a unicast response to its lowest cost originator, which similarly relays the response to backtrack the lowest cost route. Other similar routing techniques include Dynamic Source Routing (DSR) (Perkins and Belding-Royer 1996), Optimized Link State Routing (OLSR) (Adjih et al. 2004), and B.A.T.M.A.N (Garroppo et al. 2010). Different metrics are implemented depending on the objectives, which include lowest hop count, expected transmission count (ETX in Eq. (66.10), Salonidis et al. 2007; Decouto et al. 2005), and airtime metric (C_{Aij} in Eq. (66.11), Ribeiro 2008). Here, p_{pkt} is the packet loss ratio (based on BER), O is overhead, B_t is the test-frame length, and r is the data rate (based on capacity):

$$ETX_{ij} = \frac{1}{1 - p_{pkt}} \quad (66.10)$$

$$C_{Aij} = \left(O + \frac{B_t}{r} \right) \frac{1}{1 - p_{pkt}} \quad (66.11)$$

The protocols listed above are implemented in practice, such as on Zigbee modules common in robotics research (Bhatia and Kaushik 2008), because of their rapid response to changes in wireless channel performance and topology. A significant drawback to them, however, is that information from different nodes is routed mostly independently of other nodes and tends to use the same high-performance routes. This can lead to congestion in high data rate applications and unfairness in node usage (Salonidis et al. 2007). Researchers have proposed different network optimization schemes to address this, which invoke conservation of information flow at each node and link utilization constraints. For instance, authors in Toumpis and Gitzenis (2009) optimize routing fairness by considering interference between nodes and using electrical flow analysis. Similarly, convex optimization methods have optimized routing through simultaneous link scheduling and transmitter power allocation (Xiao et al. 2004; Cruz and Santhanam 2003). While these schemes produce better routing solutions, they mostly consider static networks and would require frequent recomputation in a dynamic environment to adapt to realistic changes in the topology (Goldsmith 2005).

66.4 Coordination Under Communication Constraints

For multi-UAV teams to cooperatively execute a set of objectives, the communication network must exchange command and control messages and, when necessary, remotely sensed data. The inability to communicate sensed data, such as live video, to the designated processing centers may render the multi-UAV system ineffective for its mission. Worse however, failure to properly exchange command and control messages can lead to dangerous system failures or unintended consequences. This is particularly true in decentralized systems, where these messages enable vehicle control and team decision making. Inadequate team control can lead to formation instability where one or multiple agents increasingly diverge from the desired behavior and risk colliding or separating from the team. Poor decision making may prevent the team from reaching consensus on a plan and remain idle without accomplishing anything or, sometimes worse, take action on an undesirable plan leading to unwanted consequences. Despite taking appropriate measures to control the communication network, transmission delays and dropouts are virtually unavoidable and may occur unexpectedly. This naturally limits the rate of information agents have available to make control decisions or formulate a team plan. Graph theoretic properties introduced in Sect. 66.3, such as the graph Laplacian \mathcal{L} and its set of eigenvalues λ_i , can provide insight into network bounds on formation stability or team consensus (Olfati-Saber et al. 2007). This section focuses on identifying these limitations, and the following two sections describe network control strategies to improve team performance and network connectivity.

66.4.1 Information Consensus

A critical component of team coordination strategies involves performing consensus, which is defined as reaching an agreement on quantities of interest that depend upon the states of all agents (Olfati-Saber et al. 2007). Most distributed planning and control approaches employ *consensus algorithms* to reach agreement on plans, situational awareness, or other desired parameters. A consensus algorithm is a set of rules, or protocol, that determines how the information is exchanged between agents to ensure that the team will convergence to an equilibrium on the parameters of interest. For example, a simple algorithm to perform consensus on a continuous parameter z can be expressed as an n th order linear system where each agent executes the following protocol:

$$\begin{aligned}\dot{x}_i(t) &= \sum_{j \in N_i} (x_j(t) - x_i(t)) \quad \forall i \\ x_i(0) &= z_i \in \mathbb{R}\end{aligned}\tag{66.12}$$

Collectively the team dynamics can be written as the linear system

$$\dot{\mathbf{x}}(t) = -\mathcal{L}\mathbf{x}(t)\tag{66.13}$$

where \mathcal{L} is the graph Laplacian as described in Sect. 66.3.3. Since Eq. (66.13) is a linear system, the spectral properties of \mathcal{L} determine the stability and convergence rate of the consensus algorithm. In particular, as mentioned in Sect. 66.3.3, for any strongly connected graph, all of the nontrivial eigenvalues of \mathcal{L} are positive, and thus the consensus algorithm is globally asymptotically stable. Furthermore, the algorithm has an exponential convergence rate determined by the algebraic connectivity λ_2 and is guaranteed to achieve a unique equilibrium $\bar{z} = 1/n \sum_i z_i$, which is the average of all the agents' initial values (Almost identical guarantees can be made for discrete representations of Eq. (66.13) as well.) (Olfati-Saber et al. 2007). Note that, in general, these convergence and stability properties are dependent upon \mathcal{L} and λ_2 , which are defined by the network structure, and not on the particular consensus algorithm used.

In realistic environments, there are several effects that complicate the transmission of messages between agents and thus impact the performance of consensus algorithms. Some examples include time delays for transmitted messages and varying network topologies where links between agents are subject to change over time. Recent research has focused on analyzing consensus convergence and stability properties in the presence of time delays and dynamic networks. It turns out that the global exponential stability guarantees can be extended to dynamic networks. As long as the graph $G(t)$ is strongly connected for all time t , the agents will asymptotically reach an average consensus with convergence rate greater than or equal to $\lambda_2^* = \min_t \lambda_2(G(t))$, where $\lambda_2(G(t))$ represents the algebraic connectivity value associated with the Laplacian for graph $G(t)$. Likewise, in time-delayed networks, where messages are received after a delay τ instead of instantaneously, the dynamics become

$$\dot{\mathbf{x}}(t) = -\mathcal{L}\mathbf{x}(t - \tau) \quad (66.14)$$

and global exponential stability to the average-consensus equilibrium can be guaranteed for delays within the range $\tau < \pi/2\lambda_n$. Note that here the last eigenvalue of the Laplacian, λ_n , provides bounds on the maximum allowable delay τ , and thus networks with lower λ_n are more robust to time delays. Recall from Sect. 66.3.3 that $\lambda_n < 2\Delta$, favoring network topologies with lower maximum degree Δ , such as random graphs and ad hoc networks (Fig. 66.2b), instead of networks with large hubs (Fig. 66.2a) (Olfati-Saber et al. 2007).

The remainder of this section describes how control and decision-making algorithms employ consensus to achieve common goals for teams of UAVs and the associated challenges and limitations. In particular, the stability and convergence properties described by the network Laplacian affect the performance of these planning and control algorithms as well, in very similar ways to the effects on consensus discussed here. As such, many multi-vehicle approaches explore strategies to directly control the network topology in order to increase mission performance (Ibrahim et al. 2007, 2009; Yan and Mostofi 2010; Michael et al. 2009; Zavlanos et al. 2011; Ponda et al. 2011). Sections 66.5 and 66.6 describe some of these methods and associated challenges in further detail.

66.4.2 Effects on Cooperative Control

The feedback control process uses sensed feedback information with a model of the system dynamics to select a control input. If feedback information is delayed or corrupted due to communication degradations, control decisions may be flawed. For instance, in military MQ-1 Predator procedures, a pilot remotely controls the UAV manually using live forward-looking video on the nose of the aircraft as reference. Delays in video feedback and control commands from link latencies have resulted in pilot-induced oscillations (PIO), an instability from poor feedback control which has resulted in numerous aircraft mishaps (Defense Research and Development Canada 2010). The same phenomenon can occur in autonomously controlled cooperative teams of UAVs.

In general, network controlled systems (NCS) require a proper balance of control and feedback update rates: too slow can prevent the system from being controlled sufficiently well, but too fast can overload the communication network (Branicky et al. 2000) or excite system dynamics resulting in instabilities (Schwager et al. 2011). The stability of the system also depends on control implementation and feedback filtering techniques, as described in detail in Fax and Murray (2004). For networked multi-agent systems, it is helpful to define the control graph G^{ctrl} , similar but separate from the communication graph G^{comm} introduced in Sect. 66.3 as G (Schwager et al. 2011). G^{ctrl} is a directed graph that defines control dependencies (Fig. 66.4). In “leader-follower” formations, each vehicle selects control inputs based on the dynamics of the vehicle(s) directly preceding it (Fig. 66.4a). Tracking errors can propagate down the formation and under certain conditions exceed control authority, leading to instability, even with no delay in information sharing (Al-Karaki and Kamal 2004). In “string” formations studied in the context of an Automated Highway System, each vehicle communicates with the vehicle directly ahead and must follow it by a prescribed safe distance (Liu et al. 2001). Agents are slotted to communicate at separate times, which propagates

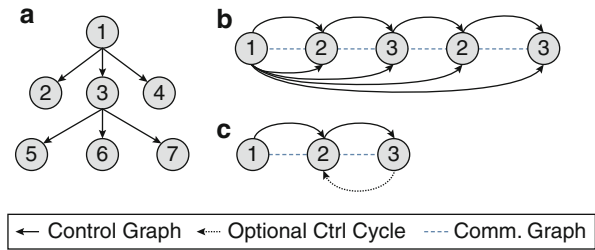


Fig. 66.4 Control and communication graphs. (a) represents a “leader-follower” formation free of network delays as discussed in Al-Karaki and Kamal (2004). (b) is the configuration of the Automated Highway System in Liu et al. (2001). (c) illustrates the graph in the three UAV formation flight test in Michael et al. (2010) (Figure (c) copyright Michael et al. (2010))

an information delay down the chain. Each vehicle bases its control inputs on the dynamics of the vehicle ahead and of it and on the lead vehicle dynamics (Fig. 66.4b). In this structure, even infinitely small communication delays cause string instabilities if agents compensate for error immediately. The formation can, however, be stabilized given even greater communication delays if the vehicles delay their control response to all apply it in near unison.

In Schwager et al. (2011) and Michael et al. (2010), the effects of network delays on formation stability of a team of decentralized UAVs are explored in indoor flight testing. Vehicles can communicate only with their neighbors and are similarly slotted to broadcast knowledge of vehicle states at different times. Agents store outdated team member states x_j and update them as more recent information is relayed to them. Control inputs $u_i(t)$ in Eq. (66.15) for each UAV are based on this information:

$$\begin{aligned} u_i(t) &= \sum_{j \in N_i^{\text{ctrl}}} K_{ij} (\hat{y}_{ij}(t) - y_i(t) - \delta_{ij}) \\ \hat{y}_{ii}(t) &= C_i x_i(t) \\ \hat{y}_{ij}(t) &= C_j x_j(t - \tau_{ij}T) \end{aligned} \quad (66.15)$$

where G^{ctrl} specifies the set of neighbors N_i^{ctrl} , T is the overall time cycle of communications (for all agents to transmit once), τ_{ij} is the number of communication hops from j to i , δ_{ij} is the desired interagent distance, and K_{ij} is a control gain. To derive stability bounds on the system, the authors rewrite the system to be in continuous time at T second intervals and couple the entire system of agents as (fully defined in Schwager et al. (2011)):

$$X(t + T) = AX(t) + \Delta \quad (66.16)$$

The structure of matrix A depends on G^{ctrl} and G^{comm} , and its eigenvalues λ_i are a function of K_{ij} and T . The convergence rate of the formation is then directly related to $\alpha = |\lambda_{\max}(A)|^{1/T}$, where $\alpha \leq 1$ indicates Lyapunov stability and $\alpha < 1$ asymptotic stability. The indoor flight experiment consists of three UAVs with control and communication graphs in Fig. 66.4c and second-order control as in Eq. (66.17). Results show the following: (1) the formation is stable with small gains and converges faster with faster network cycles T , (2) at higher control gains, decreasing network cycle time may actually result in slower converge due to unstable dynamics being excited, and (3) introducing feedback cycles in G^{ctrl} can also introduce instabilities.

Communication delays can also be problematic in multi-UAV path planning for collision avoidance. In an outdoor flight test in Kopeikin et al. (2012), decentralized vehicles continuously replan the next steps of their receding horizon paths (not in formation). UAVs deconflict by not planning crossing paths to go

around each other. Short network delays and rapidly changing dynamics cause the vehicles to sometimes plan using outdated information. On occasion, two UAVs approaching each other head on may alternate in replanning deconflicting routes left or right multiple times. This phenomenon, known as “churning,” is similar to the common awkward situation where two people mutually block each other in a hallway or sidewalk, and causes delays, hazards, and system performance degradation.

66.4.3 Effects on Cooperative Decision Making

In order for multiple UAVs to cooperate in a mission, the team as a whole must plan and make decisions collectively. The objective in cooperative decision making is therefore for the networked team to reach an agreement on the tasks, plans, and actions required to execute the mission. Different strategies exist to achieve *conflict-free* plans, which require information exchange through the network. The dynamics and uncertainties of the communication network have several implications on decision-making strategies: (1) bandwidth limitations can restrict the content and frequency of planning messages, (2) even small network delays can cause agents to plan asynchronously making consensus difficult to achieve, and (3) network dropouts may prevent agents from participating in team decision making which can impact plan execution.

Because network bandwidth is a limited resource, cooperative decision-making methods need to limit the content of planning messages, efficiently select which pieces of information need to be shared with which members, and overall reach consensus with as few messages as possible to mitigate delays and conflicts. If the system operates under centralized control, every UAV needs to communicate every required bit of situational awareness information to a central leader (e.g., the base station). This strategy minimizes the number of messages required to reach consensus, but may require large amounts of low-level planning data to be communicated to the base, and can cause the system to be slow to react in a rapidly changing environment due to network delays. To overcome this, decentralized planning strategies provide individual agents with greater decision-making capability, where information can be shared more selectively and agents can locally decide on courses of action based on their own situational awareness, thus improving reaction time. Now, however, consensus on plans must be explicitly reached by exchanging messages over the network. A common distributed strategy, referred to as *implicit coordination*, consists of agents independently solving the entire centralized planning problem for all agents and carrying out actions assigned to them. This requires agents to reach consensus on individual situational awareness and planning objectives prior to planning (Ren and Beard 2005; Olfati-Saber and Murray 2004; Alighanbari et al. 2006), which may require large exchanges of data across the network, especially if inconsistencies exist (Alighanbari and How 2005). Another popular planning method

involves using *distributed auction algorithms*, where agents individually plan for themselves and consensus is reached through an auction mechanism (Sariel and Balch 2005; Ahmed et al. 2005; Atkinson 2004). A particular distributed auction approach that performs consensus on agent plans rather than on information is the Consensus-Based Bundle Algorithm (CBBA) (Choi et al. 2009). This algorithm employs a consensus protocol that explicitly specifies rules for reaching agreements on tasks, thus guaranteeing a conflict-free solution despite possible inconsistencies in situational awareness. In addition to being robust to inconsistent information, performing consensus in the task space is usually significantly faster, requiring fewer messages between the agents (Choi et al. 2009; Johnson et al. 2011).

Another challenge that affects distributed planning algorithms is that they usually rely on synchronous message passing in order to guarantee convergence. This is difficult to enforce in dynamic environments where the number of tasks or agents in the network may be changing and the network is subjected to the uncertainties and dynamics described in the previous section. When messages are asynchronous, time delays are not usually constant and messages may be received out of order, which impacts typical consensus algorithms since old information may be processed as new, thus hindering consensus. Being robust to these effects requires developing new consensus protocols or plan deconfliction rules that can recognize and process out of order messages. For instance, an asynchronous extension to CBBA addresses these issues in the decentralized task planning domain (Johnson et al. 2011), and several algorithms have been developed to perform linear asynchronous consensus in the information domain (parameter consensus) (Ren et al. 2005; Olfati-Saber et al. 2007; Blondel et al. 2005; Bemporad et al. 2010). This however remains a general problem for most multi-UAV systems of interest.

Finally, rapid link degradation or specific mission requirements (e.g., sending a reconnaissance UAV on a long duration mission at a location far from the base) may cause the network to disconnect. Most current distributed tasking algorithms, however, require continued connectivity in order to converge, and literature on limited communication environments has focused on maintaining this network connectivity through relay teams (Nguyen et al. 2003; Ibrahim et al. 2007; Ponda et al. 2011) by optimally placing vehicles (Dixon and Frew 2009) or by restricting allowable tasks. In realistic mission scenarios, where communication links are broken and created in real time and teams of agents often reconfigure themselves dynamically into different subnetwork structures, it is desirable to have distributed tasking algorithms that can ensure conflict-free solutions even when the network is disconnected. A recent study has considered distributed methods to locally adapt the task planning space within each subnetwork of agents, thus ensuring conflict-free assignments in the presence of network disconnects (Ponda et al. 2010). The approach in Ponda et al. (2010) results in assignments that would be equivalent to a mission operations center subdividing task sets among the different subnetworks, but without requiring the centralized overhead.

66.5 Motion Planning for Communications

Envisioned multi-UAV system applications involve controlling the agents to achieve a defined primary mission objective. This commonly includes maximizing area coverage, tracking a target, rendezvous in space, or formation and flocking control (Razafindralambo and Simplot-Ryl 2011; Goddemeier et al. 2011; Ghaffarkhah and Mostofi 2009; Zavlanos et al. 2011). Oftentimes, the primary objective does not specifically define connectivity objectives, but implicitly requires network communications control for the system to operate effectively. As described in the previous section, network performance depends on the quality of wireless links, the number of interconnected neighbors, and the ability to route information based on the topology. A common way to control network communications in a team of multiple UAVs is to plan the motion of the vehicles in such a way that the resulting topology supports the exchange of data required by the system. The general problem then becomes controlling a set of $N = \{1, \dots, n\}$ team members to achieve the mission objective and maintain the required state of connectivity. For mathematical simplification, vehicles are often considered to be point nodes with first-order dynamics, $\dot{x}_i(t) = u(t)$, where $u(t)$ is the control input at time t and $x_i(t)$ is the position of agent i . Some formulations go further and extend control to second-order dynamics as in Eq. (66.17) (Zavlanos et al. 2011) where v is velocity,

$$\dot{x}_i(t) = v_i(t) \quad \text{and} \quad \dot{v}_i(t) = u(t). \quad (66.17)$$

whereas others include greater detail in modeling actual vehicle kinematics (Dixon and Frew 2007). Network communication requirements vary from maintaining connectivity with a specific number of neighbors to establishing routes with sufficient throughput to send data to a base station. One of the earliest studies where motion control was employed to maintain connectivity consisted of a team of distributed robots with the task of meeting at an undefined point (rendezvous problem). The strategy involved each robot independently moving a step amount toward the geometric center of the area outlined by its perceived neighbors (Ando et al. 1999). This process was repeated iteratively until robots converged on a central location. Since, methods have expanded to include graph theoretic techniques, continuous control through potential fields, reactive control strategies, and adaptive techniques based on learning from the environment. Many of the efforts described here were focused on ground robotic systems but are relevant to aerial applications as well.

66.5.1 Graph Theoretic Methods and Potential Fields

Section 66.3 introduced the network graph G and the *algebraic connectivity* metric λ_2 , the second eigenvalue of the graph Laplacian \mathcal{L} . Several studies have formulated \mathcal{L} by assigning weights $w \in [0, 1]$ to each interagent link based on relative distance, with bounds $w_{ij} = 1$ if $d_{ij} \leq d_o$ where the link is strong and $w_{ij} = 0$ if

$d_{ij} \geq d_{\max}$ where the signal is considered too weak to connect Zavlanos et al. (2011). Connectivity objectives in multi-agent control problems often consist of meeting some threshold λ_2 or even maximizing its value. Initially, discrete methods were proposed to maximize λ_2 using iterative control inputs for each agent using greedy algorithms in both centralized (Kim and Mesbahi 2005) and decentralized frameworks (Gennaro and Jadbabaie 2006). These ensure that the team drives to connected configurations throughout the mission. Continuous control methods to maximize λ_2 using properties of \mathcal{L} were then formulated in Zavlanos and Pappas (2007) using potential fields in a centralized framework. Potential fields, which are commonly used in multi-vehicle control applications, are virtual energy fields that lead to attractive or repelling control inputs formulated as

$$u_i(t) = -k \nabla_i \phi_i - \sum_{j \neq i} \nabla_i \psi_{ij} \quad (66.18)$$

$$\psi_{ij} = \frac{1}{\|x_{ij}\|^2} + \frac{1}{d_{\max}^2 - \|x_{ij}\|^2} \quad (66.19)$$

One benefit of this formulation is that it allows multiple, sometimes competing, control objectives to be considered (Fig. 66.5). Here ψ_{ij} is designed according to the desired connectivity dynamics between i and j , as illustrated by the simple function in Eq. (66.19) which keeps the nodes within communication range and prevents them from colliding (Michael et al. 2009). ϕ_i can be any other steering function, used, for example, for obstacle avoidance or target tracking (Hsieh et al. 2008) as implemented in Fig. 66.5 (obtained from Goddemeier et al. (2011)). As the system approaches an unwanted state, such as a vehicle becoming disconnected or dangerously approaching an obstacle, that effect dominates the effective force generated by the potential field to correct the situation.

A decentralized version of this framework was proposed in Yang et al. (2010) using a power iteration algorithm for each agent to compute its local λ_2 using information from its neighbors and then estimate the global λ_2 to generate potential fields. Potential fields have also been used to control multi-vehicle systems for other connectivity requirements, using both graph theory and considering realistic channel dynamics and measurements. For instance, Ajorlou et al. (2010) and Dimarogonas and Johansson (2010) used bounded forces to minimize distance with connected neighbors for a team to rendezvous or assemble in a formation. One study considered Rician fading in wireless links to control multiple robots to track a target and maintain connectivity with the base (Ghaffarkhah and Mostofi 2009). A primal-dual optimization algorithm was used to control agents through a deployment objective while subjected to stochastic fading effects (Ny et al. 2011). Finally, in Goddemeier et al. (2011) potential fields controlled a team of eight UAVs to form a moving aerial communication network. Forces were created based on the deployment objective of maximizing area coverage over a moving point, and connectivity requirements which included maintaining a prescribed SNR over each link subject to realistic fading, and staying connected with a specified number of other UAVs.

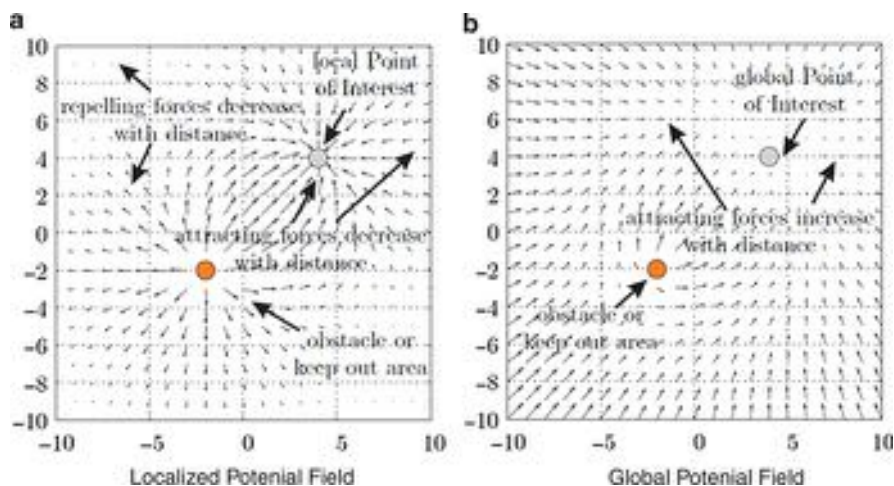


Fig. 66.5 Potential fields for motion. (a) shows a set of potential fields which dominate as vehicles approach the objects defined, and (b) represents potential fields dominating when further away to generally guide the vehicles (Figure copyright Goddemeier et al. (2011))

66.5.2 Reactive and Adaptive Techniques

Instead of controlling motion for connectivity at every time, another strategy is to allow agents to navigate freely according to mission objectives and take action only when connectivity is threatened. One method involves each robot iteratively planning how far it can travel at every step before losing connectivity with a required minimum number of neighbors (Razafindralambo and Simplot-Ryl 2011). Other studies were motivated by the difficulty of predicting wireless link performance with enough accuracy to make control decisions in a realistic environment. In these cases, each robot travels according to deployment objectives (using potential fields or random motion) until the measured signal strength with a set number k of neighbors drops below a designated threshold. At this point, the robot is commanded to halt (Reich et al. 2008) or even backtrack toward its nearest neighbor (Hsieh et al. 2008) and can only resume deployment objectives if communication requirements are again met. In these strategies, the required number of neighbors k is part of the design trade. A higher value increases network robustness to unpredicted disconnects but constrains the configuration of agents to be tighter and therefore cover less surface area (Reich et al. 2008). When warranted, the team may benefit from two agents disconnecting to add flexibility to the formation (so long as the graph does not disconnect). For such cases, the study in Zavlanos and Pappas (2008), which is explored in greater detail in the next subsection, employs a market-based algorithm for agents to come to consensus on which link to remove to meet formation objectives. Finally, certain scenarios involve agents taking on tasks beyond the reach of communications where the only option is to temporarily disconnect the network. Authors in Hollinger and Singh (2010) therefore define an implicit coordination

approach which is used to find optimal agent paths to execute the mission but reach a point of full connectivity at some specific time interval.

Many recent robotic efforts have turned to adaptive and learning techniques to adjust their behavior and overcome unpredicted changes in the environment (Leonard et al. 2010). Because it is so difficult to predict actual wireless channel performance in dynamic environments, several approaches have used measurements in the field to adapt the motion planning strategy. One basic approach is to take measurements of the environment a priori and construct a “radio map” used to plan constraints on relative node positions to stay connected (Hsieh et al. 2008). However, it may not be practical to perform this type of pre-mission survey as access to the location may be denied, and the environment may change over time due to different obstacles or other emitting devices. For this reason, several studies propose probabilistic methods based on the spatial correlation of channel fading to select locations to transmit. For instance, in Lindhe and Johansson (2010) a single mobile agent tracks a moving target and must send collected data to a fixed base. Current measures of the SNR are used with link dynamics previously described to adaptively schedule stops where the channel is predicted to be strong to transmit information rather than waste energy transmitting during fades. Bounds on the dynamics are derived to ensure system stability and prevent the agent from falling behind the target. In a similar scenario, a single agent must repeatedly visit multiple areas and send information to the base. Here the planner sequentially solves for the optimal visit path using a Mixed-Integer Linear Program (MILP) and budgets the power needed to transmit given the environment throughout and the path using a nonlinear program (Ghaffarkhah et al. 2011) for energy conservation. Finally in Mostofi (2009), a team of multiple UAVs learn link dynamics online to plan motion cooperatively and track a target. At each step, the vehicles seek to gain information $\mathcal{I}_i[k]$:

$$\mathcal{I}_i[k] = \mathcal{I}_i^s[k] + \sum_{j \neq i} \mathcal{I}_j^c[k], \quad (66.20)$$

about the target either through direct sensing $\mathcal{I}_i^s[k]$ (which degrades with distance) or by exchanging information with other agents $\mathcal{I}_j^c[k]$ (which degrades with poor communications). The measured SNR is used to learn the communication environment and properly balance sensing and communication objectives. Furthermore, channel fades correlated over multiple steps are used to identify areas of shadowing, and adapt the motion strategy to improve communications.

66.5.3 Implementation Examples

Motion planning to control network communications is a relatively new research topic in robotics, and as such, there have only been a few implementations of these principles using actual vehicles. Most of the studies described previously in this section were conducted in simulation. The following discusses two experiments that are particularly relevant because they incorporate several of the motion planning methods discussed previously.

In Reich et al. (2008), four UGVs are each tasked to a designated location, and the mission objective is to transmit imagery back to the base, using multiple hops if needed. A radio map is constructed by measuring the signal strength between many pairs of locations in the area, and it is employed in assigning tasks to the robots so that a connected multi-hop path exists to the base. During the deployment, a centralized scheme uses potential fields to navigate the robots to their targets. A reactive controller based on signal strength and throughput measurements halts individual robots if connectivity thresholds are violated. The experiment successfully shows the ability to control robots using the scheme. It also highlights that signal strengths and throughputs vary significantly during the mission with previous static measurements due to fading and multi-user interference. Results support the choice to implement a reactive controller.

In Michael et al. (2009), potential fields are implemented in a decentralized framework on a team of seven UGVs (Fig. 66.6). The objective is for each robot to maintain distance-based connectivity with k neighbors while a leader robot is given navigation commands. When flexibility is needed in the formation, links are switched off from the potential field using a market-based consensus algorithm. In this process, robots place bids on their link most at risk with bid value based on their number of neighbors. The bids are propagated through the network, and the highest bid link is allowed to be deleted. The framework only allows one link to be deleted at a time to ensure that the graph stays connected, and the approach introduces methods to handle asynchronous planning issues (see Sect. 66.4). In this effort, robots compute their own local estimate of the network and exchange position and bid information using Transmission Control Protocol (TCP) messages only with their neighbors.

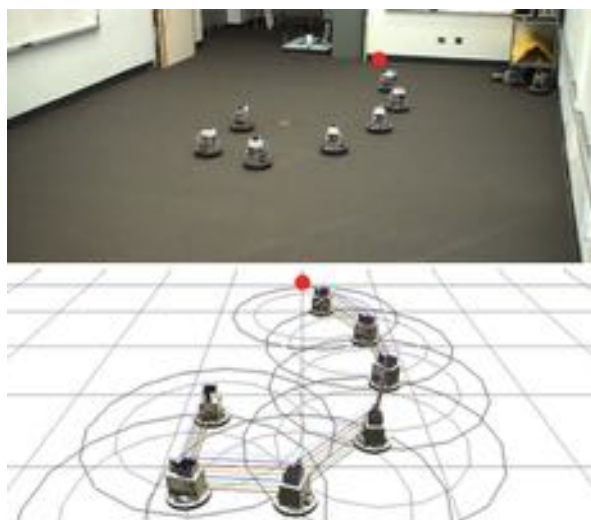


Fig. 66.6 Experiment with seven UGVs maintaining connectivity through motion control. Each *colored line* in the visualization depicts the current connectivity graph estimate for a particular robot (Figure and caption copyright Michael et al. (2009))

66.6 Communication Relays for Network Support

Because of their rapid and flexible deployment capabilities, persistence, ability to fly above obstacles, and relative low cost, UAVs have generated significant interest to serve as communication relays. Relays can be used to extend the range of a network, add capacity for increased throughput, and increase the robustness of the system to failure. Large-scale military UAVs are already delivering wide-area communication connectivity to ground troops in mountainous areas (Jonson et al. 2008), and similar smaller tactical concepts have been proposed for disaster area recovery (Federal Communications Commission 2011), and wild land fire fighting (Zajkowski et al. 2006) and to connect scattered wireless sensors and mobile ad hoc networks (Palat et al. 2005). When deploying UAVs to serve as communication relays, their primary mission objective is to maximize connectivity. Many of the communication control methods described in the previous section apply to this problem, but here the underlying network to be supported cannot usually be controlled and may be dynamic. Control objectives typically involve guiding relay agents to optimized locations to support data exchange requirements at the lowest resource cost possible. Two classes of problems are provided in this section. First, methods are investigated to optimize the control of a chain of relays linking two mobile end nodes. Second, the problem of optimal relay deployment to support a larger, typically fixed network is considered. In all cases, controlling the communication network involves considering the dynamics of wireless links, the topology, and information routing options.

66.6.1 Controlling Relay Chains

Several applications have called for a chain of relay nodes to extend communications through multiple hops from one end of the chain to the other. In this case, once the chain forms, the relationship between pairs of nodes remains constant (same two neighbors), and information routing is known since it simply follows the nodes in the chain by default. A common application under which this is studied involves maintaining communications between a moving agent and its base station. This problem has been examined in complex obstructed environments where the objective is to optimally navigate and position N relays to establish multi-hop line of sight with the base. Optimal deployment paths can be generated in a centralized framework using a Mixed-Integer Linear Program (Schouwenaars et al. 2006) or a Gauss pseudospectral solver initialized using a rapidly-exploring random tree (RRT) solution (Kulling 2009). A suboptimal decentralized receding horizon planner can also be used to provide faster real-time control to repeatedly “redeploy” relays as the mission progresses (Kulling 2009). In a similar problem, a chain of UGV relays must connect a mobile node to a base and optimize their formation to minimize the end-to-end *bit error rate* (BER) (see Sect. 66.3 and Eq. (66.7)). Here, wireless links are subjected to path loss, shadowing, and multipath fading

(Yan and Mostofi 2010; Bezzo and Fierro 2011). The resulting optimization can be written as

$$\max \mathcal{J}(x) = \sum_{i=2}^N \ln(1 - \text{BER}_{i-1,i}) \quad (66.21)$$

which is reformulated as a set of decentralized potential fields used for vehicle control as in Eq. (66.19). In Yan and Mostofi (2010), the formulation does not assume deterministic link dynamics, since actual channel qualities can vary significantly with predicted models. It instead employs a probabilistic framework with least squares regression to estimate the value of the path loss exponent α in Eq. (66.2) and shadowing variance and correlation in Eq. (66.4) from several SNR measurements. These estimates are constantly updated to plan motion to optimize the chain BER. Since multipath fading is assumed to be spatially uncorrelated and varies quickly over small distances, once the relay reaches its destination, it “jitters” around its position to further reduce the quantity $\text{BER}_{i-1,i}$. With a slight modification to the problem formulation, the objective is changed to provide a chain configuration which meets a threshold end-to-end BER but minimizes relay energy costs. This allows motion cost and transmission costs (through variable power transmission) to be traded off in the optimization, with tunable weights associated with each to favor one over the other (Yan and Mostofi 2010).

Again, due to the recency of these studies, most of the above efforts were conducted in simulation or controlled hardware experiments with simplifying assumptions. However, authors in Dixon and Frew (2007) and Dixon (2010) tackle the problem of forming a chain of relays using actual fixed wing UAVs flying over a mountainous terrain. In this study, the objective is to maximize the end-to-end *capacity* c of the chain found from Eq. (66.6) or in other words maximize the minimum link capacity in the chain (Fig. 66.7). Two complicating but realistic factors in this work include: (1) fixed wing aircraft have velocity limits $0 < v_{\min} < v < v_{\max}$ and must loiter in orbits also constrained by ϕ_{\max} bank angle, and (2) the authors chose to control the chain based on SINR signal strength measurements only and avoid using channel models deemed inaccurate in dynamic environments. Because of the first challenge, vehicle dynamics are modeled as being nonholonomic with bicycle kinematics:

$$\dot{x}_i = v_i \cos \psi_i, \quad \dot{y}_i = v_i \sin \psi_i, \quad \text{and} \quad \dot{\psi}_i = v_i s_i \quad (66.22)$$

where $s_i = \frac{g \tan \phi}{v_i^2}$ is the path curvature and ψ_i is the heading angle. Consequently control inputs take on the form $u_i = v_i s_i$. Since each relay is controlled based on its SINR measurements only, an extremal seeking algorithm is implemented where each UAV hones in on the heading which maximizes γ with its neighbors using

$$\theta^*(t) = \operatorname{argmax} \left(\frac{1}{\min\{\gamma_{i,i-1}, \gamma_{i-1,i}, \gamma_{i,i+1}, \gamma_{i+1,i}\}} \right). \quad (66.23)$$

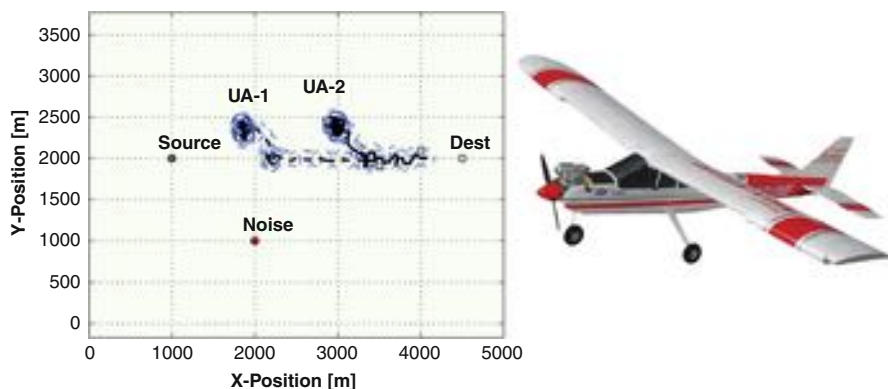


Fig. 66.7 Flight testing a chain of UAV relays. The evolution of a simulation with two UAV relays (*left*) and the type of fixed wing aircraft used in the experiment (*right*) (Figure copyright Dixon (2010))

This framework does not assume equal interference levels for neighboring UAVs, and as such, the SINR measurements are asymmetric on each side of the link. The optimization therefore maximizes the minimum link capacity on either side of each link. Then, a Lyapunov Guidance Vector Field (LGVF) (Dixon and Frew 2007) drives the UAVs to an orbital pattern about a center point θ_i , and a consensus algorithm is used to synchronize the UAVs on a common orbiting phase angle. Flight test results demonstrate the ability to control an aircraft to climb the SINR gradient and improve capacity using signal strength measurements only. However, RF fluctuations encountered are detrimental to the chaining algorithm and at times cause the measured γ to actually increase while flying away from a neighbor. This exemplifies the complexity in controlling the communication network with operational dynamics using feedback measurements alone.

66.6.2 Deploying Relays to Support a Network

So far this section has explored problems involving the control of a chain of relays between two designated points. In a number of other applications, however, UAVs (or other mobile agents) serving as relays can be deployed to provide general connectivity support to an existing network consisting of many nodes. The problem here is to determine how best to deploy each relay and which nodes they should form links with to satisfy the data requirements of the network. Objectives here include maximizing the performance of the supported network or minimizing the number of relays deployed to meet communication thresholds. These problems are often studied in the context of UAVs relaying information from poorly connected ground nodes or for wireless sensor network (WSN) nodes with connectivity and energy limitations sending data to a base station.

Network graph theoretic processes described in Sect. 66.3 have also been explored in these problems. One example consists of maximizing the lifetime of a fixed WSN collecting data which depletes its battery over time as the energy is used to transmit information (Ibrahim et al. 2007, 2009). Here K relays are available to be deployed and redeployed to support the network. The graph Laplacian \mathcal{L} in Eq. (66.8) is composed of weights w_{ij} in its adjacency matrix A based on the transmitter power needed to maintain a threshold BER across the link. This formulation assumes all starting nodes have equal battery life, and therefore the algebraic connectivity of the graph λ_2 can be used as a metric of remaining network life. A centralized semi-definite programming (SDP) algorithm optimizes the placement of the K relays to maximize λ_2 over the network. It discretizes the area into cells, finds the best combination of placements for the K relays at the center of these cells, and then rediscrretizes selected cells to refine relay placement over several iterations. From there, a minimum cost path routing algorithm (see Sect. 66.3) is implemented where relays are encoded with reduced cost to incentivize their use. The deployment process is repeated as needed as other nodes deplete. The formulation is also modified to solve for the minimum number of additional relays required to support the network at a threshold level (Ibrahim et al. 2009).

In a similar problem, WSN nodes compute their k -redundancy in a distributed framework using neighbor discovery messages (Atay and Bayazit 2008). The k_i^{red} is the number of links that need to be removed to disconnect any two neighbors of i and is a measure of the importance of that node in global connectivity. Here if any k_i^{red} falls below some threshold because of a change in topology, the network requests a relay which is positioned using several different optimization schemes to meet connectivity requirements. Finally, the study in Zhu et al. (2009) discusses how the deployment strategy changes for one UAV relay to support an ad hoc network according to different connectivity objectives. Optimization methods are presented to maximize (1) global and (2) worst-case connectivity based on properties of the minimum spanning tree (MST) graph (Bertsimas and Tsitsiklis 1997) or (3) minimize network bisection probability and (4) maximize the Fiedler value using graph Laplacian \mathcal{L} . Each objective results in a different UAV relay deployment plan.

In the previous three examples, the deployed relays essentially became nodes in the supported ad hoc network, thus changing the topology and information routing options. However, another strategy is to assume that relays are similar to gateways and have a separate strong communication network to connect to other relays and the rest of the world. These can then be used as a communication backbone to support a disconnected ad hoc network (as illustrated with the “UAV Comm. Relay” in Fig. 66.1). The network then takes on a form more consistent with a cellular network architecture, where relays serve as mobile base stations and connect to nodes in their area of responsibility. This is how current military large-scale UAV communication relays operate (Jonson et al. 2008). Mobile backbone network (MBN) optimization seeks to simultaneously position and assign K MBN relays over a network in order to (1) maximize the minimum throughput in the network (Srinivas and Modiano 2008), (2) maximize the aggregate network throughput

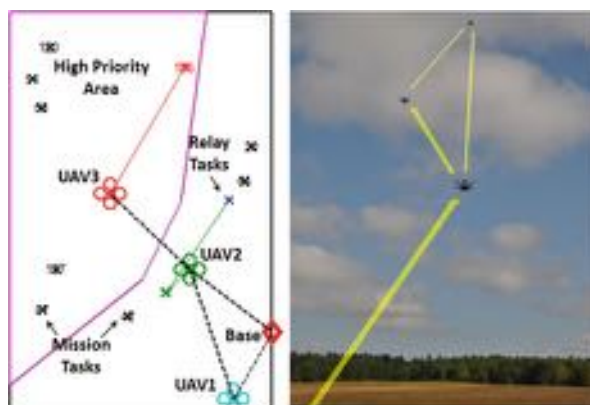
(Srinivas and Modiano 2008), (3) maximize the number of nodes meeting a threshold throughput (Craparo and How 2010), and (4) optimize relay deployment and node positioning (if those nodes can be controlled) for objective (3) (Craparo and How 2010). In objectives (3) and (4), the optimization is solved using a Mixed-Integer Linear Program (MILP) for a network optimization formulation (Bertsimas and Tsitsiklis 1997). Similar to other previous studies, the optimal solver scales poorly with increasing numbers of nodes and relays, and a suboptimal greedy solution based on solving the Max-Flow problem (Bertsimas and Tsitsiklis 1997) is provided in parallel for fast and provably good solutions.

66.6.3 Controlling Communications Through Task Allocation

A similar, yet distinct method to control network communications in a team of UAVs is through task allocation. Here, the system consists of a multi-agent team with a set of objectives, and similar to motion planning problems discussed in Sect. 66.5, connectivity is an implicitly derived requirement. The framework can use task allocation to ensure agents only take on tasks which result in a suitable network topology, or better yet, allocate additional tasks to certain agents to serve as communication relays as needed (Ponda et al. 2011). The challenges now include: (1) determining which tasks should be assigned to which agents, (2) identifying when the network topology will not support the assignment, (3) designating which agents to take away from executing mission tasks to assign as relays and where to position them, and (4) adapting to dynamic scenarios and uncertainty. These frameworks follow the same general principles of multi-UAV mission planning methods addressed in chapter ▶[Cooperative Mission Planning for Multi-UAV Teams](#).

The authors in Tardioli et al. (2010) discuss a task allocation-based network control mechanism. A set of N UGVs have a set of \mathcal{J} possible tasks to accomplish and must maintain a threshold receive signal strength (RSS), through multiple hops if needed, to a base. A centralized task allocation planner initializes by dividing the team of robots into subset clusters. Each cluster is assigned a common task to accomplish. Then, motion planning potential fields similar to Eq. (66.19) are used to attract the cluster to the goal, to the base, and to maintain connectivity within the cluster while accomplishing the task. If the cluster cannot achieve the goal and stay connected, it is merged with a neighboring cluster to expand the multi-hop capability. Clusters can monotonically grow in size until the entire team is included in one large cluster. Unachievable tasks are then discarded. At every planning cycle, clusters are reset to their original size to again allow more flexibility in task assignment. Different optimization strategies can be implemented to select which task gets assigned to each cluster at every step (Tardioli et al. 2010). The framework is implemented in an outdoor hardware system experiment with three small UGVs. Results, while successful, again highlight the difficulty in predicting link quality due to rapid variations and uncertainty in the dynamics.

Fig. 66.8 Flight testing a task allocation framework which maintains connectivity. On the *left*, UAV2 relays UAV3 as it executes a task, and both plan to implement a similar strategy for UAV3 to accomplish the task in *red* at the top of the image



A different task allocation strategy is proposed in Ponda et al. (2011) and Kopeikin et al. (2011). A decentralized task allocation planner is used to coordinate a team of N UAVs in maximizing reward in the mission by taking on high-value tasks while maintaining distance-based connectivity requirements. The scenario is dynamic, similar to realistic operations, and tasks are created at random times and locations throughout the mission and have a limited time window of validity. Because the assignment problem is combinatorial and difficult to solve optimally, an auction-based multi-agent multi-task assignment algorithm (Consensus-Based Bundle Algorithm) allows agents to place bids on each task (Choi et al. 2009). Bids are based on task values, agent capabilities, and availability for execution. Agents then share bids, assign winners, and iterate through the algorithm to reach consensus on a conflict-free plan. The framework then predicts the network topology for the execution of each task assigned, identifies disconnected configurations, and creates new relay tasks to repair the network. Underutilized UAVs can then bid on these relay tasks similar to any other. Two challenges in this planning framework discussed in Ponda et al. (2011) involve (1) properly incentivizing agents to take on the relay tasks which by themselves add no reward to the mission and (2) ensuring the algorithm converges quickly to mitigate planning delays. This strategy does not plan for connectivity through motion as UAVs travel to and from tasks, and instead ensures the network is connected during task execution only. The system was implemented in an outdoor flight test with a team of three UAVs to demonstrate initial functionality (Fig. 66.8) and is being modified to account for more complex wireless connectivity models (Kopeikin et al. 2011).

66.7 Concluding Remarks and Remaining Challenges

Teams of multiple UAVs show tremendous potential in a growing number of military, first response, exploration, and other applications. As technology advances, these cooperative systems will make their way to the operational domain. To realize

this, the research community is involved in a wide range of studies including aircraft design, control, cooperative mission planning, sensing, and exploitation, among many others. Effective and robust control of the communication network will be a near universal requirement for these systems.

This chapter presented several topics on controlling the network for a team of UAVs to ensure command and control messages are exchanged properly, and remotely sensed data is transmitted back to desired locations. Control methods discussed include motion planning to create an effective network topology, transmitter power regulation, deployment and control of mobile communication relay nodes, and task allocation algorithms to ensure connectivity. While progress is ongoing in all of these areas, there are nevertheless several high-level challenges which need to be considered:

1. *Robustness to uncertainty*: Because of the recent surge in wireless devices for every day applications, the electromagnetic spectrum is becoming more crowded and communication systems are competing for bandwidth. Furthermore, as described earlier, it is often difficult to accurately predict wireless channel performance in dynamic environments. The need to control and implement multi-vehicle systems to be robust to uncertainties and interference, and degrade gracefully when the network underperforms, is critical (Goldsmith 2005).
2. *Limiting bandwidth usage*: Each bit transmitted costs valuable resources in both bandwidth and energy. In order to maximize resources, all sensing and control functions must minimize the amount of information to transmit. Consensus in cooperative control and decision making must converge accurately across team members with as few messages as possible. Methods to process and sort sensed data onboard the UAV to minimize transfer requirements for analysis is also paramount.
3. *Ad hoc networking protocols*: Researchers in the cooperative control and ad hoc networking communities need to partner to jointly optimize control and routing strategies to maximize resources. Network protocols need to quickly adapt to the dynamics of multi-UAV networks, be flexible enough for the wide scope of envisioned applications, and ensure that the network is secure from unwanted users.

The communication network is one of many critical components of a multi-UAV system. Many of the methods presented in this chapter require a more in-depth understanding of other focused disciplines. Interested readers are encouraged to read other chapters in this handbook describing ►Multi-UAV Cooperative Mission Planning, ►Aircraft Flight Control, ►Trajectory Planning, and ►Advanced UAV Communication Systems. In addition, the references listed in this chapter contain detailed information on all of the topics covered. The following references are recommended for readers interested in specific topics: wireless communications and modeling (Goldsmith 2005; Mostofi et al. 2009), ad hoc topology control (Santi 2005), mesh networking standards (Hiertz et al. 2010), consensus in network graphs (Olfati-Saber et al. 2007), and graph theoretic vehicle control (Zavlanos et al. 2011).

References

- C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, The Optimised Routing Protocol for Mobile Ad-Hoc Networks: Protocol Specification – Report N5145, Tech. rep., Institut National de Recherche en Informatique et en Automatique, France, 2004
- A. Ahmed, A. Patel, T. Brown, M. Ham, M. Jang, G. Agha, Task assignment for a physical agent team via a dynamic forward/reverse auction mechanism, in *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, 2005
- N. Ahmed, S. Kanhere, S. Jha, Link characterization for aerial wireless sensor networks, in *IEEE Globecom*, Dec 2011
- A. Ajorlou, A. Momeni, A.G. Aghdam, A class of bounded distributed control strategies for connectivity preservation in multi-agent systems. *IEEE Trans. Autom. Control* **55**(12), 2828–2833 (2010)
- M. Alighanbari, J. How, Decentralized task assignment for unmanned aerial vehicles, in *IEEE Conference on Decision and Control (CDC)* (2005), pp. 5668–5673
- M. Alighanbari, L. Bertuccelli, J. How, A robust approach to the UAV task assignment problem, in *IEEE Conference on Decision and Control (CDC)* (2006), pp. 5935–5940
- J.N. Al-Karaki, A.E. Kamal, Routing techniques in wireless sensor networks: a survey. *IEEE Wirel. Commun.* **12**, 6–28 (2004)
- J.N. Al-Karaki, A.E. Kamal, Leader-to-formation stability. *IEEE Trans. Robot. Autom.* **20**(3), 443–455 (2004)
- S. Anderson, R. Simmons, D. Goldberg, Maintaining line of sight communications networks between planetary rovers, in *International Conference on Intelligent Robots and Systems*, Oct 2003
- H. Ando, Y. Oasa, I. Suzuki, M. Yamashita, Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robot. Autom.* **15**(5), 2418–2428 (1999)
- N. Atay, B. Bayazit, Mobile wireless sensor network connectivity repair with K-redundancy. *Workshop on the Algorithmic Foundations of Robotics*, Dec 2008
- M.L. Atkinson, Results analysis of using free market auctions to distribute control of UAVs, in *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, 2004
- A. Bhatia, P. Kaushik, A cluster based minimum battery cost AODV routing using multipath route for zigbee, in *IEEE Conference on Networks*, St. Louis, MO, Dec 2008
- A. Bemporad, M. Heemels, M. Johansson, *Networked Control Systems* (Springer, Berlin/Heidelberg, 2010)
- D. Bertsekas, R. Gallager, *Data Networks* (Prentice-Hall, Englewood Cliffs, 1992)
- D. Bertsimas, J.N. Tsitsiklis, *Introduction to Linear Optimization* (Athena Scientific, Belmont, 1997)
- N. Bezzo, R. Fierro, Decentralized connectivity and user localization via wireless robotic networks, in *IEEE Globecom*, Dec 2011
- V.D. Blondel, J.M. Hendrickx, A. Olshevsky, J.N. Tsitsiklis, Convergence in multiagent coordination, consensus, and flocking, in *Proceedings of the IEEE Conference on Decision and Control*, 2005
- M.S. Branicky, S.M. Phillips, W. Zhang, Stability of networked control systems: explicit analysis of delay, in *American Control Conference (ACC)*, 2000
- P. Chandler, M. Pachter, Hierarchical control for autonomous teams, in *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001
- P.R. Chandler, M. Pachter, S. Rasmussen, UAV cooperative control, in *American Control Conference (ACC)*, June 2001
- H. Choi, L. Brunet, J.P. How, consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robot.* **25**, 443–455 (2009)
- H.C. Christmann, E.N. Johnson, Design and implementation of a self-configuring ad-hoc network for unmanned aerial systems, in *AIAA Infotech@Aerospace Conference*, May 2007

- E. Craparo, How J.P., E.M., Throughput optimization in mobile backbone networks, in *IEEE Transactions on Mobile Computing*, March 2010
- R.L. Cruz, A. Santhanam, Optimal routing, link scheduling and power control in multihop wireless networks, in *IEEE INFOCOM*, April 2003
- D. Decouto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing. *Wirel. Netw.* **11**, 419–434 (2005). Springer
- Defense Research and Development Canada, Human Factors Issues for Controlling Uninhabited Aerial Vehicles, Tech. rep., 2010
- D.V. Dimarogonas, K.H. Johansson, Bounded control of network connectivity in multi-agent systems. *IET Control Theory Appl.* **4**(8), 1330–1338 (2010)
- C. Dixon, Controlled Mobility of Unmanned Aircraft Chains to Optimize Network Capacity in Realistic Communication Environments, Ph.D. thesis, University of Colorado at Boulder, Boulder, 2010
- C. Dixon, E. Frew, *Decentralized Extremum-Seeking Control of Nonholonomic Vehicles to Form a Communication Chain*, Lecture Notes in Control and Information Sciences, vol. 369 (Springer, Berlin/New York, 2007), pp. 311
- C. Dixon, E. Frew, Maintaining optimal communication chains in robotic sensor networks using mobility control. *Mobile Netw. Appl.* **14**(3), 281–291 (2009)
- J.A. Fax, R.M. Murray, Information flow and cooperative control of vehicle formations, in *Automatic Control, IEEE Transactions on*, **49**(9), 1465–1476, (2004)
- Federal Communications Commission, The Role of Deployable Aerial Communications Architecture in Emergency Communications and Recommended Next Steps, Tech. rep., Public Safety and Homeland Security Bureau, Washington, DC, 2011
- R.G. Garroppo, S. Giordano, L. Tavanti, Experimental evaluation of two open source solutions for wireless mesh routing at layer two, in *International Symposium on Wireless Pervasive Computing*, 2010
- M.C.D. Gennaro, A. Jadbabaie, Decentralized control of connectivity for multi-agent systems, in *IEEE Conference on Decision and Control*, Dec 2006
- A. Ghaffarkhah, Y. Mostofi, Communications-aware navigation functions for cooperative target tracking, in *American Control Conference (ACC)*, June 2009
- A. Ghaffarkhah, Y. Yan, Y. Mostofi, Dynamic coverage of time varying environments using a mobile robot – a communication aware perspective, in *IEEE Globecom*, Dec 2011
- N. Goddemeier, S. Rohde, J. Pojda, C. Wietfeld, Evaluation of potential fields mobility strategies for aerial network provisioning, in *IEEE Globecom*, Dec 2011
- A. Goldsmith, *Wireless Communications* (Cambridge University Press, New York, 2005)
- G.R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, B. Walke, IEEE 802.11s: the wlan mesh standard. *IEEE Wirel. Commun.* **2**, 104–111 (2010)
- G. Hollinger, S. Singh, Multi-robot coordination with periodic connectivity, *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010
- M.A. Hsieh, A. Cowley, R.V. Kumar, C.J. Taylor, Maintaining network connectivity and performance in robot teams. *IEEE J.* **25**(1–2), 111–131 (2008)
- A. Ibrahim, K. Seddik, K. Liu, Improving Connectivity via Relays Deployment in Wireless Sensor Networks (2007), pp. 1159–1163 http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4411133&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4411133
- A.S. Ibrahim, K. Seddik, K.J.R. Liu, Connectivity-aware network maintenance via relays deployment. *IEEE Trans. Wirel. Commun.* **8**(1), 356–366 (2009)
- L.B. Johnson, S. Ponda, H.-L. Choi, J.P. How, Asynchronous decentralized task allocation for dynamic environments, in *Proceedings of the AIAA Infotech@Aerospace Conference*, St. Louis, MO, Mar 2011
- T. Jonson, J. Pezeshki, V. Chao, K. Smith, J. Fazio, Applications of delay tolerant networking (DTN) in airborne networks, in *IEEE MILCOM*, 2008.
- F. Kaabi, F. Filali, Channel allocation and routing in wireless mesh networks: a survey and qualitative comparison between schemes. *Int. J. Wirel. Mobile Netw.* **2**(1), 104–111 (2010)

- V. Kawadia, P.R. Kumar, Simultaneous routing and resource allocation via dual decomposition. *IEEE J. Sel. Areas Commun.* **1**, 76–88 (2005)
- Y. Kim, M. Mesbahi, On maximizing the second smallest eigenvalue of a state-dependent graph laplacian, in *American Control Conference (ACC)*, July 2005
- A.N. Kopeikin, S.S. Ponda, H.-L. Choi, J.P. How, Real-time dynamic planning to maintain network connectivity in a team of heterogeneous unmanned systems, in *Wi-UAV 2011, 2nd International Workshop on Wireless Networking for Unmanned Autonomous Vehicles at the IEEE GlobeComm Conference*, Dec 2011
- A.N. Kopeikin, A. Clare, O. Toupet, J.P. How, M.L. Cummings, Flight testing a heterogeneous multi-UAV system with human supervision, in *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2012
- Q.J. Krueger, Electromagnetic Interference and Radio Frequency Interference Shielding of Carbon-Filled Conductive Resins, Master's thesis, Michigan Technological University, Houston, MI, Feb 2002
- K. Kulling, Optimal and Receding-Horizon Path Planning Algorithms for Communications Relay Vehicles in Complex Environments, Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, June 2009
- H.T. Kung, C.-K. Lin, T.-H. Lin, S.J. Tarsa, D. Vlah, Measuring diversity on a low-altitude UAV in ground-to-air wireless 802.11 mesh network, in *IEEE Globecom*, Dec 2010
- B.M. Leiner, R.J. Ruth, A.R. Sastry, goals and challenges of the DARPA GloMo program. *IEEE Pers. Commun.* **3**(6), 34–43 (1996)
- N.E. Leonard, D.A. Paley, R.E. Davis, D.M. Fratantoni, F. Lekien, F. Zhang, Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *J. Field Robot.* **26**(6), 718–740 (2010)
- M. Lindhe, K.H. Johansson, Adaptive exploitation of multipath fading for mobile sensors, *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010
- X. Liu, S.S. Mahal, A. Goldsmith, J.K. Hedrick, Effects of communication delay on string stability in vehicle platoons, in *IEEE International Conference on Intelligent Transportation Systems*, Aug 2001
- C. Marshall, M. Mears, S. Rasmussen, 2010 ICE-T cooperative control flight testing, in *AIAA Infotech@Aerospace Conference*, April 2011
- N. Michael, M.M. Zavlanos, V. Kumar, G.J. Pappas, *Maintaining Connectivity in Mobile Robot Networks*, Springer Tracts on Advanced Robotics, vol. 54 (2009), pp. 117–126
- N. Michael, M. Schwager, V. Kumar, D. Rus, An experimental study of time scales and stability in networked multi-robot systems, in *Springer Tracts on Advanced Robotics – International Symposium on Experimental Robotics*, 2010
- E. Modiano, 16.36: *Communication Systems and Networks*, Lecture 1 (Massachusetts Institute of Technology, Cambridge, 2012)
- Y. Mostofi, Decentralized communication-aware motion planning in mobile networks: an information-gain approach. *J. Intell. Robot Syst.* **56**(2), 718–740 (2009)
- Y. Mostofi, A.G. Ruiz, A. Ghaffarkhah, D. Li, Characterization and modeling of wireless channels for networked robotic and control systems – a comprehensive overview, in *IEEE International Conference on Intelligent Robots and Systems*, 2009
- T. Mulkerin, M. Zernic, General aviation aircraft data communications analysis using a web-based tool, in *Digital Avionics Systems Conference*, Dec 2003
- K. Negus, A. Stephens, J. Lansford, Goals and challenges of the DARPA GloMo program. *IEEE Pers. Commun.* **7**(1), 20–27 (2000)
- H. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, J. Spector, Autonomous communication relays for tactical robots, in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, 2003
- J.L. Ny, A. Ribeiro, G.J. Pappas, Adaptive Communication-Constrained Deployment of Unmanned Vehicle Systems – (Internet Draft), Tech. rep., University of Pennsylvania, Philadelphia, PA., 2011, <http://www.seas.upenn.edu/aribeiro/preprints/2012lenyetal.pdf>
- Office of the Secretary of Defense, FY2009-2034 Unmanned Systems Integrated Roadmap, 2009

- R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **49**(9), 1520–1533 (2004)
- R. Olfati-Saber, A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems. *IEEE Trans. Autom. Control* **95**(1), 215–233 (2007)
- R.C. Palat, A. Annamalai, J.H. Reed, Cooperative relaying for ad-hoc ground networks using swarm UAVs, in *IEEE Military Communications Conference*, 2005
- C. Perkins, E. Belding-Royer, *Dynamic Source Routing in Ad-Hoc Wireless Networks* (Kluwer Academic, Dordrecht, 1996)
- C. Perkins, E. Belding-Royer, Ad-hoc on-demand distance vector (AODV) routing, in *IEEE Workshop on Mobile Computing Systems*, Feb 1999
- S. Ponda, J. Redding, H.-L. Choi, J.P. How, M. Vavrina, J. Vian, Decentralized planning for complex missions with dynamic communication constraints, in *American Control Conference (ACC)*, June 2010
- S. Ponda, L.B. Johnson, H.-L. Choi, J.P. How, Ensuring network connectivity for decentralized planning dynamic environments, in *AIAA Infotech@Aerospace Conference*, April 2011
- T. Razafindralambo, D. Simplot-Ryl, Connectivity preservation and coverage schemes for wireless sensor networks. *IEEE Trans. Autom. Control* **56**(10), 2418–2428 (2011)
- J. Reich, V. Misra, D. Rubenstein, G. Zussman, Spreadable Connected Autonomic Networks (SCAN) – Technical Report CUCS-016-08, Tech. rep., Columbia University, New York, 2008
- W. Ren, R. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **50**(5), 655–661 (2005)
- W. Ren, R.W. Beard, D.B. Kingston, Multi-agent Kalman consensus with relative uncertainty, in *American Control Conference (ACC)*, vol. 3 (2005), pp. 1865–1870
- A. Ribeiro, Notes on Implementing a IEEE 802.11s Mesh Point, in *Euro NGI Workshop*, Barcelona, Spain, Jan 2008
- A. Ribeiro, Stochastic learning algorithms for optimal design of wireless fading networks, in *IEEE Workshop on Signal Processing and Advances in Wireless Communications*, June 2011
- M.W. Ritter, R.J. Friday, M. Cunningham, T.G. Logan, The architecture of Metricom’s microcellular data network and details of its implementation as the second and third generation ricochet wide-area mobile data service, in *IEEE Emerging Technology Symposium on Broadband Communications*, Sept 2001, pp. 143–152
- T. Salonidis, M. Garetto, A. Saha, E. Knightly, Identifying high throughput paths in 802.11 mesh networks: a model-based approach, in *IEEE International Conference on Network Protocols*, Oct 2007
- P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks* (Wiley, Chichester, 2005)
- S. Sariel, T. Balch, Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments, in *Proceedings of the AIAA Workshop on Integrating Planning Into Scheduling*, 2005
- T. Schouwenaars, E. Feron, J. How, Multi-vehicle path planning for non-line of sight communication, in *Proceedings American Control Conference*, Jun 2006, Minneapolis, Minnesota, pp. 5757–5762.
- M. Schwager, N. Michael, V. Kumar, D. Rus, Time scales and stability in networked multi-robot systems, in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011
- A. Srinivas, E. Modiano, Joint node placement and assignment for throughput optimization in mobile backbone networks. *IEEE INFOCOM*, Dec 2008
- D.M. Stipanović, G. Inalhan, R. Teo, C.J. Tomlin, Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica* **40**, 1285–1296 (2004)
- D. Tardioli, A. Mosteo, L. Riazuelo, J. Villarroel, L. Montano, Enforcing Network Connectivity in Robot Team Missions. *Int. J. Robot. Res.* **29**(4), 460–480 (2010)
- S. Toumpis, S. Gitis, Load balancing in wireless sensor networks using Kirchhoff’s voltage law, in *IEEE INFOCOM*, April 2009
- U.S. Air Force Chief Scientist (AF/ST), Technology Horizons: A Vision for Air Force Science & Technology During 2010–2030, Tech. rep., United States Air Force, 2010

- N. Wagle, E. Frew, Spatio-temporal characterization of airborne radio frequency environments, *IEEE Globecom*, Dec 2011
- G. Warwick, AFRL Aims for MAV Lead (2009), Available: www.aviationweek.com, Accessed July 2011
- L. Xiao, M. Johansson, S.P. Boyd, Simultaneous routing and resource allocation via dual decomposition. *IEEE Trans. Commun.* **52**(7), 1136–1144 (2004)
- F. Xue, P. Kumar, The number of neighbors needed for connectivity of wireless networks. *Wirel. Netw.* **10**, 169–181 (2004)
- Y. Yan, Y. Mostofi, Robotic router formation in realistic communication environments- a bit error rate approach, in *IEEE Military Communications Conference*, 2010
- P. Yang, R.A. Freeman, G.J. Gordon, K.M. Lynch, S.S. Srinivasa, R. Sukthankar, Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica* **46**, 390–396 (2010)
- E. Yanmaz, R. Kuschnig, C. Bettstetter, Channel measurements over 802.11a-based UAV-to-ground links, in *IEEE Globecom*, Dec 2011
- T. Zajkowski, S. Dunagan, J. Eilers, Small UAS communications mission, in *11th Biennial USDA Forest Service Remote Sensing Application Conference*, April 2006
- M.M. Zavlanos, G.J. Pappas, Potential fields for maintaining connectivity of mobile networks. *IEEE Trans. Robot.* **23**(4), 812–816 (2007)
- M.M. Zavlanos, G.J. Pappas, Distributed connectivity control of mobile networks, *IEEE Trans. Robot.* **24**(6), 1416–1428 (2008)
- M.M. Zavlanos, M.B. Egerstedt, G.J. Pappas, Graph theoretic connectivity control of mobile robot networks. *IEEE J.* **99**(9), 1525–1540 (2011)
- Y. Zhang, M. Quilling, Optimal backbone generation for robotic relay networks, in *IEEE International Conference on Computer Communications and Networks*, 2011
- H. Zhu, A.L. Swindlehurst, K. Liu, Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles. *IEEE J. Veh. Technol.* **58**(7), 3533–3546 (2009)

Information-Theoretic Exploration of Unmanned Aerial Vehicle in Unknown Cluttered Environment

67

Kwangjin Yang, Seng Keat Gan, and Salah Sukkarieh

Contents

| | | |
|--------|---|------|
| 67.1 | Introduction | 1656 |
| 67.2 | 3D Gaussian Process Occupancy Map | 1658 |
| 67.2.1 | Foundations of Gaussian Processes | 1658 |
| 67.2.2 | Probabilistic Least Square Classification | 1659 |
| 67.2.3 | 3D Laser Sensor Model | 1659 |
| 67.2.4 | Training GP and Classifier Hyperparameters | 1659 |
| 67.2.5 | Sparse Training Data Extraction | 1661 |
| 67.2.6 | Safety Boundary Construction | 1661 |
| 67.3 | Information-Theoretic Path Planning for Exploration Mission | 1662 |
| 67.3.1 | Information Measures | 1662 |
| 67.3.2 | Decision of the Appropriate Information Measure | 1664 |
| 67.3.3 | Evaluation of Uncertainty of the Environment | 1664 |
| 67.3.4 | Information-Theoretic Path Planning Algorithm | 1666 |
| 67.4 | Simulation Experiments | 1669 |
| 67.5 | Conclusions | 1674 |
| | References | 1674 |

K. Yang (✉)

Department of Aerospace and Mechanical Engineering, Korea Air Force Academy, Chungbuk, Republic of Korea

e-mail: ykj4957@gmail.com

S.K. Gan

Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW, Australia

e-mail: s.gan@acfr.usyd.edu.au

S. Sukkarieh

Australian Centre for Field Robotics (ACFR), School of Aerospace, Mechanical & Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW, Australia

e-mail: salah@acfr.usyd.edu.au

Abstract

An information-theoretic path planning for exploration mission in unknown environment is proposed. Instead of using traditional grid-based occupancy maps, the Gaussian process (GP) is used to build an environmental map. The GP map performs inference directly on the collected sensor datasets which allows it to infer the probability of collision for any query point in continuous 3D space, removing the need for maintaining a full discrete map. This GP map is fused with the Rapidly exploring random tree (RRT) path planner to plan a safe path and acquire information about the unknown environments. Using mutual information as an information measure, the most informative path is chosen as the path for exploration. Simulation results show that GP map combined with RRT planner achieves exploration task successfully in unknown complex environment and present their potential implementation in unmanned aerial vehicle (UAV) missions.

67.1 Introduction

Autonomous navigation in unknown cluttered environments is a challenging problem. To achieve this task the vehicle must have the ability to build an environment representation, generate a safe and feasible path, and execute the plan reliably. A path planner plans a path based on its understanding about the environment. In a global collision-free path planning problem, the environment is represented by an occupancy map that contains the joint sensory information about the robot operation space (Scherer et al. 2008). The most straightforward approach is to discretize the operational area into a predefined uniform grid where each grid cell holds the probability of occupancy within the cell (LaValle 2006). This grid based path planning approach has been widely used and successfully demonstrated for real-time applications, especially well suited for robots that operate in two-dimensional planar domains (Ferguson and Stentz 2006; Grabowski et al. 2006; Montemerlo et al. 2008). However, in three-dimensional applications, the grid-based approach suffers greatly from the curse of dimensionality because the grid density required to reasonably represent the 3D operational environment to a certain resolution exponentially increases. Furthermore, the flexibility of the path planner is restricted by the resolution and orientation of the grid.

In this research, the traditional grid-based occupancy map approach is replaced with a Gaussian process (GP) occupancy map for path planning. The GP map operates directly on the collected environment data, which eliminates the grid-based map and the concerns with grid size, grid shape, obstacle shape, and map resolution. There is no need to associate the data to discrete cells, and thus no information is lost. The GP map outputs the probability of occupancy given any selected query point in the continuous space. This grants it the potential to explore all possible paths in the full space when used in conjunction with a continuous path planner. Furthermore, a path planned with a GP map is embedded with the probability of collision along the path. This valuable information is not available in other grid- or graph-based frameworks.

In this research, a Rapidly exploring random tree (RRT) path planner is used for the planning algorithm. The GP map is well matched with the RRT framework since it works as collision detection module that implicitly represents free space. The RRT expands its network as part of the exploration mission to get information of the unknown cluttered environment. Autonomous vehicles with sensing capabilities can actively take advantage of their mobility to get information about the environment. A key component of active environment sensing systems is the dependence of information gathering on robot motion, and the ability to predict in advance the effect of robot motion on the quality of information that is collected (Frew 2007a). Information-theoretic approaches, which use information measures, such as entropy, Fisher information, and mutual information as an objective functionality, have been actively investigated (Bourgault et al. 2002; Stachniss et al. 2005; Ryan et al. 2007; Frew 2007b). An appropriate selection of the information criteria is important because each information measure has a different way of characterizing the uncertainty. Mutual information is used as an information measure because it provides a pre-observation measure of the usefulness of obtaining information about the value of the sensor measurement (Grocholsky 2002).

The role of the planner in the exploration mission is to generate a path that maximizes the information of the environment. Therefore, it is important to evaluate properly the uncertainty of the environment based on the accumulated environment data, because the quality of path is quantified by this evaluation. GP is an excellent method to evaluate the uncertainty because it has an inference capability in occluded areas. The exploration problem, which aims to minimize the uncertainty in the environment, is similar to the sensor management problem. The main difference between mobile robotic networks and sensor management is that there is no restriction to the deployment of sensors in sensor networks (Singh et al. 2010; Krause et al. 2008), but there are severe constraints in a mobile robot such as a UAV: kinematic and dynamic constraints which restrict the evolution of the vehicle states. In addition to the differential constraint of the vehicle, there are other constraints which makes the exploration task challenging. The observation is also constrained by the limited sensor field of view, and the difficulties are compounded by the occlusion due to obstacles. Finally, the vehicle must satisfy the safety condition which ensures no collision with obstacles. All of these constraints complicate the planning problem. In this research, an information-theoretic path planning algorithm is proposed for the exploration mission which takes into account these constraints within the RRT framework. Simulation results show that GP map combined with RRT planner achieves exploration task successfully in unknown complex environment and present their potential implementation in unmanned aerial vehicle (UAV) missions.

The rest of the chapter is organized as follows: Sect. 67.2 presents a brief background of the Gaussian process and how it can be used as a tool to generate a continuous occupancy map. Section 67.3 addresses information-theoretic exploration in cluttered environments. Section 67.4 presents simulation results of a UAV performing a navigation and exploration task, and the conclusions are presented thereafter.

67.2 3D Gaussian Process Occupancy Map

67.2.1 Foundations of Gaussian Processes

Gaussian processes is a powerful nonparametric tool for regression based on locally available datasets. It represents a family of distributions over functions and inference takes place directly in the function space (Rasmussen and Williams 2006). It is particularly useful to infer the most likely function value f_* and its corresponding variance for a given test input \mathbf{x}_* when exact observation on \mathbf{x}_* is not available. Especially for the case when there is no information about the underlying function, inference with GP can be achieved by placing a multivariate Gaussian distribution across the function space.

Consider $\chi = \{(\mathbf{x}_i, y_i) | i = 1, \dots, n\}$ as a collection of N datasets, where $\mathbf{x}_i \in \mathbb{R}^D$ represents a D dimensional training input and $y_i \in \mathbb{R}$ the corresponding scalar training output (in this research, y_i is +1 or -1 for occupied and non-occupied, respectively). Condition on χ , for any selected testing input \mathbf{x}_* , the predictive distribution of the function output becomes

$$f_* | \mathbf{x}_*, \chi \sim N(\mu_*, \sigma_*^2) \quad (67.1)$$

with mean

$$\mu_* = k(\mathbf{x}_*, \mathbf{x}) [k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} \mathbf{y} \quad (67.2)$$

and variance

$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{x}) [k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} k(\mathbf{x}, \mathbf{x}_*) \quad (67.3)$$

Here, $k(\mathbf{x}, \mathbf{x}')$ is the GP covariance or kernel function, which effectively sets the correlation between the output of \mathbf{x} and \mathbf{x}' based on trained kernel hyperparameters.

Selection of the appropriate covariance function highly depends on the application. A commonly used kernel is the stationary squared exponential(SE) kernel:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{M}(\mathbf{x} - \mathbf{x}')\right) \quad (67.4)$$

where σ_f^2 is the signal strength, σ_n^2 the Gaussian noise variance, and $\mathbf{M} = \text{diag}[1/l_1^2, \dots, 1/l_D^2]$, where l_i is the length scale that sets the kernel smoothness characteristic in different dimensions. There exist other covariance functions that outperform squared exponential in terms of modeling noncontinuous spatial functions, such as the neural-network demonstrated in OCallaghan et al. (2009). Nevertheless, the squared exponential is by far the most computationally efficient kernel that suits in this application.

67.2.2 Probabilistic Least Square Classification

The aim here is to estimate the probability of collision for any chosen location in 3D workspace for collision-free path planning. There are various classification algorithms available, but, as described by Rasmussen and Williams (2006) and implemented by OCallaghan et al. (2009), one simple yet efficient method is to treat the classification as a GP regression problem and then further squash the GP predictions through a cumulative Gaussian sigmoid function to obtain a valid probability

$$p(y_i | \mathbf{x}, \mathbf{y}_{-i}, \theta) = \Phi \left(\frac{y_i (\alpha \mu_i + \beta)}{\sqrt{1 + \alpha^2 \sigma_i^2}} \right) \quad (67.5)$$

where Φ is a cumulative unit Gaussian which is defined as $\Phi = (2\pi)^{-1/2} \int_{-\infty}^z \exp(-t^2/2) dt$, θ denotes the hyperparameters of the covariance function, α and β are the classifier hyperparameters that can be trained using the leave-one-out cross-validation procedure as described in Sect. 67.2.4. The full method is called probabilistic least square classification (PLSC).

67.2.3 3D Laser Sensor Model

A rotary UAV (RUAV) is considered in this research. It is assumed that the RUAV platform is equipped with a sparse 3D laser range finder, capable of scanning n_b beams every instant the sensor is activated, as depicted in Fig. 67.1a. Each time the sensor scans the environment, the following information is stored for each beam segment $\Lambda^k = \{(\mathbf{s}_i, \mathbf{u}_i, b_i) | i = 1 : n_b\}^k$ where \mathbf{s}_i , \mathbf{u}_i and b_i are, respectively, the sensor position, beam vector from sensor to return position, and a boolean indicating either 1 if the beam hits an obstacle and 0 otherwise.

67.2.4 Training GP and Classifier Hyperparameters

Training datasets can be obtained by firstly flying the RUAV platform around an obstacle in a simulated environment, like one in Fig. 67.1a, collecting laser beam data. The collected laser beam segments are then divided into discrete points where each point is described by its 3D location \mathbf{x} and corresponding label y , used for the hyperparameter training described below.

Training the GP hyperparameters is effectively setting the appropriate properties for the covariance function. A common method is based on maximization of the log of the marginal likelihood:

$$\theta_{\text{opt}} = \arg \max [\ln p(\mathbf{y} | \mathbf{x}, \theta)] \quad (67.6)$$

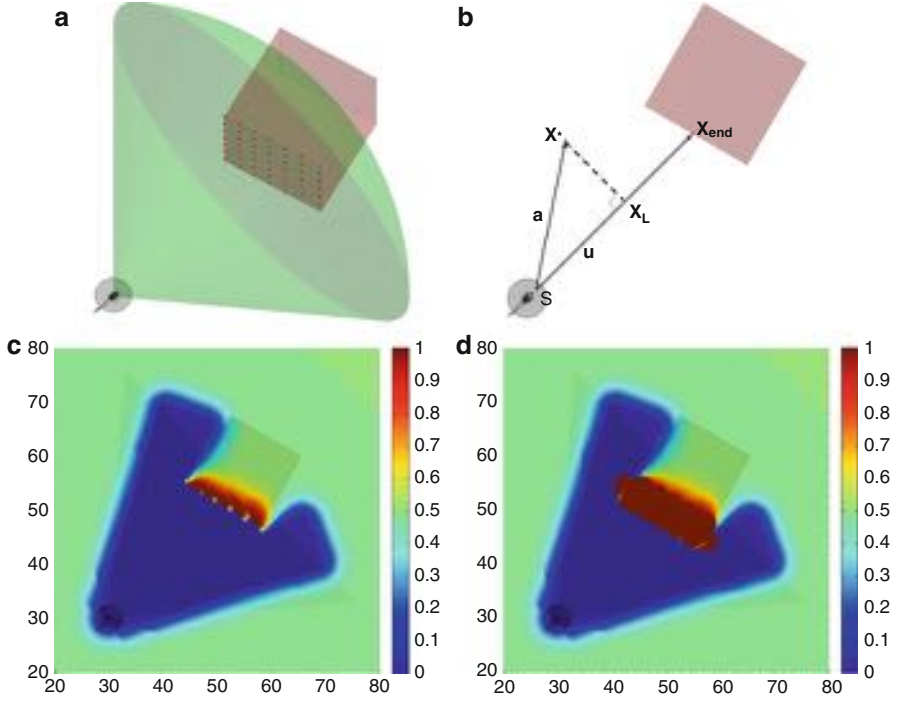


Fig. 67.1 Gaussian process occupancy map construction: (a) shows the instant where an obstacle is detected within the FOV of the 3D laser range finder. The notation used in Algorithm 1 for extracting training data from a single laser beam is illustrated in (b). Diagram (c) illustrates the high resolution horizontal plane GP map at the height of the sensor with the *color* indicating the probability of obstacle. The modifications made to suit path planning are shown in (d) where a safety boundary parameter d_{safety} of 5 m is used. Note that (c) and (d) are for visualization purposes. There is no need to construct the full map for online path planning applications

where

$$\ln p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2}\mathbf{y}^T K_y \mathbf{y} - \frac{1}{2} \ln |K_y| - \frac{n}{2} \ln 2\pi \quad (67.7)$$

and $K_y = k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I$. The optimal solution of this method produces a model that balances between model complexity and data fitness, which is known as the Occam's razor principle.

Having trained the GP hyperparameters, the classifier hyperparameters can then be trained using the cross validation approach, based on maximizing the sum of the log leave-one-out (LOO) probability:

$$\{\alpha, \beta\}_{\text{opt}} = \arg \max \left[\sum_{i=1}^N \ln p(y_i | \mathbf{x}, \mathbf{y}_{-i}, \theta, \alpha, \beta) \right] \quad (67.8)$$

where

$$\ln p(y_i | \mathbf{x}, \mathbf{y}_{-i}, \theta) = -\frac{1}{2} \ln \sigma_i^2 - \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \frac{1}{2} \ln 2\pi \quad (67.9)$$

and

$$\mu_i | \theta = \mathbf{y}_i - \left[k(\mathbf{x}, \mathbf{x})^{-1} \mathbf{y} \right]_i / \left[k(\mathbf{x}, \mathbf{x})^{-1} \right]_{ii} \quad (67.10)$$

$$\sigma_i^2 | \theta = 1 / \left[K(\mathbf{x}, \mathbf{x})^{-1} \right]_{ii} \quad (67.11)$$

are the mean and variance after leaving out the training dataset y_i . Note that these hyperparameters are fixed once they are trained in this application.

67.2.5 Sparse Training Data Extraction

Λ is a database of collected continuous beam segments representing the true occupancy information about the environment. When the occupancy hypothesis of a point \mathbf{x}_* is queried, a series of sparse spatial training data is to be extracted from its local neighborhood database subset $\Lambda_{\mathbf{x}_*} = \{\Lambda_{j, \mathbf{x}_*} | j = 1 : n_s\}$ where n_s is the number of the nearest beam segment. Among $\Lambda_{\mathbf{x}_*}$, training input \mathbf{x} and its corresponding label \mathbf{y} can thus be extracted using Algorithm 1. Lines 1 and 2 determine the distance d and location of the point that is projected orthogonally from \mathbf{x}_* onto the beam segment \mathbf{u} (note that $\hat{\mathbf{u}}$ is the unit vector of \mathbf{u}). Line 3 extracts the end of the beam segment. Lines 4–10 simply infer the conditions when the point \mathbf{x}_\perp lies outside of the beam segment. Lines 11–13 use a simple logic to set a safety boundary around obstacles for the path planning purpose that will be further described in the following section.

67.2.6 Safety Boundary Construction

Setting a safety region around obstacles is crucial for an airborne platform as flying close to obstacles significantly increases the risk of collision. A simple yet efficient way is to include this feature in the GP Map framework, appearing in lines 11–13 in Algorithm 1.

The aim is to enforce maximum correlation between the output of \mathbf{x}_{end} and \mathbf{x}_* , for \mathbf{x}_{end} that is extracted from an obstacle beam, when the distance between them is less than the safety distance d_{safety} . The simplest way to achieve this without modifying the GP framework is to simply set \mathbf{x} to be the same as \mathbf{x}_* , which, referring to (Eq. 67.4), effectively sets $k(\mathbf{x}, \mathbf{x}')$ to the maximum value of the covariance function. This method directly embeds the safety boundary feature in the training dataset extraction process, flexibly controlled by the single parameter d_{safety} with no induced cost while transitioning from 2D to 3D workspace.

Algorithm 1 Sparse training data extraction

Require: Λ_{x_*}, x_*

```

1:  $d \leftarrow a \cdot \hat{u}$ 
2:  $x_{\perp} \leftarrow s + d \hat{u}$ 
3:  $x_{end} \leftarrow s + u$ 
4: if  $d \leq 0$  then
5:    $x \leftarrow s, y \leftarrow -1$ 
6: else if  $d \geq |u|$  then
7:    $x \leftarrow x_{end}, y \leftarrow 2b - 1$ 
8: else
9:    $x \leftarrow x_{\perp}, y \leftarrow -1$ 
10: end if
11: if  $b = 1$  &  $|x_{end} - x_*| \leq d_{safety}$  then
12:    $x \leftarrow x_*, y \leftarrow 1$ 
13: end if

```

67.3 Information-Theoretic Path Planning for Exploration Mission

If a perfect information about an environment is known a priori, it is possible to plan a path before the mission. In real applications, however, this is rarely the case. The vehicle must sense the environment using onboard sensors, plan the path using the latest information on the environment and vehicle, and execute this plan.

A key component of active environment sensing systems is the dependence of information gathering on robot motion, and the ability to predict in advance the effect of robot motion on the quality of information that is collected (Frew 2007a). Information-theoretic approaches, which use information measures, such as entropy, Fisher information, and mutual information as an objective functionality, are actively investigated (Bourgault et al. 2002; Stachniss et al. 2005; Ryan et al. 2007; Frew 2007b). Here, appropriate selection of the information criteria is important because each information measure has a different way of characterizing the uncertainty.

67.3.1 Information Measures

The purpose of exploration is to reduce the uncertainty of the environment. Information measures can be used to quantify the level of uncertainty of the environment. Several different information measures are described in the following subsections from which the best measure for this application will be chosen.

67.3.1.1 Shannon Information

The Shannon information, a.k.a entropy, associated with a probability distribution of $P(\mathbf{x})$ is defined as the expected value of a negative log-likelihood:

$$H(\mathbf{x}) \triangleq -E[\log P(\mathbf{x})] = - \int_{-\infty}^{\infty} P(\mathbf{x}) \log P(\mathbf{x}) d\mathbf{x} \quad (67.12)$$

where $H(\mathbf{x})$ denotes entropy associated with the variable \mathbf{x} .

If $P(\mathbf{x})$ is a probability distribution function for an n -dimensional Gaussian, the entropy for this distribution is obtained as follows:

$$H(\mathbf{x}) = -\frac{1}{2} \log[(2\pi e)^n |\mathbf{K}|] \quad (67.13)$$

where \mathbf{K} is a covariance matrix of the probability distribution and $|\cdot|$ is a determinant.

Entropy provides a compactness to the probability distribution because the determinant of a matrix is a volume measure.

67.3.1.2 Fisher Information

The Fisher information associated with a probability distribution of $P(\mathbf{x})$ is defined as the second derivative of the log-likelihood:

$$J(\mathbf{x}) \triangleq \frac{\partial^2}{\partial \mathbf{x}^2} \log P(\mathbf{x}) \quad (67.14)$$

If $P(\mathbf{x})$ is a probability distribution function for an n -dimensional Gaussian, the Fisher information for this distribution is obtained as follows:

$$J(\mathbf{x}) = \mathbf{K}^{-1} \quad (67.15)$$

From (Eq. 67.15), it can be seen that the Fisher information of a multivariate Gaussian distribution is simply the inverse of the covariance.

Fisher information is inversely related to the surface area of the probability distribution.

67.3.1.3 Mutual Information

The mutual information is defined as a negative expectation of the conditional log-likelihood:

$$I(\mathbf{x}, \mathbf{z}) \triangleq -E \log[P(\mathbf{x}|\mathbf{z})] = -E \left[\log \left\{ \frac{P(\mathbf{z}|\mathbf{x})}{P(\mathbf{z})} \right\} \right] = H(\mathbf{z}) - H(\mathbf{z}|\mathbf{x}) \quad (67.16)$$

Mutual information measures the amount of information about one variable contained in another. If \mathbf{x} is a state and \mathbf{z} is an observation, then mutual information represents an a priori measure of the information to be gained through observation. It is a function of the ratio of the density $P(\mathbf{x}|\mathbf{z})$ following an observation to the

prior density $P(\mathbf{x})$. The expectation is taken over \mathbf{z} and \mathbf{x} , so the mutual information gives an average measure of the gain to be expected before making the observation (Grocholsky 2002).

67.3.2 Decision of the Appropriate Information Measure

The objective of path planning for an exploration mission is to compute a path towards the most informative area. In order to judge the quality of the path, it is necessary to establish the notion of *informativeness*. Several information measures were introduced in the previous section. Each class of information measure results in a different output. The existence of different information measures naturally raises the question of deciding upon the most appropriate measure.

The Fisher information is a useful information measure for Gaussian representations because it is simply the information matrix (Eq. 67.15). However, it is not scalar, and so cannot be used explicitly for preference ordering of posteriors as a measure of information gain (Cole 2009).

Shannon information is a good measure for making a decision on which observations are the most uncertain. Though Shannon information is a proper measure to evaluate the absolute amount of information, it cannot capture the objective of the exploration task. What is important in the exploration mission is the amount of uncertainty reduction after the observation is made instead of the absolute magnitude of the information itself.

Mutual information provides an average measure of how much more information can be obtained if the observations are made. Most importantly, mutual information provides a preexperimental measure of the usefulness of obtaining information (through observation) about the value of \mathbf{z} (Grocholsky 2002). In that sense, mutual information is a more useful measure than entropy. The mutual information is chosen as a measure of evaluation of the information.

67.3.3 Evaluation of Uncertainty of the Environment

Consider Λ is a database of collected continuous beam segments. When a point \mathbf{x}_* is queried for the evaluation of information at this point, a series of sparse spatial training data are to be extracted from its local neighborhood database subset $\Lambda_{\mathbf{x}_*} = \{\Lambda_{j,\mathbf{x}_*} | j = 1 : n\}$, where n is the number of the nearest beam segment. Using the *Sparse Training Data Extraction* algorithm in Sect. 67.2, n number of orthogonal points are obtained from $\Lambda_{\mathbf{x}_*}$. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be the extracted orthogonal points. Then the covariance at this testing point \mathbf{x}_* is calculated using (Eq. 67.17) which comes from (Eq. 67.3):

$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{x}) k(\mathbf{x}, \mathbf{x})^{-1} k(\mathbf{x}, \mathbf{x}_*) \quad (67.17)$$

Fig. 67.2 Laser beam data and covariance map: *Blue rays* in **(a)** show laser beam data and *red circles* represent laser beam hit locations with obstacles. **(b)** shows the covariance map of the environment based on the laser beam data collected by the vehicle. The uncertainty of the occluded area is smaller than the unexplored area though there is no laser beam data behind obstacles

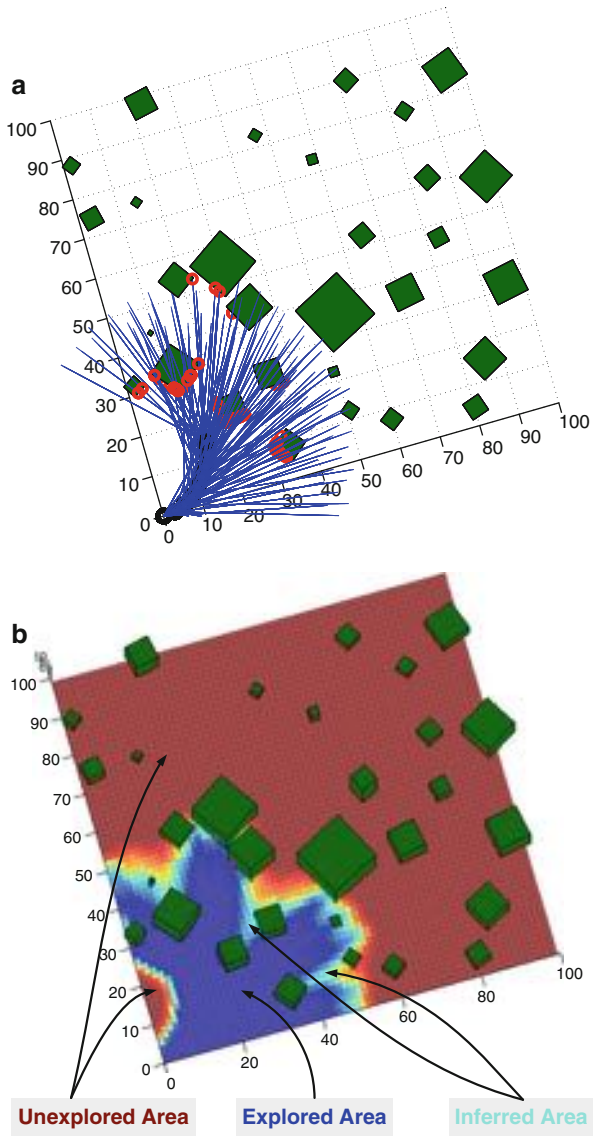


Figure 67.2 shows the laser beam data and the covariance map of the environment. Blue rays in Fig. 67.2a show laser beam data ($\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$), and red circles represent laser beam hit locations on obstacles, and squares are obstacles. The environment is discretized to evaluate the covariance map of the environment. The purpose of this discretization is only intended for visualization of the covariance. In practice, this discretization is not required. The result of the covariance evaluation of the environment is shown in Fig. 67.2b.

The advantage of GP is that it has an ability to infer into an area occluded by obstacles. Figure 67.2b shows this property. Purple denotes a highly uncertain area and blue denotes a certain area; the color between purple and blue denotes the level of inference of uncertainty. If there is laser beam data surrounding a testing point, the GP function generates a small covariance value which is represented as a blue color. However, if there is no laser beam data around a testing point, it outputs a high covariance value which is represented as a purple color. Though there is no laser beam data near the testing point, the GP can infer the uncertainty based on neighboring laser beam data. The ability to extract information from an unexplored area is a strength of the GP.

67.3.4 Information-Theoretic Path Planning Algorithm

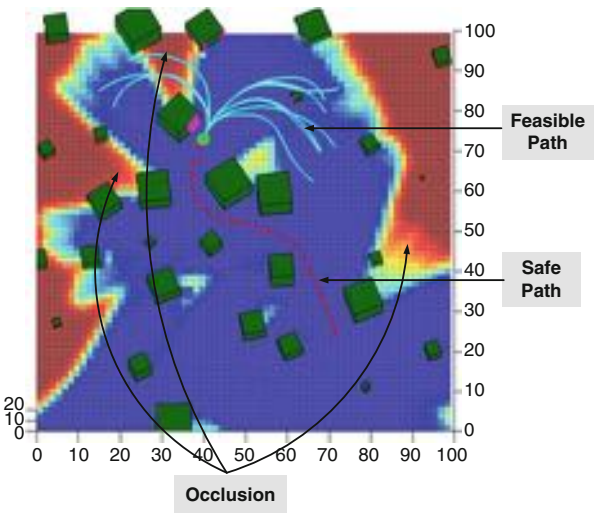
The objective of exploration is to get as much information as possible on the environment in a cost effective way.

Algorithm 2 shows the proposed information-theoretic path planning algorithm. There are four functions in this algorithm. First, the **Candidate Path Generation** function generates several candidate paths as shown in Fig. 67.3. Second, the **Sensor Emulation** function simulates laser beams along the path. Thirdly, the **Information Evaluation** function calculates the mutual information of each path. Finally, the **Informative Path Selection** function compares the information gain of each path and chooses the most informative path.

67.3.4.1 Candidate Path Generation

The role of the **Candidate Path Generation** function is to generate several candidate paths which satisfy the feasibility and safety conditions using GP map.

Fig. 67.3 The difficulties of an exploration mission: The planner must satisfy the differential constraints of the vehicle, and generate a collision-free path, overcoming a limited sensor field of view and occlusion due to obstacles



Algorithm 2 Information-theoretic path planning algorithm

- 1: Candidate Path Generation
- 2: Sensor Emulation
- 3: Information Evaluation
- 4: Informative Path Selection

Figure 67.4 shows the generated paths at the current vehicle position in Fig. 67.2a. All paths satisfy the nonholonomic constraints of the vehicle. There are no collisions between obstacles near the starting point, but there are collisions where there is little information about the environment. When a path is planned, the uncertain area is regarded as free space. If the uncertain area is considered an obstacle, the planner cannot generate a path outside of the certain area. This assumption does not affect the safety of the vehicle because as the vehicle receives information about the environment, the obstacles are identified. If any potential collisions are detected along the path, the planner triggers replanning of the path to generate a new collision-free path.

67.3.4.2 Sensor Emulation

To evaluate the information of the path, it is necessary to emulate the laser beam along the path. Firstly, the sensor emulation points are decided, and then laser beam data are radiated at these points.

Figure 67.5 shows the laser beam data. Blue rays in Fig. 67.5a represent accumulated real laser beam data (Λ^r), red rays in Fig. 67.5b represent simulated

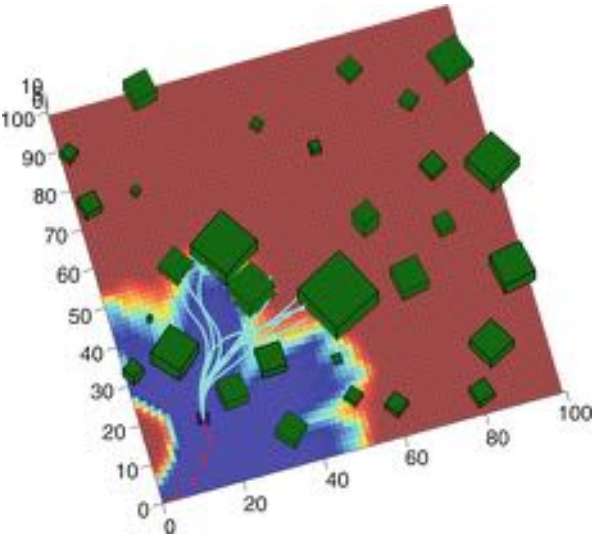


Fig. 67.4 Candidate path generation satisfying the feasibility requirement and safety condition: The *square* is a starting point of the planning, the *red dashed line* is the flight path of the vehicle, and the *cyan square* are generated candidate paths

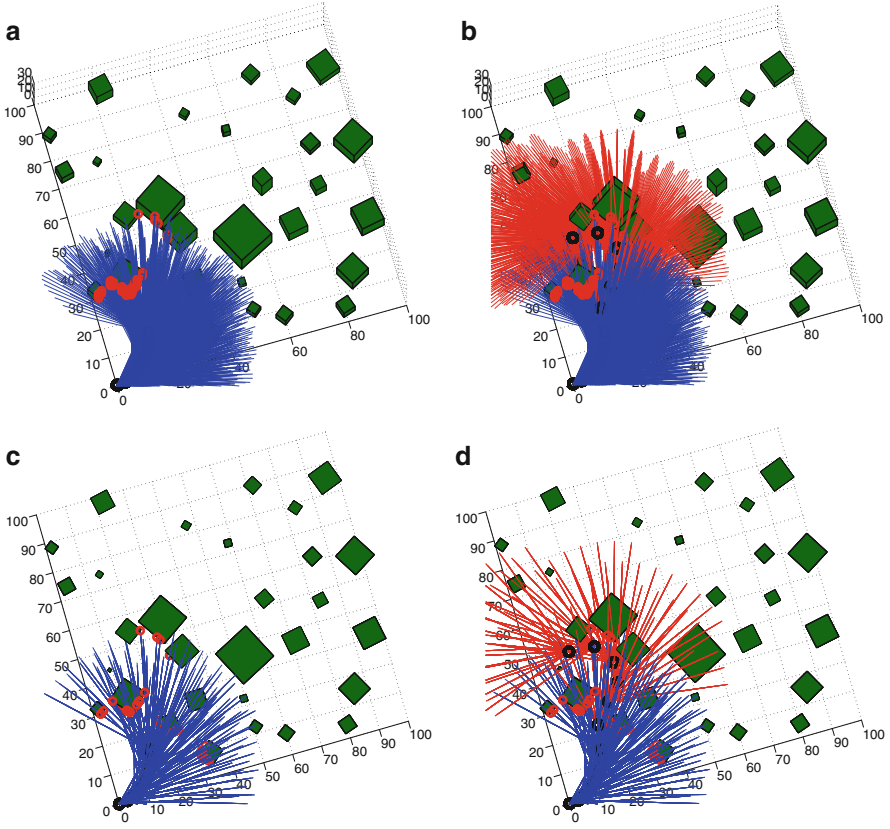


Fig. 67.5 Laser beam data: (a) shows 3D accumulated real laser beam data. (b) Red rays represent simulated laser beam data. (c) and (d) are the 2D representations of the laser beams of (a) and (b)

laser beam data (Λ^s), and Λ^c represents combined blue and red laser beams in Fig. 67.5b. Figure 67.5c, d are 2D representations of Fig. 67.5a, b.

67.3.4.3 Information Evaluation

The quality of the path can be determined by measuring the amount of reduction in the uncertainty of the environment. The goal of information evaluation is to select a path with the least uncertainty after observations are made. Mutual information is the fittest measure to quantify the variance of information about the unsensed area. Mutual information can be obtained by comparing the covariance of the environment using the real accumulated laser data with the combined laser data.

Consider Λ^r as the accumulated real laser beam data, Λ^s as the simulated laser beam data at points obtained from the *Sensor Emulation* algorithm, and $\Lambda^c = \Lambda^r \cup \Lambda^s$ as the combined data of both laser beams. There are N testing points $\mathcal{X}^t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_N^t\}$ in the environment for the evaluation of the information.

Using the *Sparse Training Data Extraction* algorithm in Sect. 67.2, n number of orthogonal points are obtained. Let $\mathcal{X}^r = \{\mathbf{x}_1^r, \mathbf{x}_2^r, \dots, \mathbf{x}_n^r\}$ be the extracted orthogonal points from accumulated real laser beam data (Λ_r), and $\mathcal{X}_i^c = \{\mathbf{x}_1^c, \mathbf{x}_2^c, \dots, \mathbf{x}_n^c\}$ be the extracted orthogonal points from i th combined laser beam data (Λ_i^c).

Condition on \mathcal{X}^r and \mathcal{X}_i^c , the covariance at testing input \mathcal{X}^t becomes

$$\mathbf{K}^r = k(\mathbf{x}^t, \mathbf{x}^t) - k(\mathbf{x}^t, \mathbf{x}^r) k(\mathbf{x}^r, \mathbf{x}^r)^{-1} k(\mathbf{x}^r, \mathbf{x}^t) \quad (67.18)$$

$$\mathbf{K}_i^c = k(\mathbf{x}^t, \mathbf{x}^t) - k(\mathbf{x}^t, \mathbf{x}^c) k(\mathbf{x}^c, \mathbf{x}^c)^{-1} k(\mathbf{x}^c, \mathbf{x}^t) \quad (67.19)$$

where \mathbf{K}^r is a covariance obtained from accumulated real laser beam data and \mathbf{K}_i^c is a predicted covariance of i th path.

Then, the mutual information in i th candidate path can be obtained:

$$I_i = \log |\mathbf{K}^r| - \log |\mathbf{K}_i^c| \quad (67.20)$$

where $\log |\mathbf{K}^r|$ is an entropy of the environment before sensor observation, and $\log |\mathbf{K}_i^c|$ is an entropy of the environment after sensor observation of i th path is made.

67.3.4.4 Informative Path Selection

Mutual information explicitly presents the amount of the uncertainty reduction. Therefore, a path which outputs maximum mutual information can offer more information about the environment. After evaluating the mutual information of all candidate paths, a path which gets the largest mutual information is chosen as the most informative path.

67.4 Simulation Experiments

Simulations were conducted of an exploration mission of an unknown environment. The objective of exploration is to maximize information about the environment. The environment is filled with obstacles of which the locations and size are randomly generated. The environment is discretized (1.5×1.5 m) for the covariance evaluation of the environment. In practice, this process is not required during the planning, but it is discretized here only for visualization purposes.

Figures 67.6 and 67.7 show the first exploration results. The candidate path number is set to 10. Figure 67.6a shows the environment to explore. Initially, the planner has no information about the environment. Figure 67.6b shows ten candidate paths and a covariance map of the environment. Purple denotes a highly uncertain areas and the blue denotes certain areas; the color between the purple and blue denotes the level of inference of uncertainty. The magenta asterisks represent laser beam hits to obstacles. The red curve denotes the selected best informative path, the black curve is the previous best path, the cyan curves represents the other

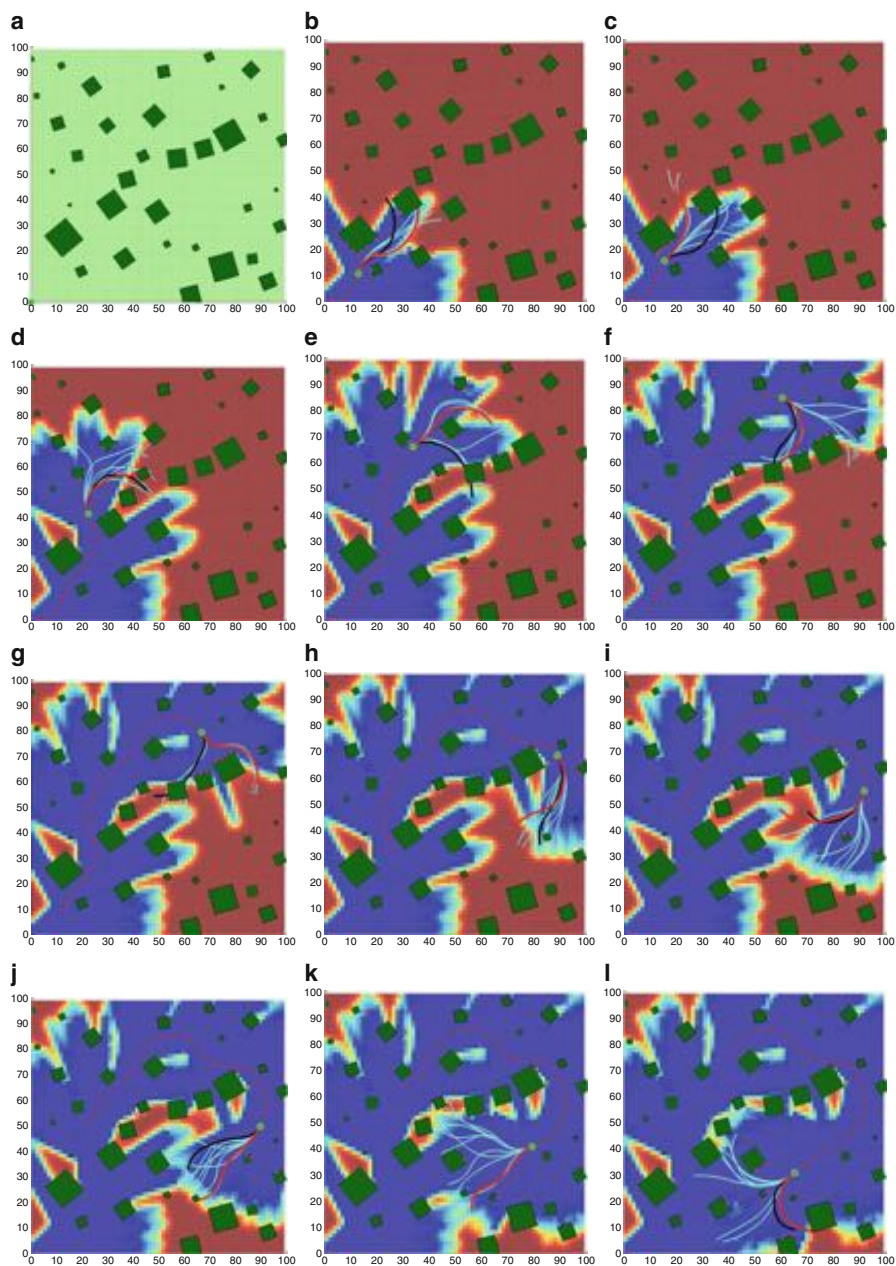


Fig. 67.6 Information-theoretic exploration case I

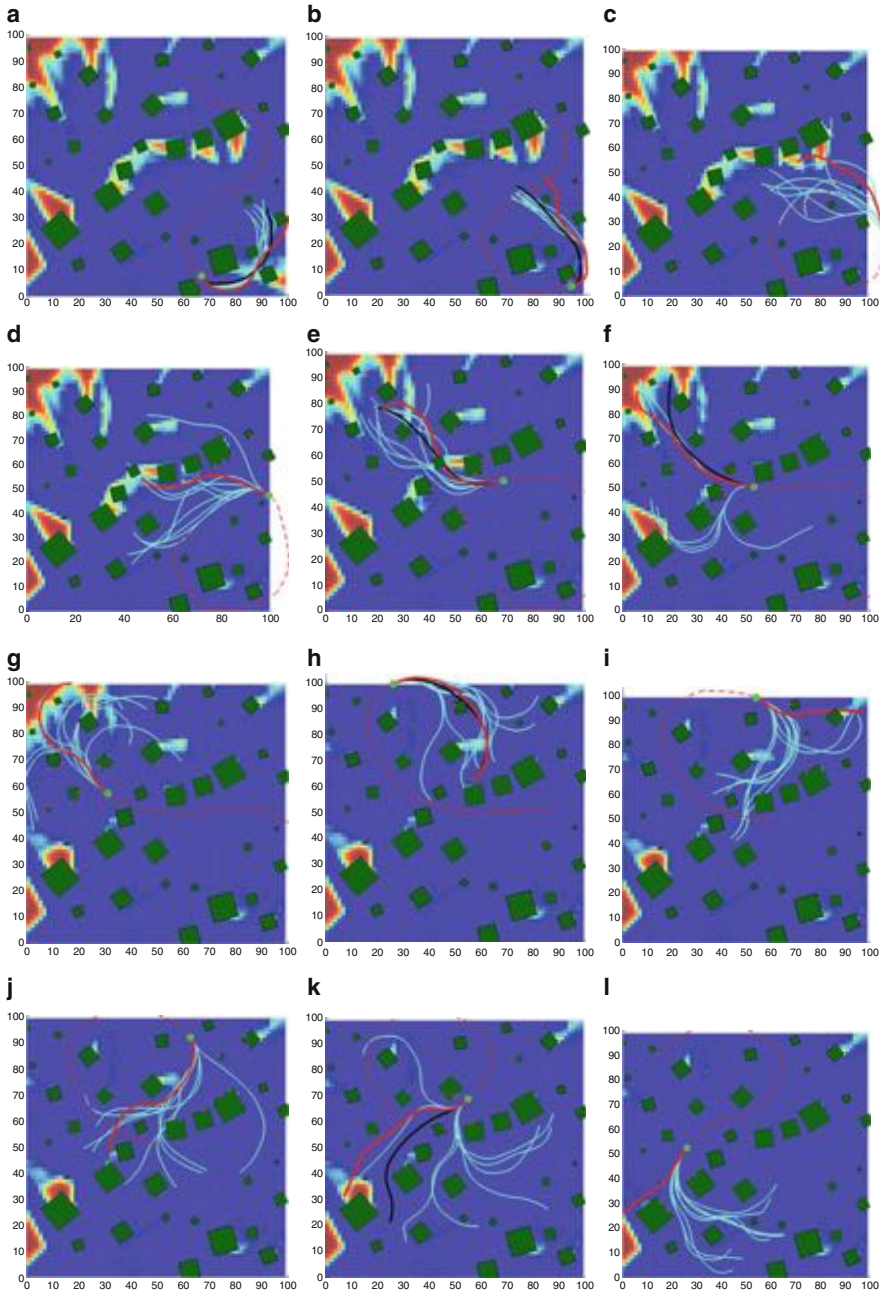


Fig. 67.7 Information-theoretic exploration case I (continue)

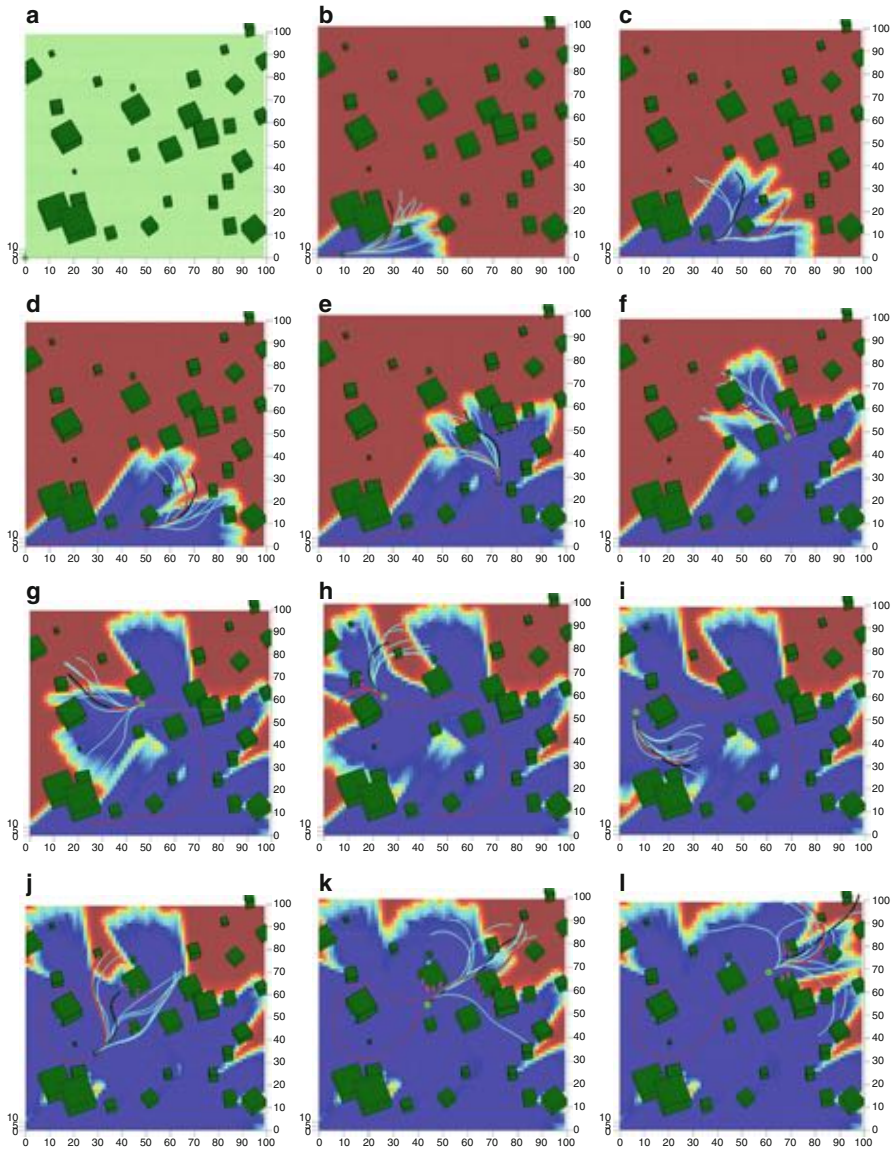


Fig. 67.8 Information-theoretic exploration case II

candidate paths, and the dotted red line is the flight path of the UAV. There are severe occlusions due to obstacles as can be seen in Fig. 67.6b–l. However, even though there are no laser beam data within the occluded area, GP can infer the information about these areas. With this inference ability, the planner can choose the most informative path. The planner selects the most right-side candidate path

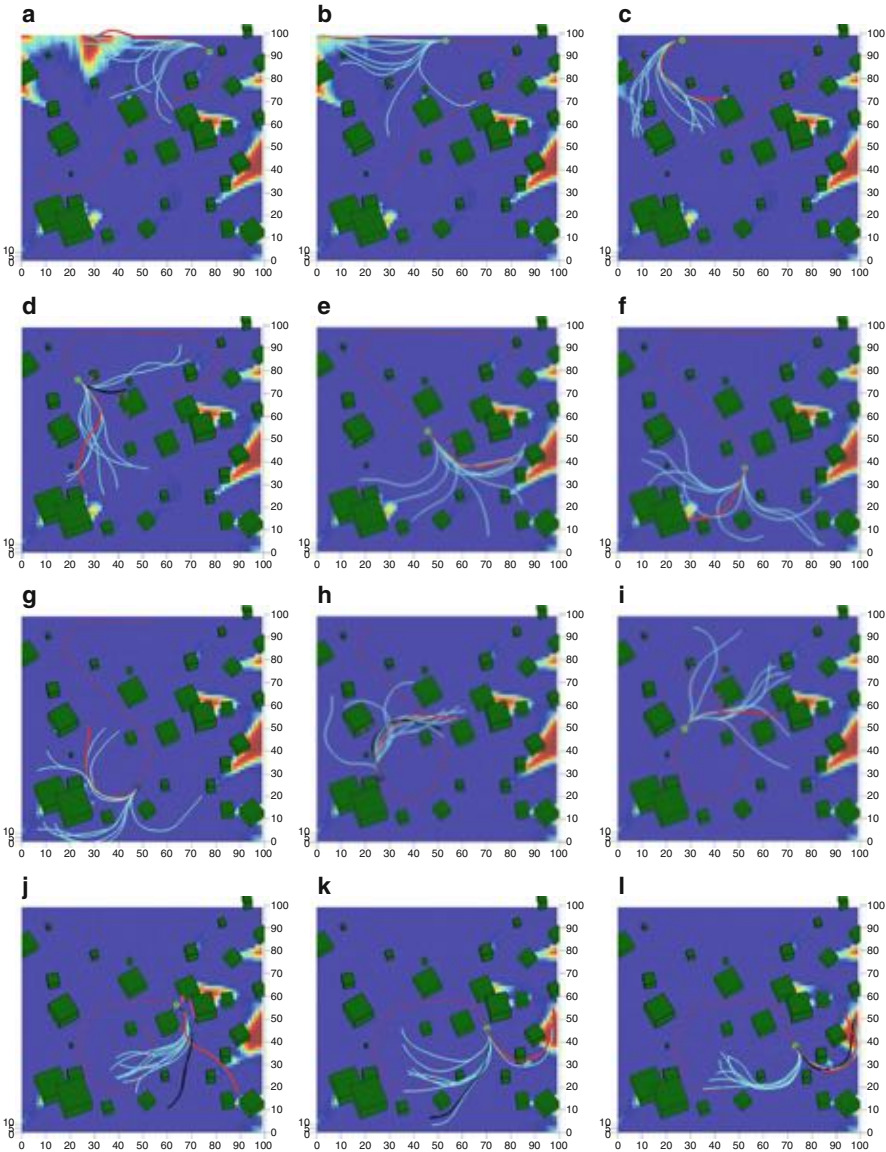


Fig. 67.9 Information-theoretic exploration case II (continue)

as a best informative path because this path can offer more information about the environment as shown in Fig. 67.6b. On the contrary, the planner selects the left-side as a best informative path because the UAV already acquire more information about the right direction area than that of the left as shown in Fig. 67.6c. In this way, the UAV selects the path which expands toward unexplored area as can be seen

in Fig. 67.6. As can be expected, the mutual information measure selects the most informative path which maximally reduces the uncertainty of the environment.

Figure 67.7 shows the progress of the exploration mission of Fig. 67.6. Gradually the uncertainty of the environment is reduced. Figure 67.71 shows that the planner generates a path toward the left-bottom corner of the environment because this area is the most uncertain region in the environment.

Figures 67.8 and 67.9 show the second exploration results. The first exploration results are plotted from the top view to clearly visualize the paths and uncertainty in the environment. On the contrary, the second exploration results are plotted in 3D view. Similar to the first example, there are many occlusions due to the densely populated obstacles as shown in Fig. 67.8. However, the RRT planner incorporating the mutual information measure generates an informative collision-free feasible path which maximizes the information of the environment.

Figure 67.9 shows the progress of the exploration mission of Fig. 67.8. There are several uncertain areas in the environment, but the RRT planner gradually reduces the uncertainty. At the end of simulation, a small portion of the environment on the right edge is not observed. Figure 67.9 shows that the planner computes a path to explore the last unvisited areas.

67.5 Conclusions

A new framework which adopts a Rapidly exploring random tree path planner with a Gaussian Process occupancy map is developed for the exploration of an unknown but cluttered environment. The GP map outputs the probability of occupancy given any selected query point in the continuous space and thus makes it possible to explore the full space when used in conjunction with a continuous path planner. Furthermore, the GP map-generated path is embedded with the probability of collision along the path which lends itself to obstacle avoidance. Using mutual information as an information-theoretic measure, an informative path which reduces the uncertainty of the environment is generated. Simulation results show that GP map combined with RRT planner can achieve the 3D navigation and exploration task successfully in unknown and complex environments.

References

- F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, H. Durrant-Whyte, Information based adaptive robotic exploration, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, 2002
- D.T. Cole, A cooperative UAS architecture for information-theoretic search and track, Ph.D. thesis, The University of Sydney, 2009
- D. Ferguson, A. Stentz, Using interpolation to improve path planning: the field D* algorithm. *J. Field Robot.* **23**(2), 79–101 (2006)
- E.W. Frew, Cooperative Standoff tracking of uncertain moving targets using active robot networks, in *IEEE International Conference on Robotics and Automation*, Roma, 2007a

- E.W. Frew, Information-theoretic integration of sensing and communication for active robot networks, in *International Conference on Robot Communication and Coordination*, Athens, 2007b
- R. Grabowski, R. Weatherly, R. Bolling, D. Seidel, M. Shadid, A. Jones, MITRE meteor: an off-road autonomous vehicle for DARPA's grand challenge. *J. Field Robot.* **23**(9), 811–835 (2006)
- B. Grocholsky, Information-theoretic control of multiple sensor platforms, Ph.D. thesis, The University of Sydney, 2002
- A. Krause, A. Singh, C. Guestrin, Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.* **9**, 235–284 (2008)
- S.M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, 2006)
- M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Thrun, Junior: the Stanford entry in the urban challenge. *J. Field Robot.* **25**(9), 569–597 (2008)
- S. OCallaghan, F.T. Ramos, H. Durrant-Whyte, Contextual occupancy maps using Gaussian processes, in *IEEE International Conference on Robotics and Automation*, Kobe, 2009
- C. Rasmussen, C. Williams, *Gaussian Processes for Machine Learning* (MIT, Cambridge, 2006)
- A.D. Ryan, H. Durrant-Whyte, J.K. Hedrick, Information-theoretic sensor motion control for distributed estimation, in *International Mechanical Engineering Congress and Exposition*, Seattle, 2007
- S. Scherer, S. Singh, L. Chamberlain, M. Elgersma, Flying fast and low among obstacles: methodology and experiments. *Int. J. Robot. Res.* **27**(5), 549–574 (2008)
- A. Singh, F. Ramos, H. Durrant-Whyte, W. Kaiser, Modeling and decision making in spatio-temporal process for environmental surveillance, in *IEEE International Conference on Robotics and Automation*, Anchorage, 2010
- C. Stachniss, G. Grisetti, W. Burgard, Information gain-based exploration using Rao-Blackwellized particle filters, in *Robotics: Science and Systems* (MIT, Cambridge, 2005)

Implementing Dubins Airplane Paths on Fixed-Wing UAVs*

68

Mark Owen, Randal W. Beard, and Timothy W. McLain

Contents

| | | |
|------------|---|------|
| 68.1 | Introduction | 1678 |
| 68.2 | Equations of Motion for the Dubins Airplane | 1680 |
| 68.3 | 3D Vector-Field Path Following | 1682 |
| 68.3.1 | Vector-Field Methodology | 1683 |
| 68.3.2 | Straight-Line Paths | 1684 |
| 68.3.3 | Helical Paths | 1686 |
| 68.4 | Minimum-Distance Airplane Paths | 1687 |
| 68.4.1 | Dubins Car Paths | 1688 |
| 68.4.2 | Dubins Airplane Paths | 1690 |
| 68.4.3 | Path Manager for Dubins Airplane | 1696 |
| 68.4.4 | Simulation Results | 1696 |
| 68.5 | Conclusion | 1697 |
| References | | 1699 |

*Contributed Chapter to the Springer Handbook for Unmanned Aerial Vehicles

M. Owen
Sandia National Labs, Albuquerque, NM, USA

R.W. Beard (✉)
Electrical and Computer Engineering Department, Brigham Young University,
Provo, UT, USA
e-mail: beard@byu.edu

T.W. McLain
Mechanical Engineering Department, Brigham Young University, Provo, UT, USA
e-mail: mclain@byu.edu

Abstract

A well-known path-planning technique for mobile robots or planar aerial vehicles is to use Dubins paths, which are minimum-distance paths between two configurations subject to the constraints of the Dubins car model. An extension of this method to a three-dimensional Dubins airplane model has recently been proposed. This chapter builds on that work showing a complete architecture for implementing Dubins airplane paths on small fixed-wing UAVs. The existing Dubins airplane model is modified to be more consistent with the kinematics of a fixed-wing aircraft. The chapter then shows how a recently proposed vector-field method can be used to design a guidance law that causes the Dubins airplane model to follow straight-line and helical paths. Dubins airplane paths are more complicated than Dubins car paths because of the altitude component. Based on the difference between the altitude of the start and end configurations, Dubins airplane paths can be classified as low, medium, or high altitude gain. While for medium and high altitude gain there are many different Dubins airplane paths, this chapter proposes selecting the path that maximizes the average altitude throughout the maneuver. The proposed architecture is implemented on a six degree-of-freedom Matlab/Simulink simulation of an Aerosonde UAV, and results from this simulation demonstrate the effectiveness of the technique.

68.1 Introduction

Unmanned aerial vehicles (UAVs) are used for a wide variety of tasks that require the UAV to be flown from one particular pose (position and attitude) to another. Most commonly, UAVs are flown from their current position and heading angle to a new desired position and heading angle. The ability to fly from one pose (or waypoint) to another is the fundamental building block upon which more sophisticated UAV navigation capabilities are built. For UAV missions involving sensors, the ability to position and orient the sensor over time is critically important. Example applications include wildlife observation and tracking, infrastructure monitoring (Rathinam et al. 2005; Frew et al. 2004; Egbert and Beard 2011), communication relays (Frew et al. 2009), meteorological measurements (Elston et al. 2010), and aerial surveillance (Rahmani et al. 2010; Elston and Frew 2008; Spry et al. 2005). Positioning and orienting the sensor is accomplished in part by planning and following paths through or above the sensing domain. Two-dimensional (2D) path planning and following at a constant altitude through unobstructed airspace is common, but as mission scenarios become increasingly sophisticated by requiring flight in three-dimensional (3D) terrain, the need for full 3D planning and guidance algorithms is becoming increasingly important.

For a vehicle that moves in a 2D plane at constant forward speed with a finite turn-rate constraint, the minimum-distance path between two configurations is termed a Dubins path. The initial and final configurations are defined by a 2D position in the plane of motion and an orientation. It has been shown that the optimal Dubins path in the absence of wind is composed of a constant radius turn, followed by a straight-line path, followed by another constant radius

turn (Dubins 1957). A vehicle that follows Dubins paths is often termed a Dubins car. There have been a wide variety of path-planning techniques proposed for mobile robots based on the Dubins car model (e.g., Hanson et al. 2011; Balluchi et al. 1996; Anderson and Milutinovic 2011; Karaman and Frazzoli 2011; Cowlagi and Tsiotras 2009; Yong and Barth 2006). The Dubins car model has also been used extensively for UAV applications by constraining the air vehicle to fly at a constant altitude (e.g., Yu and Beard 2013; Sujit et al. 2007; Yang and Kapila 2002; Shima et al. 2007).

The Dubins car model was recently extended to three dimensions to create the Dubins airplane model, where in addition to turn-rate constraints, a climb-rate constraint was added (Hosak and Ghose 2010; Chitsaz and LaValle 2007; Rahmani et al. 2010). Minimum-distance paths for the Dubins airplane were also derived in Chitsaz and LaValle (2007), using the Pontryagin minimum principle (Lewis 1986). However, in Chitsaz and LaValle (2007) some practical issues were not considered, leaving a gap between the theory and implementation on actual UAVs. The purpose of this chapter is to fill that gap. In particular, alternative equations of motion for the Dubins airplane model that include airspeed, flight-path angle, and bank angle are given. The kinematic equations of motion presented in this chapter are standard in the aerospace literature. The chapter also describes how to implement Dubins airplane paths using low-level autopilot loops, vector-field guidance laws for following straight lines and helices, and mode switching between the guidance laws.

In addition to the complexity of a third dimension of motion, Dubins airplane paths are more complicated than Dubins car paths. In particular, when the altitude component of the path falls within a specific range, there are an infinite number of paths that satisfy the minimum-distance objective. This chapter also proposes specific choices for paths that make practical sense for many UAV mission scenarios. Specifically, the path that also maximizes the average altitude of the path is selected.

The software architecture proposed in this chapter is similar to that discussed in Beard and McLain (2012) and is shown in Fig. 68.1. At the lowest level is the fixed-wing UAV. A state estimator processes sensors and produces the estimates of the state required for each of the higher levels. A low-level autopilot accepts airspeed, flight-path angle, and bank angle commands. The commands for the low-level autopilot are produced by a vector-field guidance law for following either straight-line paths or helical paths. The minimum-distance Dubins airplane path between two configurations is computed by the path manager, which also switches between commanding straight-line paths and helical paths.

This chapter will not discuss state estimation using the available sensors, but the interested reader is referred to Beard and McLain (2012) and other publications on state estimation for UAVs (see, e.g., Mahony et al. 2008; Misawa and Hedrick 1989; Brunke and Campbell 2004). Section 68.2 briefly describes assumptions on the unmanned aircraft and how the low-level autopilot is configured to produce the Dubins airplane kinematic model. Section 68.3 describes the vector-field guidance strategy used in this chapter for following both straight lines and helices. The Dubins airplane paths and the path manager used to follow them are discussed in Sect. 68.4. Section 68.5 offers concluding remarks.

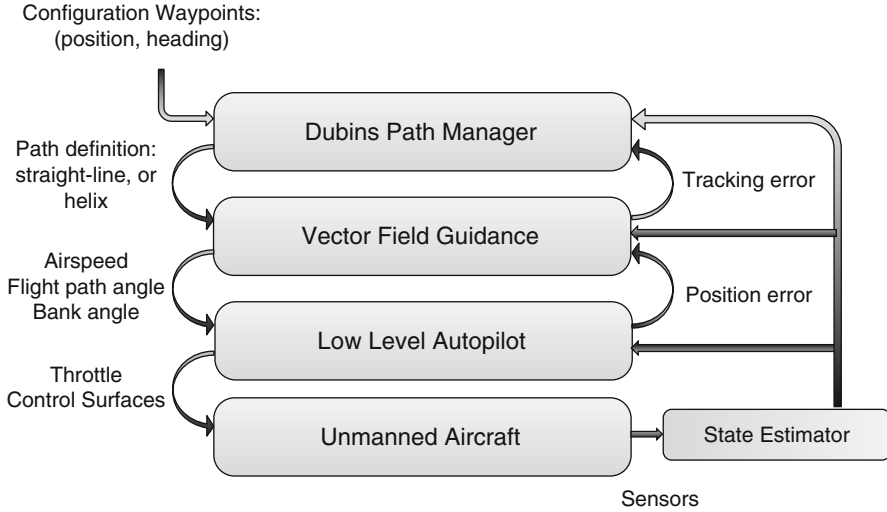


Fig. 68.1 Flight control architecture proposed in this chapter

68.2 Equations of Motion for the Dubins Airplane

Unmanned aircraft, particularly smaller systems, fly at relatively low airspeeds causing wind to have a significant effect on their performance. Since wind effects are not known prior to the moment they act on an aircraft, they are typically treated as a disturbance to be rejected in real time by the flight control system, rather than being considered during the path planning. It has been shown that vector-field-based path following methods, such as those employed in this chapter, are particularly effective at rejecting wind disturbances (Nelson et al. 2007). Treating wind as a disturbance also allows paths to be planned relative to the inertial environment, which is important as UAVs are flown in complex 3D terrain. Accordingly, when the Dubins airplane model is used for path planning, the effects of wind are not accounted for when formulating the equations of motion. In this case, the airspeed V is the same as the groundspeed, the heading angle ψ is the same as the course angle (assuming zero sideslip angle), and the flight-path angle γ is the same as the air-mass-referenced flight-path angle (Beard and McLain 2012).

Figure 68.2 depicts a UAV flying with airspeed V , heading angle ψ , and flight-path angle γ . Denoting the inertial position of the UAV as $(r_n, r_e, r_d)^T$, the kinematic relationship between the inertial velocity, $\mathbf{v} = (\dot{r}_n, \dot{r}_e, \dot{r}_d)^T$, and the airspeed, heading angle, and flight-path angle can be easily visualized as

$$\begin{pmatrix} \dot{r}_n \\ \dot{r}_e \\ \dot{r}_d \end{pmatrix} = \begin{pmatrix} V \cos \psi \cos \gamma \\ V \sin \psi \cos \gamma \\ -V \sin \gamma \end{pmatrix},$$

where $V = \|\mathbf{v}\|$.

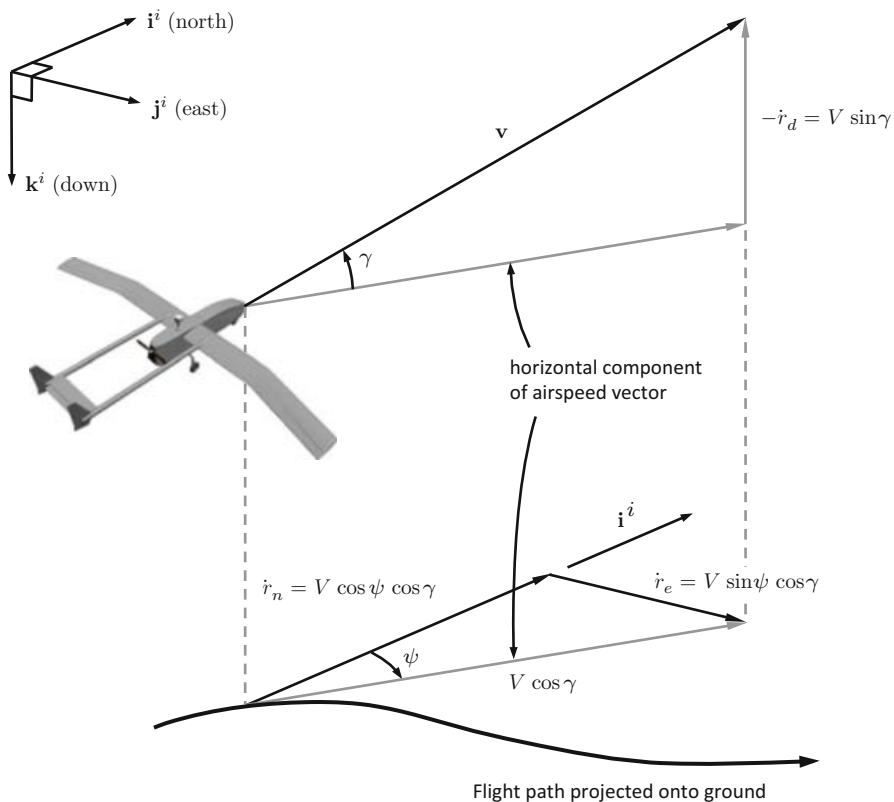


Fig. 68.2 Graphical representation of aircraft kinematic model

This chapter assumes that a low-level autopilot regulates the airspeed V to a constant commanded value V^c , the flight-path angle γ to the commanded flight-path angle γ^c , and the bank angle ϕ to the commanded bank angle ϕ^c . In addition, the dynamics of the flight-path angle and bank angle loops are assumed to be sufficiently fast that they can be ignored for the purpose of path following. The relationship between the heading angle ψ and the bank angle ϕ is given by the coordinated turn condition (Beard and McLain 2012)

$$\dot{\psi} = \frac{g}{V} \tan \phi,$$

where g is the acceleration due to gravity.

Under the assumption that the autopilot is well tuned and the airspeed, flight-path angle, and bank angle states converge with the desired response to their commanded values, then the following kinematic model is a good description of the UAV motion:

$$\begin{aligned}
\dot{r}_n &= V \cos \psi \cos \gamma^c \\
\dot{r}_e &= V \sin \psi \cos \gamma^c \\
\dot{r}_d &= -V \sin \gamma^c \\
\dot{\psi} &= \frac{g}{V} \tan \phi^c.
\end{aligned} \tag{68.1}$$

Physical capabilities of the aircraft place limits on the achievable bank and flight-path angles that can be commanded. These physical limits on the aircraft are represented by the following constraints:

$$|\phi^c| \leq \bar{\phi} \tag{68.2}$$

$$|\gamma^c| \leq \bar{\gamma}. \tag{68.3}$$

The kinematic model given by (68.1) with the input constraints (68.2) and (68.3) will be referred to as the Dubins airplane. This model builds upon the model originally proposed for the Dubins airplane in Chitsaz and LaValle (2007), which is given by

$$\begin{aligned}
\dot{r}_n &= V \cos \psi \\
\dot{r}_e &= V \sin \psi \\
\dot{r}_d &= u_1 \quad |u_1| \leq 1 \\
\dot{\psi} &= u_2 \quad |u_2| \leq 1.
\end{aligned} \tag{68.4}$$

Although (68.1) is similar to (68.4), it captures the aircraft kinematics with greater accuracy and provides greater insight into the aircraft behavior and is more consistent with commonly used aircraft guidance models. Note, however, that (68.1) is only a kinematic model that does not include aerodynamics, wind effects, or engine/thrust limits. While it is not sufficiently accurate for low-level autopilot design, it is well suited for high-level path-planning and path-following control design. In-depth discussions of aircraft dynamic models can be found in Phillips (2010), Stevens and Lewis (2003), Nelson (1998) and Yechout et al. (2003).

68.3 3D Vector-Field Path Following

This section shows how to develop guidance laws to ensure that the kinematic model (68.1) follows straight-line and helical paths. Section 68.4 shows how straight-line and helical paths are combined to produce minimum-distance paths between start and end configurations.

68.3.1 Vector-Field Methodology

The guidance strategy will use the vector-field methodology proposed in Gonçalves et al. (2010), and this section provides a brief overview. The path to be followed in \mathbb{R}^3 is specified as the intersection of two two-dimensional manifolds given by $\alpha_1(\mathbf{r}) = 0$ and $\alpha_2(\mathbf{r}) = 0$ where α_1 and α_2 have bounded second partial derivatives, and where $\mathbf{r} \in \mathbb{R}^3$. An underlying assumption is that the path given by the intersection is connected and one-dimensional. Defining the function

$$V(\mathbf{r}) = \frac{1}{2}\alpha_1^2(\mathbf{r}) + \frac{1}{2}\alpha_2^2(\mathbf{r})$$

gives

$$\frac{\partial V}{\partial \mathbf{r}} = \alpha_1(\mathbf{r}) \frac{\partial \alpha_1}{\partial \mathbf{r}}(\mathbf{r}) + \alpha_2(\mathbf{r}) \frac{\partial \alpha_2}{\partial \mathbf{r}}(\mathbf{r}).$$

Note that $-\frac{\partial V}{\partial \mathbf{r}}$ is a vector that points toward the path. Therefore following $-\frac{\partial V}{\partial \mathbf{r}}$ will transition the Dubins airplane onto the path. However, simply following $-\frac{\partial V}{\partial \mathbf{r}}$ is insufficient since the gradient is zero on the path. When the Dubins airplane is on the path, its direction of motion should be perpendicular to both $\frac{\partial \alpha_1}{\partial \mathbf{r}}$ and $\frac{\partial \alpha_2}{\partial \mathbf{r}}$. Following Gonçalves et al. (2010) the desired velocity vector $\mathbf{u}' \in \mathbb{R}^3$ can be chosen as

$$\mathbf{u}' = -K_1 \frac{\partial V}{\partial \mathbf{r}} + K_2 \frac{\partial \alpha_1}{\partial \mathbf{r}} \times \frac{\partial \alpha_2}{\partial \mathbf{r}}, \quad (68.5)$$

where K_1 and K_2 are symmetric tuning matrices. It is shown in Gonçalves et al. (2010) that the dynamics $\dot{\mathbf{r}} = \mathbf{u}'$, where \mathbf{u}' is given by Eq. (68.5), results in \mathbf{r} asymptotically converging to a trajectory that follows the intersection of α_1 and α_2 if K_1 is positive definite, and where the definiteness of K_2 determines the direction of travel along the trajectory.

The problem with Eq. (68.5) is that the magnitude of the desired velocity \mathbf{u}' may not equal V , the velocity of the Dubins airplane. Therefore \mathbf{u}' is normalized as

$$\mathbf{u} = V \frac{\mathbf{u}'}{\|\mathbf{u}'\|}. \quad (68.6)$$

Fortunately, the stability proof in Gonçalves et al. (2010) is still valid when \mathbf{u}' is normalized as in Eq. (68.6).

Setting the NED components of the velocity of the Dubins airplane model given in Eq. (68.1) to $\mathbf{u} = (u_1, u_2, u_3)^\top$ gives

$$V \cos \psi^d \cos \gamma^c = u_1$$

$$V \sin \psi^d \cos \gamma^c = u_2$$

$$-V \sin \gamma^c = u_3.$$

Solving for the commanded flight-path angle γ^c and the desired heading angle ψ^d results in the expressions

$$\begin{aligned}\gamma^c &= -\text{sat}_{\bar{\gamma}} \left[\sin^{-1} \left(\frac{u_3}{V} \right) \right] \\ \psi^d &= \text{atan2}(u_2, u_1),\end{aligned}\tag{68.7}$$

where atan2 is the four quadrant inverse tangent and where the saturation function is defined as

$$\text{sat}_a[x] = \begin{cases} a & \text{if } x \geq a \\ -a & \text{if } x \leq -a \\ x & \text{otherwise} \end{cases}.$$

Assuming the inner-loop lateral-directional dynamics are accurately modeled by the coordinated-turn equation, roll-angle commands yielding desirable turn performance can be obtained from the expression

$$\phi^c = \text{sat}_{\bar{\phi}} [k_{\phi}(\psi^d - \psi)],\tag{68.8}$$

where k_{ϕ} is a positive constant.

Sections 68.3.2 and 68.3.3 apply the framework described in this section to straight-line following and helix following, respectively.

68.3.2 Straight-Line Paths

A straight-line path is described by the direction of the line and a point on the line. Let $\mathbf{c}_{\ell} = (c_n, c_e, c_d)^{\top}$ be an arbitrary point on the line, and let the direction of the line be given by the desired heading angle from north ψ_{ℓ} and the desired flight-path angle γ_{ℓ} measured from the inertial north-east plane. Therefore,

$$\mathbf{q}_{\ell} = \begin{pmatrix} q_n \\ q_e \\ q_d \end{pmatrix} \triangleq \begin{pmatrix} \cos \psi_{\ell} \cos \gamma_{\ell} \\ \sin \psi_{\ell} \cos \gamma_{\ell} \\ -\sin \gamma_{\ell} \end{pmatrix}$$

is a unit vector that points in the direction of the desired line. The straight-line path is given by

$$\mathcal{P}_{\text{line}}(\mathbf{c}_{\ell}, \psi_{\ell}, \gamma_{\ell}) = \{\mathbf{r} \in \mathbb{R}^3 : \mathbf{r} = \mathbf{c}_{\ell} + \sigma \mathbf{q}_{\ell}, \sigma \in \mathbb{R}\}.\tag{68.9}$$

A unit vector that is perpendicular to the longitudinal plane defined by \mathbf{q}_{ℓ} is given by

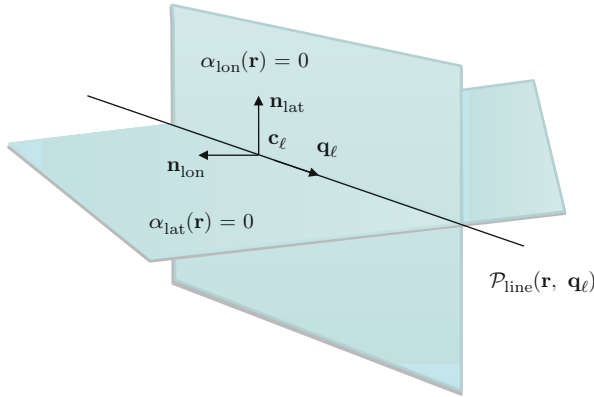


Fig. 68.3 This figure shows how the straight-line path $\mathcal{P}_{\text{line}}(\mathbf{c}_\ell, \psi_\ell, \gamma_\ell)$ is defined by the intersection of the two surfaces given by $\alpha_{\text{lon}}(\mathbf{r}) = 0$ and $\alpha_{\text{lat}}(\mathbf{r}) = 0$

$$\mathbf{n}_{\text{lon}} \triangleq \begin{pmatrix} -\sin \psi_\ell \\ \cos \psi_\ell \\ 0 \end{pmatrix}.$$

Similarly, a unit vector that is perpendicular to the lateral plane defined by \mathbf{q}_ℓ is given by

$$\mathbf{n}_{\text{lat}} \triangleq \mathbf{n}_{\text{lon}} \times \mathbf{q}_\ell = \begin{pmatrix} -\cos \psi_\ell \sin \gamma_\ell \\ -\sin \psi_\ell \sin \gamma_\ell \\ -\cos \gamma_\ell \end{pmatrix}.$$

It follows that $\mathcal{P}_{\text{line}}$ is given by the intersection of the surfaces defined by

$$\alpha_{\text{lon}}(\mathbf{r}) \triangleq \mathbf{n}_{\text{lon}}^\top (\mathbf{r} - \mathbf{c}_\ell) = 0 \quad (68.10)$$

$$\alpha_{\text{lat}}(\mathbf{r}) \triangleq \mathbf{n}_{\text{lat}}^\top (\mathbf{r} - \mathbf{c}_\ell) = 0. \quad (68.11)$$

Figure 68.3 shows \mathbf{q}_ℓ , \mathbf{c}_ℓ , and the surfaces defined by $\alpha_{\text{lon}}(\mathbf{r}) = 0$ and $\alpha_{\text{lat}}(\mathbf{r}) = 0$.

The gradients of α_{lon} and α_{lat} are given by

$$\begin{aligned} \frac{\partial \alpha_{\text{lon}}}{\partial \mathbf{r}} &= \mathbf{n}_{\text{lon}} \\ \frac{\partial \alpha_{\text{lat}}}{\partial \mathbf{r}} &= \mathbf{n}_{\text{lat}}. \end{aligned}$$

Therefore, before normalization, the desired velocity vector is given by

$$\mathbf{u}'_{\text{line}} = K_1 (\mathbf{n}_{\text{lon}} \mathbf{n}_{\text{lon}}^\top + \mathbf{n}_{\text{lat}} \mathbf{n}_{\text{lat}}^\top) (\mathbf{r} - \mathbf{c}_\ell) + K_2 (\mathbf{n}_{\text{lon}} \times \mathbf{n}_{\text{lat}}). \quad (68.12)$$

68.3.3 Helical Paths

A time-parameterized helical path is given by

$$\mathbf{r}(t) = \mathbf{c}_h + \begin{pmatrix} R_h \cos(\lambda_h t + \psi_h) \\ R_h \sin(\lambda_h t + \psi_h) \\ -t R_h \tan \gamma_h \end{pmatrix}, \quad (68.13)$$

where $\mathbf{r}(t) = \begin{pmatrix} r_n \\ r_e \\ r_d \end{pmatrix}(t)$ is the position along the path, $\mathbf{c}_h = (c_n, c_e, c_d)^\top$ is the center of the helix, and the initial position of the helix is

$$\mathbf{r}(0) = \mathbf{c}_h + \begin{pmatrix} R_h \cos \psi_h \\ R_h \sin \psi_h \\ 0 \end{pmatrix}$$

and where R_h is the radius, $\lambda_h = +1$ denotes a clockwise helix ($N \rightarrow E \rightarrow S \rightarrow W$), and $\lambda_h = -1$ denotes a counterclockwise helix ($N \rightarrow W \rightarrow S \rightarrow E$), and where γ_h is the desired flight-path angle along the helix.

To find two surfaces that define the helical path, the time parameterization in (68.13) needs to be eliminated. Equation (68.13) gives

$$(r_n - c_n)^2 + (r_e - c_e)^2 = R_h^2.$$

In addition, divide the east component of $\mathbf{r} - \mathbf{c}_h$ by the north component to get

$$\tan(\lambda_h t + \psi_h) = \frac{r_e - c_e}{r_n - c_n}$$

Solving for t and plugging into the third component of (68.13) gives

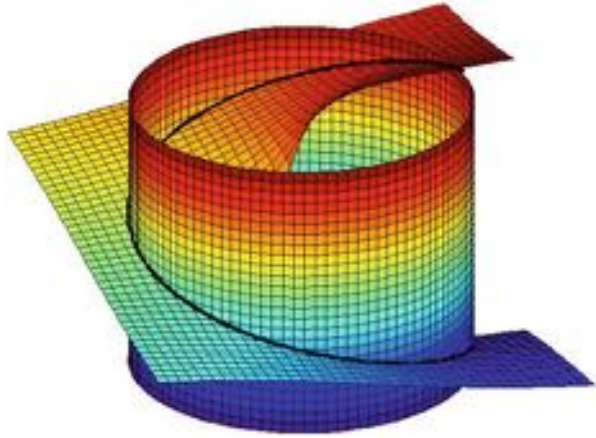
$$r_d - c_d = -\frac{R_h \tan \gamma_h}{\lambda_h} \left(\tan^{-1} \left(\frac{r_e - c_e}{r_n - c_n} \right) - \psi_h \right).$$

Therefore, normalizing these equations by R_h results in

$$\begin{aligned} \alpha_{\text{cyl}}(\mathbf{r}) &= \left(\frac{r_n - c_n}{R_h} \right)^2 + \left(\frac{r_e - c_e}{R_h} \right)^2 - 1 \\ \alpha_{\text{pl}}(\mathbf{r}) &= \left(\frac{r_d - c_d}{R_h} \right) + \frac{\tan \gamma_h}{\lambda_h} \left(\tan^{-1} \left(\frac{r_e - c_e}{r_n - c_n} \right) - \psi_h \right). \end{aligned}$$

Normalization by R_h makes the gains on the resulting control strategy invariant to the size of the orbit.

Fig. 68.4 A helical path for parameters $\mathbf{c}_h = (0, 0, 0)^\top$, $R_h = 30$ m, $\gamma_h = \frac{15\pi}{180}$ rad, and $\lambda_h = +1$



A helical path is then defined as

$$\mathcal{P}_{\text{helix}}(\mathbf{c}_h, \psi_h, \lambda_h, R_h, \gamma_h) = \{\mathbf{r} \in \mathbb{R}^3 : \alpha_{\text{cyl}}(\mathbf{r}) = 0 \text{ and } \alpha_{\text{pl}}(\mathbf{r}) = 0\}. \quad (68.14)$$

The two surfaces $\alpha_{\text{cyl}}(\mathbf{r}) = 0$ and $\alpha_{\text{pl}}(\mathbf{r}) = 0$ are shown in Fig. 68.4 for parameters $\mathbf{c}_h = (0, 0, 0)^\top$, $R_h = 30$ m, $\gamma_h = \frac{15\pi}{180}$ rad, and $\lambda_h = +1$. The associated helical path is the intersection of the two surfaces.

The gradients of α_{cyl} and α_{pl} are given by

$$\begin{aligned} \frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} &= \left(2 \frac{r_n - c_n}{R_h}, 2 \frac{r_e - c_e}{R_h}, 0 \right)^\top \\ \frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} &= \left(\frac{\tan \gamma_h}{\lambda_h} \frac{-(r_e - c_e)}{(r_n - c_n)^2 + (r_e - c_e)^2}, \frac{\tan \gamma_h}{\lambda_h} \frac{(r_n - c_n)}{(r_n - c_n)^2 + (r_e - c_e)^2}, \frac{1}{R_h} \right)^\top. \end{aligned}$$

Before normalization, the desired velocity vector is given by

$$\mathbf{u}'_{\text{helix}} = K_1 \left(\alpha_{\text{cyl}} \frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} + \alpha_{\text{pl}} \frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} \right) + \lambda K_2 \left(\frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} \times \frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} \right), \quad (68.15)$$

where

$$\frac{\partial \alpha_{\text{cyl}}}{\partial \mathbf{r}} \times \frac{\partial \alpha_{\text{pl}}}{\partial \mathbf{r}} = \frac{2}{R_h} \left(\frac{r_e - c_e}{R_h}, -\frac{r_n - c_n}{R_h}, \lambda_h \tan \gamma_h \right)^\top.$$

68.4 Minimum-Distance Airplane Paths

This section describes how to concatenate straight-line and helix paths to produce minimum-distance paths between two configurations for the Dubins airplane model. A configuration is defined as the tuple (z_n, z_e, z_d, ψ) , where $(z_n, z_e, z_d)^\top$ is a

north-east-down position referenced to an inertial frame, and ψ is a heading angle measured from north. Given the kinematic model (68.1) subject to the constraints (68.2) and (68.3), a Dubins airplane path refers to a minimum-distance path between a start configuration $(z_{ns}, z_{es}, z_{ds}, \psi_s)$ and an end configuration $(z_{ne}, z_{ee}, z_{de}, \psi_e)$. Minimum-distance paths for the Dubins airplane are derived in Chitsaz and LaValle (2007) using the Pontryagin Maximum Principle for the dynamics given in (68.4) with constraints $\bar{\gamma} = 1$ and $\bar{\phi} = 1$. This section recasts the results from Chitsaz and LaValle (2007) using the standard aircraft kinematic model given in (68.1) using the constraints (68.2) and (68.3).

68.4.1 Dubins Car Paths

Minimum-distance paths for the Dubins airplane are closely related to minimum-distance paths for the Dubins car. This section briefly reviews Dubins car paths, which were originally developed in Dubins (1957) using the notation and methods defined in Beard and McLain (2012).

The Dubins car model is a subset of (68.4) given by

$$\begin{aligned}\dot{r}_n &= V \cos \psi \\ \dot{r}_e &= V \sin \psi \\ \dot{\psi} &= u,\end{aligned}\tag{68.16}$$

where $|u| \leq \bar{u}$. For the Dubins car, the minimum turn radius is given by

$$R_{\min} = V/\bar{u}.\tag{68.17}$$

The Dubins car path is defined as the minimum-distance path from the start configuration (z_{ns}, z_{es}, ψ_s) to the end configuration (z_{ne}, z_{ee}, ψ_e) . As shown in Dubins (1957), the minimum-distance path between two different configurations consists of a circular arc of radius R_{\min} that starts at the initial configuration, followed by a straight line, and concluding with another circular arc of radius R_{\min} that ends at the final configuration.

As shown in Fig. 68.5, for any given start and end configurations, there are four possible paths consisting of an arc, followed by a straight line, followed by an arc. RSR is a right-handed arc followed by a straight line followed by another right-handed arc. RSL is a right-handed arc followed by a straight line followed by a left-handed arc. LSR is a left-handed arc followed by a straight line followed by a right-handed arc. LSL is a left-handed arc followed by a straight line followed by another left-handed arc. The Dubins path is defined as the case with the shortest path length.

As explained in Beard and McLain (2012), the guidance algorithm for following a Dubins car path consists of switching between orbit following and straight-line following. Figure 68.6 shows the parameters that are required by the guidance

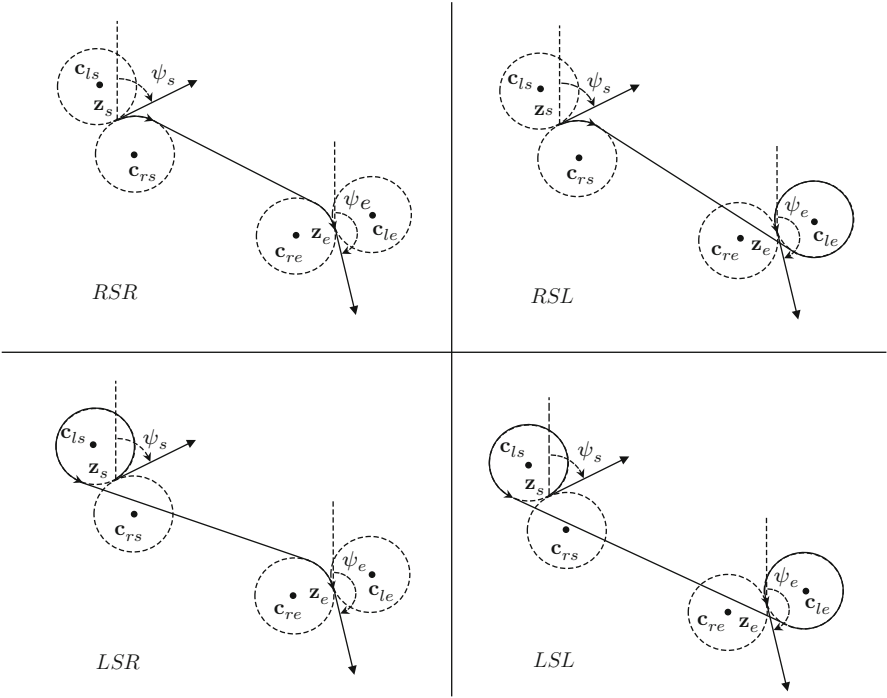


Fig. 68.5 Given a start configuration (z_{ns}, z_{es}, ψ_s) , an end configuration (z_{ne}, z_{ee}, ψ_e) , and a radius R , there are four possible paths consisting of an arc, a straight line, and an arc. The Dubins path is defined as the case that results in the shortest path length, which for this scenario is RSR

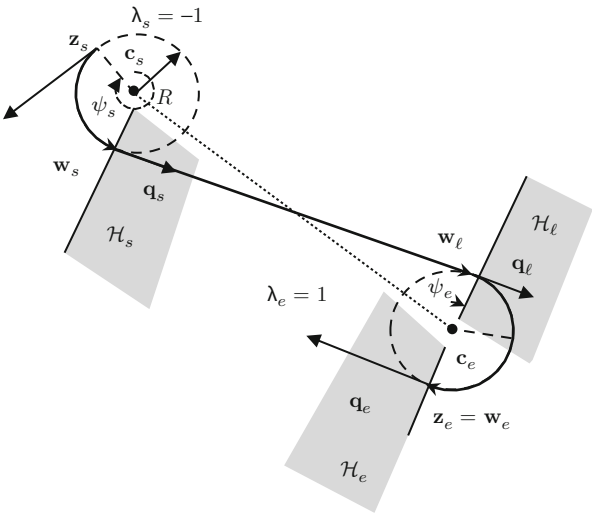


Fig. 68.6 The parameters that are required by the guidance algorithm to follow a Dubins car path

algorithm to follow a Dubins car path. Given that the vehicle configuration is close to the start configuration (\mathbf{z}_s, ψ_s) , the vehicle is initially commanded to follow an orbit with center \mathbf{c}_s and orbit direction λ_s . The orbit is followed until the vehicle crosses half-plane $\mathcal{H}_s(\mathbf{w}_s, \mathbf{q}_s)$, or in other words until its position \mathbf{r} satisfies

$$(\mathbf{r} - \mathbf{w}_s)^\top \mathbf{q}_s \geq 0,$$

where \mathbf{w}_s is a position on the half-plane and \mathbf{q}_s is a unit vector orthogonal to the half-plane. The vehicle then follows the straight line defined by $(\mathbf{w}_s, \mathbf{q}_s)$ until it crosses half-plane $\mathcal{H}_\ell(\mathbf{w}_\ell, \mathbf{q}_\ell)$. It then follows the orbit with center \mathbf{c}_e and direction λ_e until it crosses half-plane $\mathcal{H}_e(\mathbf{w}_e, \mathbf{q}_e)$ and completes the Dubins car path. Accordingly the parameters that define a Dubins car path are given by

$$\mathcal{D}_{\text{car}} = (R, \mathbf{c}_s, \lambda_s, \mathbf{w}_s, \mathbf{q}_s, \mathbf{w}_\ell, \mathbf{q}_\ell, \mathbf{c}_e, \lambda_e, \mathbf{w}_e, \mathbf{q}_e). \quad (68.18)$$

The length of the Dubins car path depends explicitly on the turning radius R and will be denoted as $L_{\text{car}}(R)$. Details of how to compute $L_{\text{car}}(R)$ as well as the parameters \mathcal{D}_{car} are given in Beard and McLain (2012).

68.4.2 Dubins Airplane Paths

Dubins airplane paths are more complicated than Dubins car paths because of the altitude component. As described in Chitsaz and LaValle (2007) there are three different cases for Dubins airplane paths that depend on the altitude difference between the start and end configuration, the length of the Dubins car path, and the flight-path limit $\bar{\gamma}$. The three cases are defined to be *low altitude*, *medium altitude*, and *high altitude*. In contrast to (68.17), the minimum turn radius for a Dubins airplane is given by

$$R_{\min} = \frac{V^2}{g} \tan \bar{\phi}. \quad (68.19)$$

The altitude gain between the start and end configuration is said to be *low altitude* if

$$|z_{de} - z_{ds}| \leq L_{\text{car}}(R_{\min}) \tan \bar{\gamma},$$

where the term on the right is the maximum altitude gain that can be obtained by flying at flight-path angle $\pm \bar{\gamma}$ for a distance of $L_{\text{car}}(R_{\min})$. The altitude gain is said to be *medium altitude* if

$$L_{\text{car}}(R_{\min}) \tan \bar{\gamma} < |z_{de} - z_{ds}| \leq [L_{\text{car}}(R_{\min}) + 2\pi R_{\min}] \tan \bar{\gamma},$$

where the addition of the term $2\pi R_{\min}$ accounts for adding one orbit at radius R_{\min} to the path length. The altitude gain is said to be *high altitude* if

$$|z_{de} - z_{ds}| > [L_{\text{car}}(R_{\min}) + 2\pi R_{\min}] \tan \bar{\gamma}.$$

The following three sections describe how Dubins car paths are modified to produce Dubins airplane paths for low, high, and medium-altitude cases.

68.4.2.1 Low-Altitude Dubins Paths

In the low-altitude case, the altitude gain between the start and end configurations can be achieved by flying the Dubins car path with a flight-path angle satisfying constraint (68.3). Therefore, the optimal flight-path angle can be computed by

$$\gamma^* = \tan^{-1} \left(\frac{|z_{de} - z_{ds}|}{L_{\text{car}}(R_{\min})} \right).$$

The length of the Dubins airplane path is given by

$$L_{\text{air}}(R_{\min}, \gamma^*) = \frac{L_{\text{car}}(R_{\min})}{\cos \gamma^*}.$$

The parameters required to define a low-altitude Dubins airplane path are the same parameters for the Dubins car given in (68.18) with the addition of the optimal flight-path angle γ^* and the angles of the start and end helices ψ_s and ψ_e . Note that for the Dubins car path ψ_s and ψ_e are not required since the orbit is flat and does not have a starting location. However, as described in Sect. 68.3.3, to follow a helix, the start angle is required. Figure 68.7 shows several Dubins airplane paths for the low-altitude case where the altitude difference is 25 m over a typical Dubins car path length of 180 m.

68.4.2.2 High-Altitude Dubins Paths

In the high-altitude case, the altitude gain cannot be achieved by flying the Dubins car path within the flight-path angle constraints. As shown in Chitsaz and LaValle (2007), the minimum-distance path is achieved when the flight-path angle is set at its limit of $\pm\bar{\gamma}$, and the Dubins car path is extended to facilitate the altitude gain. While there are many different ways to extend the Dubins car path, this chapter extends the path by spiraling a certain number of turns at the beginning or end of the path and then by increasing the turn radius by the appropriate amount.

For UAV scenarios, the most judicious strategy is typically to spend most of the trajectory at as high an altitude as possible. Therefore, if the altitude at the end configuration is higher than the altitude at the start configuration, then the path will be extended by a climbing helix at the beginning of the path, as shown in the RSR and RSL cases in Fig. 68.8. If on the other hand, the altitude at the start configuration is higher than the end configuration, then the path will be extended by a descending helix at the end of the path, as shown in the LSR and LSL cases in Fig. 68.8. If multiple turns around the helix are required, then the turns could be split between the start and end helices and still result in the same path length. For high-altitude Dubins paths, the required number of turns in the helix will be the smallest integer k such that

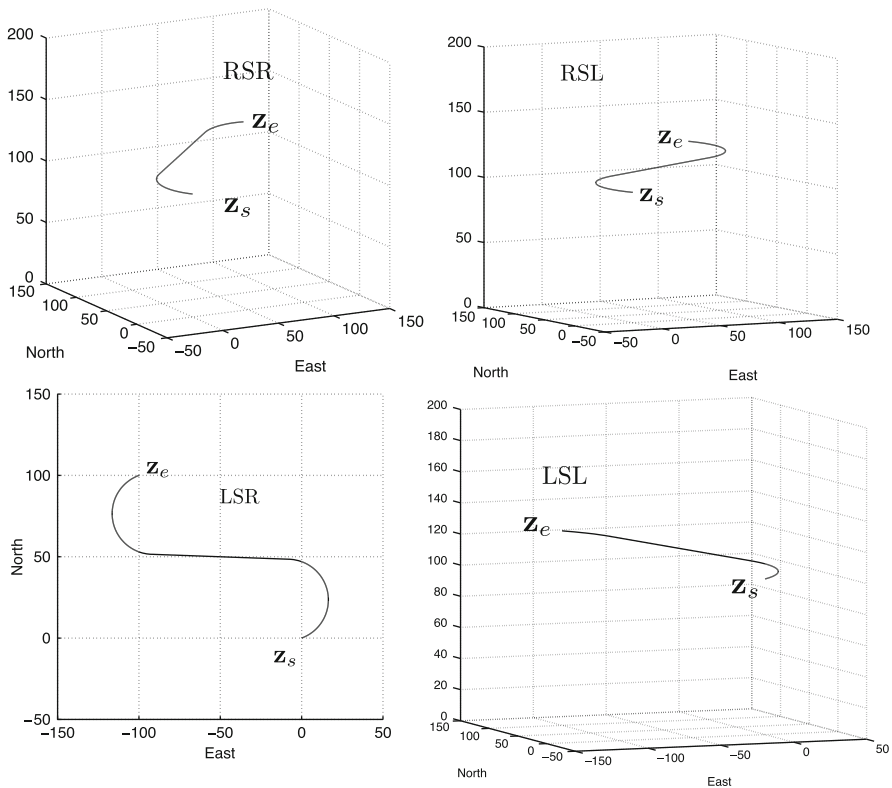


Fig. 68.7 Dubins airplane paths for the low-altitude case

$$(L_{\text{car}}(R_{\min})+2\pi kR_{\min})\tan\bar{\gamma}\leq|z_{de}-z_{ds}|<(L_{\text{car}}(R_{\min})+2\pi(k+1)R_{\min})\tan\bar{\gamma},$$

or in other words

$$k=\left\lfloor\frac{1}{2\pi R_{\min}}\left(\frac{|z_{de}-z_{ds}|}{\tan\bar{\gamma}}-L_{\text{car}}(R_{\min})\right)\right\rfloor,$$

where $\lfloor x \rfloor$ is the floor function that rounds x down to the nearest integer. The radius of the start and end helices is then increased to R^* so that

$$(L_{\text{car}}(R^*)+2\pi kR^*)\tan\bar{\gamma}=|z_{de}-z_{ds}|. \tag{68.20}$$

A bisection search is used to find R^* satisfying (68.20). The resulting path is a minimum-distance Dubins airplane path with path length

$$L_{\text{air}}(R^*,\bar{\gamma})=\frac{L_{\text{car}}(R^*)}{\cos\bar{\gamma}}.$$

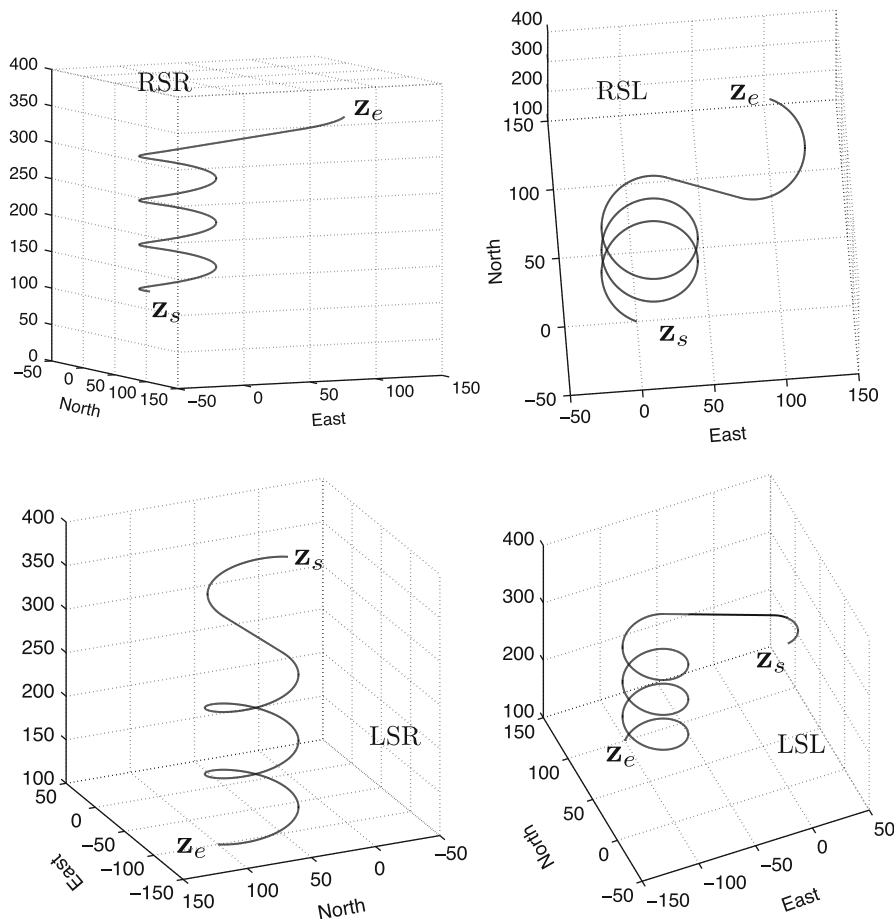


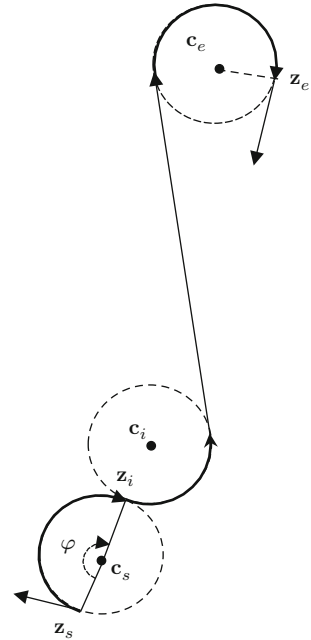
Fig. 68.8 Dubins airplane paths for the high-altitude case

The parameters required to define a high-altitude Dubins airplane path are the same parameters for the Dubins car given in (68.18) with R_{\min} replaced by R^* , the addition of the optimal flight-path angle $\pm \bar{\gamma}$, the additions of the start and end angles ψ_s and ψ_e , and the addition of the required number of turns at the start helix k_s and the required number of turns at the end helix k_e . Figure 68.8 shows several Dubins airplane paths for the high-altitude case where the altitude difference is 300 m over a typical Dubins car path length of 180 m.

68.4.2.3 Medium-Altitude Dubins Paths

In the medium-altitude case, the altitude difference between the start and end configurations is too large to obtain by flying the Dubins car path at the flight-path

Fig. 68.9 The start position of the intermediate arc is found by varying φ



angle constraint, but small enough that adding a full turn on the helix at the beginning or end of the path and flying so that $\gamma = \pm \bar{\gamma}$ results in more altitude gain than is needed. As shown in Chitsaz and LaValle (2007), the minimum-distance path is achieved by setting $\gamma = \text{sign}(z_{de} - z_{ds}) \bar{\gamma}$ and inserting an extra maneuver in the Dubins car path that extends the path length so that the altitude gain when $\gamma = \pm \bar{\gamma}$ is exactly $z_{de} - z_{ds}$. While there are numerous possible ways to extend the path length, the method proposed in this chapter is to add an additional intermediate arc to the start or end of the path, as shown in Fig. 68.9. If the start altitude is lower than the end altitude, then the intermediate arc is inserted immediately after the start helix, as shown for cases RLSR and RLSL in Fig. 68.11. If on the other hand, the start altitude is higher than the end altitude, then the intermediate arc is inserted immediately before the end helix, as shown for cases LSLR and LSRL in Fig. 68.11.

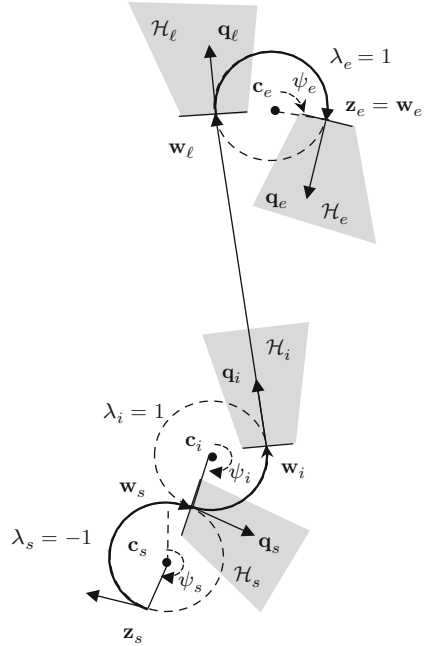
To find the Dubins path in the medium-altitude case, the position of the intermediate arc is parameterized by φ as shown in Fig. 68.9, where

$$\mathbf{z}_i = \mathbf{c}_s + R(\varphi)(\mathbf{z}_s - \mathbf{c}_s).$$

A standard Dubins car path is planned from configuration $(\mathbf{z}_i, \psi_s + \varphi)$ to the end configuration, and the new path length is given by

$$L(\varphi) = \varphi R_{\min} + L_{\text{car}}(\mathbf{z}_i, \psi_s + \varphi, \mathbf{z}_e, \psi_e).$$

Fig. 68.10 Parameters that define Dubins airplane path when an intermediate arc is inserted



A bisection search algorithm is used to find the angle φ^* so that

$$L(\varphi^*) \tan \bar{\gamma} = |z_{de} - z_{ds}|.$$

The length of the corresponding Dubins airplane path is given by

$$L_{\text{air}} = \frac{L(\varphi^*)}{\cos \bar{\gamma}}.$$

The parameters needed to describe the Dubins airplane path for the medium-altitude case are shown in Fig. 68.10. The introduction of an intermediate arc requires the additional parameters \mathbf{c}_i , ψ_i , λ_i , \mathbf{w}_i , and \mathbf{q}_i . Therefore, in analogy to (68.18), the parameters that define a Dubins airplane path are

$$\mathcal{D}_{\text{air}} = (R, \gamma, \mathbf{c}_s, \psi_s, \lambda_s, \mathbf{w}_s, \mathbf{q}_s, \mathbf{c}_i, \psi_i, \lambda_i, \mathbf{w}_i, \mathbf{q}_i, \mathbf{w}_\ell, \mathbf{q}_\ell, \mathbf{c}_e, \psi_e, \lambda_e, \mathbf{w}_e, \mathbf{q}_e). \quad (68.21)$$

Figure 68.11 shows several Dubins airplane paths for the medium-altitude case where the altitude difference is 100m over a typical Dubins car path length of 180m.

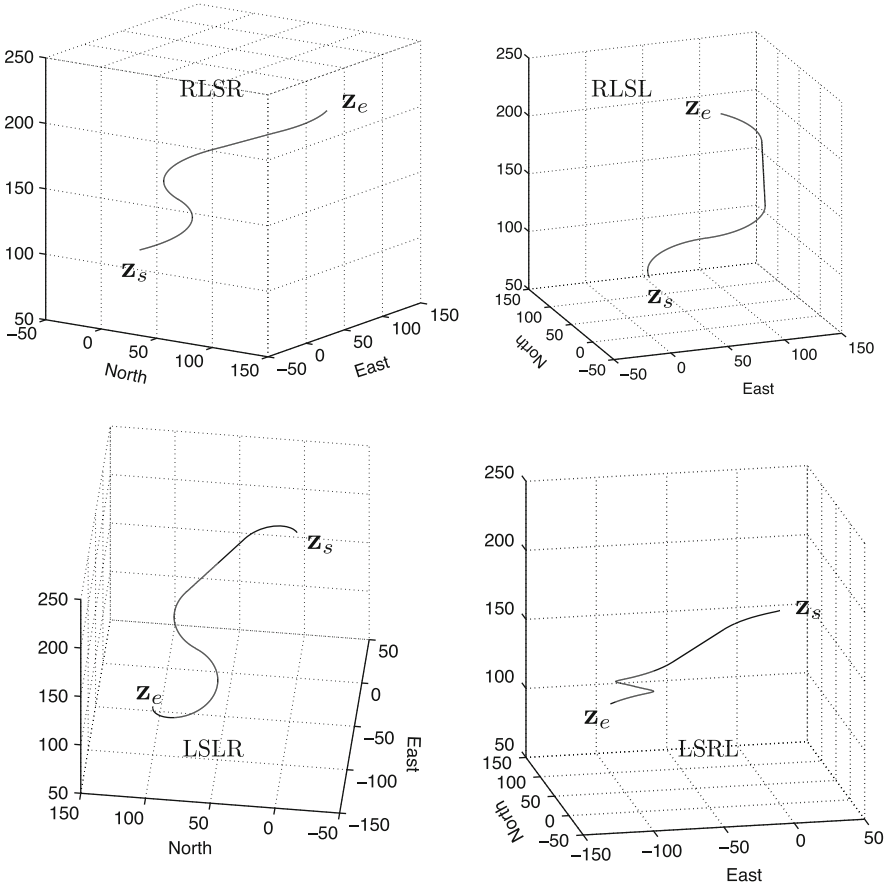


Fig. 68.11 Dubins airplane paths for the medium-altitude case

68.4.3 Path Manager for Dubins Airplane

The path manager for the Dubins airplane is shown in Fig. 68.12. With reference to (68.14), the start helix is defined as $\mathcal{P}_{\text{helix}}(\mathbf{c}_s, \psi_s, \lambda_s, R, \gamma)$. Similarly, the intermediate arc, if it exists, is defined by $\mathcal{P}_{\text{helix}}(\mathbf{c}_i, \psi_i, \lambda_i, R, \gamma)$, and the end helix is given by $\mathcal{P}_{\text{helix}}(\mathbf{c}_e, \psi_e, \lambda_e, R, \gamma)$. With reference to (68.9), the straight-line path is given by $\mathcal{P}_{\text{line}}(\mathbf{w}_\ell, \mathbf{q}_\ell)$.

68.4.4 Simulation Results

This section provides some simulation results where Dubins airplane paths are flown on a full six-degree-of-freedom UAV simulator. The aircraft used for the

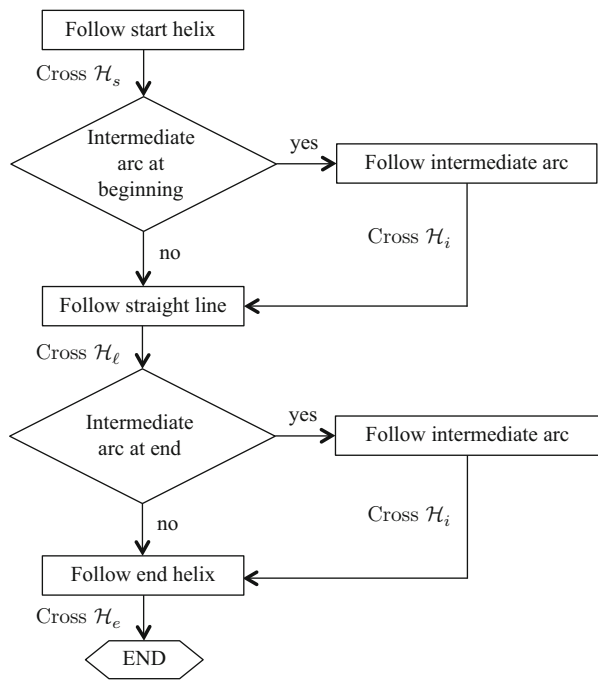


Fig. 68.12 Flow chart for the path manager for following a Dubins airplane path

simulation is the Aerosonde model described in Beard and McLain (2012). A low-level autopilot is implemented to regulate the commanded airspeed, bank angle, and flight-path angle. The windspeed in the simulation is set to zero. The simulation is implemented in Matlab/Simulink.

The simulation results for a low altitude gain maneuver are shown in Fig. 68.13, where the planned trajectory is shown in green and the actual trajectory is shown in black. The difference between the actual and planned trajectories is due to fact that the actual dynamics are much more complicated than the kinematic model given in (68.1). Simulation results for a medium altitude gain maneuver are shown in Fig. 68.14, and simulation results for a high altitude gain maneuver are shown in Fig. 68.15.

68.5 Conclusion

This chapter describes how to plan and implement Dubins airplane paths for small fixed-wing UAVs. In particular, the Dubins airplane model has been refined to be more consistent with standard aeronautics notation. A complete architecture for following Dubins airplane paths has been defined and implemented and is shown in Fig. 68.1. Dubins airplane paths consist of switching between helical and

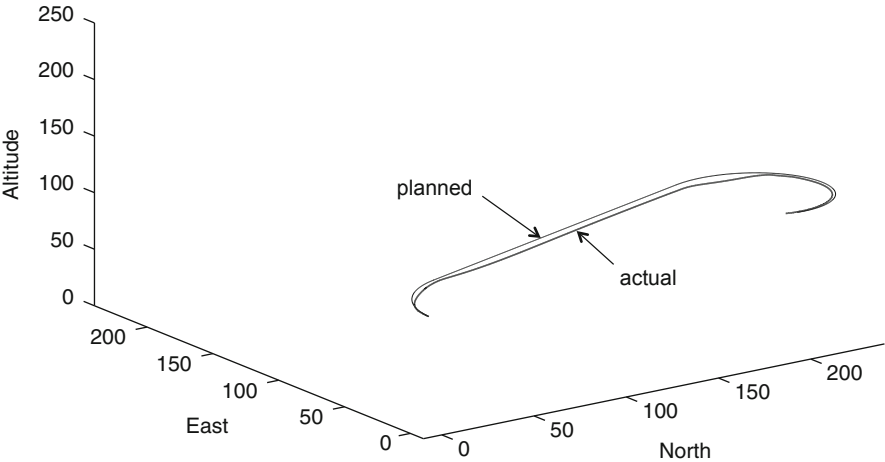


Fig. 68.13 Simulation results for Dubins path with low altitude gain

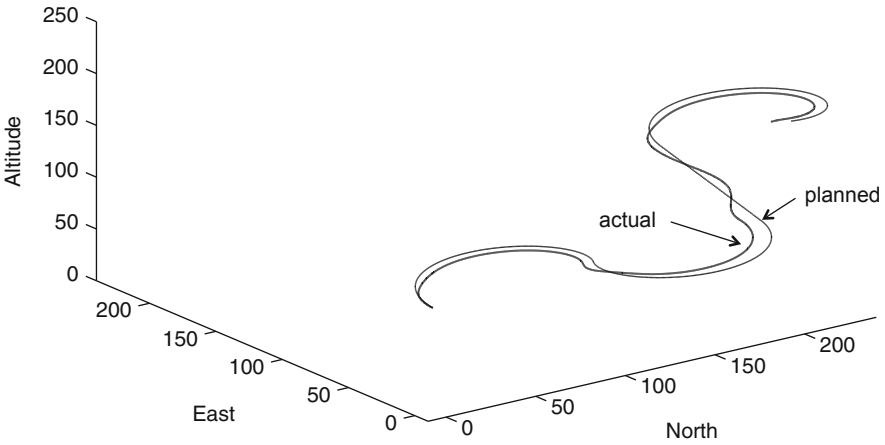


Fig. 68.14 Simulation results for Dubins path with medium altitude gain

straight-line paths. The vector-field method described in Gonçalves et al. (2010) has been used to develop guidance laws that regulate the Dubins airplane to follow the associated helical and straight-line paths. For medium and high altitude gain scenarios, there are many possible Dubins paths. This chapter suggests selecting the path that maximizes the average altitude of the aircraft during the maneuver.

There are many possible extensions that warrant future work. First, there is a need to extend these methods to windy environments, including both constant wind and heavy gusts. Second, the assumed fast inner loops on airspeed, roll angle, and flight-path angle are often violated, especially for flight-path angle. There may be

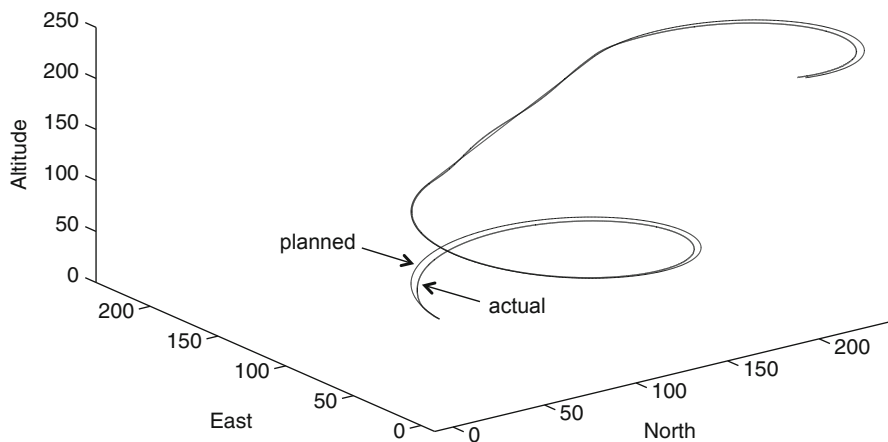


Fig. 68.15 Simulation results for Dubins path with high altitude gain

some advantage, for path optimality in particular, to factoring the time constants of the inner loops into the planning procedure. Third, this chapter assumes a decoupling between flight-path angle and airspeed. Except for highly overpowered vehicles, however, achieving a positive flight-path angle will reduce the airspeed, and achieving a negative flight-path angle will increase the airspeed. Taking these effects into account will obviously change the optimality of the paths. Finally, there are a variety of methods that have been proposed for achieving vector-field following (see Lawrence et al. 2008; Park et al. 2007; Nelson et al. 2007). The method used in this chapter is only one possibility, that, in fact, proved challenging to tune. One of the issues is that the method assumes single integrator dynamics in each direction of motion. More robust 3D vector-field following techniques that account for the nonholonomic kinematic model of the Dubins airplane need to be developed.

References

- R. Anderson, D. Milutinovic, A stochastic approach to Dubins feedback control for target tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, Sept 2011
- A. Balluchi, A. Bicchi, A. Balestrino, G. Casalino, Path tracking control for Dubins cars, in *Proceedings of the International Conference on Robotics and Automation*, Minneapolis, 1996, pp. 3123–3128
- R.W. Beard, T.W. McLain, *Small Unmanned Aircraft: Theory and Practice* (Princeton University Press, Princeton, 2012)
- S. Brunke, M.E. Campbell, Square root sigma point filtering for real-time, nonlinear estimation. *J. Guid.* **27**(2), 314–317 (2004)
- H. Chitsaz, S.M. LaValle, Time-optimal paths for a Dubins airplane, in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, Dec 2007, pp. 2379–2384

- R.V. Cowlagi, P. Tsiotras, Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning, in *Proceedings of the American Control Conference*, St. Louis, June 2009, pp. 414–419
- L.E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**, 497–516 (1957)
- J. Egbert, R.W. Beard, Low-altitude road following using strap-down cameras on miniature air vehicles. *Mechatronics* **21**(5), 831–843 (2011)
- J. Elston, E.W. Frew, Hierarchical distributed control for search and track by heterogeneous aerial robot network, in *Proceedings of the International Conference on Robotics and Automation*, Pasadena, May 2008, pp. 170–175
- J. Elston, B. Argrow, A. Houston, E. Frew, Design and validation of a system for targeted observations of tornadic supercells using unmanned aircraft, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, Taipei, Oct 2010, pp. 101–106E
- E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, R. Sengupta, Vision-based road-following using a small autonomous aircraft, in 2004 IEEE Aerospace Conference Proceedings, Big Sky, Mar 2004, vol. 5, pp. 3006–3015
- E.W. Frew, C. Dixon, J. Elston, M. Stachura, Active sensing by unmanned aircraft systems in realistic communications environments, in *Proceedings of the IFAC Workshop on Networked Robotics*, Golden, Oct 2009
- V.M. Gonçalves, L.C.A. Pimenta, C.A. Maia, B.C.O. Durtra, G.A.S. Pereira, Vector fields for robot navigation along time-varying curves in n -dimensions. *IEEE Trans. Robot.* **26**(4), 647–659 (2010)
- C. Hanson, J. Richardson, A. Girard, Path planning of a Dubins vehicle for sequential target observation with ranged sensors, in *Proceedings of the American Control Conference*, San Francisco, June 2011, pp. 1698–1703
- S. Hosak, D. Ghose, Optimal geometrical path in 3D with curvature constraint, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 113–118, Taipei, Oct 2010
- S. Karaman, E. Frazzoli, Incremental sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **30**(7), 846–894 June 2011
- D.A. Lawrence, E.W. Frew, W.J. Pisano, Lyapunov vector fields for autonomous unmanned aircraft flight control. *AIAA J. Guid. Control Dyn.* **31**, 1220–1229 (2008)
- F.L. Lewis, *Optimal Control* (Wiley, New York, 1986)
- R. Mahony, T. Hamel, J.-M. Pflimlin, Nonlinear complementary filters on special orthogonal group. *IEEE Trans. Autom. Control* **53**(5), 1203–1218 (2008)
- E.A. Misawa, J.K. Hedrick, Nonlinear observers – state-of-the-art survey. *Trans. ASME J. Dyn. Syst. Meas. Control* **111**, 344–352 (1989)
- R.C. Nelson, *Flight Stability and Automatic Control*, 2nd edn. (McGraw-Hill, Boston, 1998)
- D.R. Nelson, D.B. Barber, T.W. McLain, R.W. Beard, Vector field path following for miniature air vehicles. *IEEE Trans. Robot.* **37**(3), 519–529, (2007)
- S. Park, J. Deyst, J.P. How, Performance and Lyapunov stability of a nonlinear path-following guidance method. *AIAA J. Guid. Control Dyn.* **30**(6), 1718–1728 (2007)
- W.F. Phillips, *Mechanics of Flight*, 2nd ed. (Wiley, Hoboken, 2010)
- A. Rahmani, X.C. Ding, M. Egerstedt, Optimal motion primitives for multi-UAV convoy protection, in *Proceedings of the International Conference on Robotics and Automation*, Anchorage, May 2010, pp. 4469–4474
- S. Rathinam, Z. Kim, A. Soghikian, R. Sengupta, Vision based following of locally linear structure using an unmanned aerial vehicle, in *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, Seville, Dec 2005, pp. 6085–6090
- T. Shima, S. Rasmussen, D. Gross, Assigning micro UAVs to task tours in an urban terrain. *IEEE Trans. Control Syst. Technol.* **15**(4), 601–612 (2007)
- S.C. Spry, A.R. Girard, J.K. Hedrick, Convoy protection using multiple unmanned aerial vehicles: organization and coordination, in *Proceedings of the American Control Conference*, Portland, June 2005, pp. 3524–3529

- B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation*, 2nd ed. (Wiley, Hoboken, 2003)
- P.B. Sujit, J.M. George, R.W. Beard, Multiple UAV coalition formation, in *Proceedings of the American Control Conference*, Seattle, June 2007, pp. 2010–2015
- G. Yang, V. Kapila, Optimal path planning for unmanned air vehicles with kinematic and tactical constraints, in *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, 2002, pp. 1301–1306
- T.R. Yechout, S.L. Morris, D.E. Bossert, W.F. Hallgren, *Introduction to Aircraft Flight Mechanics*. AIAA Education Series (American Institute of Aeronautics and Astronautics, Reston, 2003)
- C. Yong, E.J. Barth, Real-time dynamic path planning for Dubins' nonholonomic robot, in *Proceedings of the IEEE Conference on Decision and Control*, San Diego, Dec 2006, pp. 2418–2423
- H. Yu, R.W. Beard, A vision-based collision avoidance technique for micro air vehicles using local-level frame mapping and path planning. *Auton. Robots* **34**(1–2), 93–109 (2013)

Health Monitoring of a Drone Formation Affected by a Corrupted Control System

69

Nicolas Léchevin, Camille Alain Rabbath, and Patrick Maupin

Contents

| | | |
|------|--|------|
| 69.1 | Introduction | 1704 |
| 69.2 | Stability Monitoring System and Recognition of Vulnerabilities | 1706 |
| 69.3 | Problem Formulation | 1706 |
| 69.4 | System Model | 1710 |
| 69.5 | Stability Condition | 1711 |
| | 69.5.1 Stability Analysis of NA | 1711 |
| | 69.5.2 Approximation for Monitoring | 1714 |
| 69.6 | Simulation Results | 1715 |
| 69.7 | Conclusion | 1717 |
| | References | 1718 |

Abstract

Safe and reliable operation of formations unmanned aerial vehicles (UAVs) necessitates developing situational awareness capacities for assessing and predicting the health status of both network and assets, particularly when evolving in adversarial environments. This chapter proposes a dynamic feature that is instrumental in achieving vulnerability assessment of a network of UAVs, whose control system is possibly affected by the diffusion of malware. The feature consists of the characterization of the transition from stability to instability with

N. Léchevin (✉)

Defence Research and Development Canada – Valcartier, Québec, QC, Canada

e-mail: Nicolas.lechevin@drdc-rddc.gc.ca

P. Maupin

Department of Engineering, Concordia University, Montreal, QC, Canada

e-mail: Patrick.maupin@drdc-rddc.gc.ca

C.A. Rabbath

Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

e-mail: Camille-Alain.Rabbath@drdc-rddc.gc.ca

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_18,

© Springer Science+Business Media Dordrecht 2015

1703

probability one. The stability of the networked UAVs can be indirectly affected by malicious attacks targeting the communication units or the control systems. The network is modeled as a discrete-time, jump, linear system whose state-space variables represent the probabilities that each node receives a malware and is infected by it. The stability analysis is obtained by means of a stochastic Lyapunov function argument and yields a sufficient condition expressed as a linear matrix inequality (LMI). This LMI involves the networked asset state-space matrices and the probability that each UAV's control system is infected. An approximation to the sufficient condition is proposed so that convergence of the system trajectories could be monitored online. The proposed detection technique is validated by simulations of a UAV formation.

69.1 Introduction

Despite local and global failure detection and recovery mechanisms (see Rabbath and L  chevin (2010) and references therein), networks are prone to malfunction or to outages. In a military context, the concept of network-centric warfare, which puts forward the use of communications network, aims to increase the capability of networked weapons and surveillance, reconnaissance, intelligence, and command-and-control systems. Yet, network-centric assets, whether it be weapons or unmanned vehicles, such as aerial robotic drones, are prone to attacks at the physical layer, data-link layer, and network layer through enemy actions including jamming, deception, destruction, and information overloading. For instance, potential risks affecting air defense systems (Fig. 69.1) include (i) uplink jamming of satellite communication regarding ground communications and ground positioning (Felstead 2007); (ii) intervehicle communications systems among cooperative UAVs, which are prone to be jammed by areal intruders, thus jeopardizing the stability of the flight (see Bhattacharya and Ba  sar (2010) and references therein for related problems); (iii) invading and hacking networked air defense from a malevolent aircraft by sending data streams along with advanced algorithms (Fulghum et al. 2007); (iv) disruption of commercial-off-the-shelf microprocessor function by sending a preprogrammed code that is part of a data stream produced by malevolent aircraft and that targets radars (Adee 2008); (v) malware with the potential to corrupt and to deceive control, guidance, and navigation systems. Malware could enter the network following a strategy similar to Stuxnet (Falliere et al. 2011), which was introduced by removable disk and which has the ability to self-replicate and to execute itself on remote computers in order to modify code on programmable logic controllers. The infection by a computer virus of ground-operating cockpits used to send command signals to Predator and Reaper drones has been detected at Creech Air Force Base in Nevada (Shachtaman 2011). It is believed that a mechanism based on infected removable hard drives introduced a computer virus to the ground-operating cockpits of Predator and Reaper drones.

Other critical vehicle-based applications that necessitate accounting for potential mishaps of the communications networks include vehicle networked control

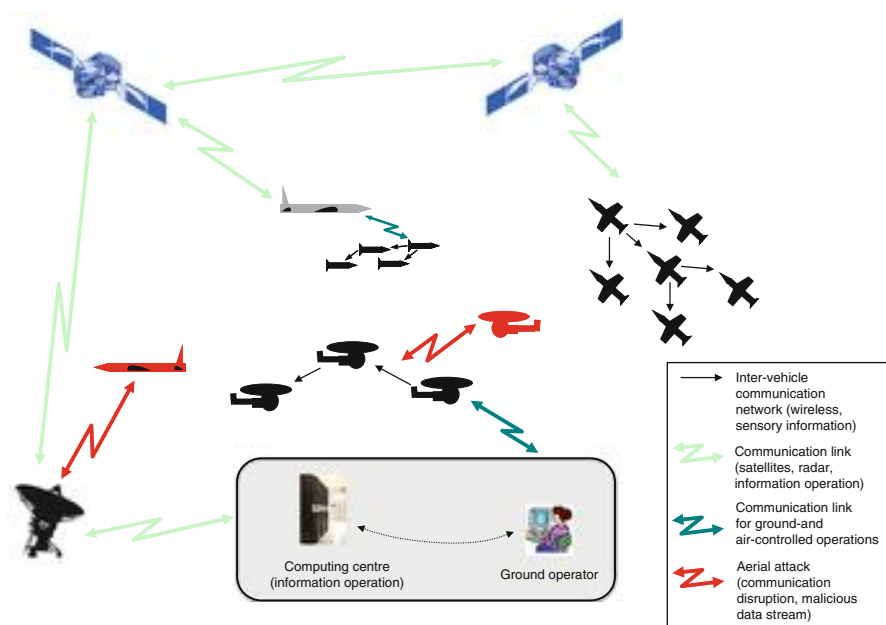


Fig. 69.1 Decentralized air system. The operation of semi-autonomous, cooperative, aircraft drones, possibly in formation flight, and of weapons is prone to vulnerabilities such as (i) intervehicle aerial communication disruption entailed by malevolent intruders and (ii) the sensitivity of satellite system's embedded protection, control, and communication devices to malicious data streams

pertaining to the automated highway (Seiler and Sengupta 2001), cooperative mobile sensor coverage (Wang and Hussein 2008; Wood et al. 2007), and wireless communications among a team of cooperative unmanned aerial vehicles (Bhattacharya and Başar 2010).

It is proposed, in this chapter, to derive stability and boundedness conditions that are expected to be implemented online for the health monitoring of a network of assets (NA). The assets consist of stabilized UAVs, each of which is equipped with a local-information-based outer-loop control law (formation controller) that ensures the stability of the formation. A healthy UAV indicates operation under nominal conditions. The network of UAVs is subject to disturbances resulting from the prevalence and propagation of infectious malicious codes (or malware) through communications network links. Malicious code includes self-replicating program such as a virus or a worm (Ford 2005). The malware affects the formation controller in such a way that the affected sensor's data are intermittently misinterpreted. In this chapter, the probability of infection is assumed to lie within the contraction region (Lohmiller and Slotine 1998) of the malware propagation's endemic state, that is, the probability that each UAV's control system is infected is always nonzero (Léchevin et al. 2011b).

Standard fault detection and identification algorithm are mostly designed to react to faulty sensors and actuators. The technique aiming to monitor the stability of networked jump dynamical systems consists in monitoring eigenvalues of an appropriate matrix of the jump system in order to detect possible stability-instability transitions. This characterization of system vulnerabilities is part of the features leveraged by the PREVU vulnerability recognition toolbox (L  chevin et al. 2011a).

The chapter is structured as follows. The context and problem formulation are presented in Sect. 69.2. The system model used for the stability analysis is given in Sect. 69.3. A stability condition, expressed as linear matrix inequalities (LMIs), and an approximate expression of this condition for online monitoring are derived in Sect. 69.4. Simulation results are presented and discussed in Sect. 69.5.

69.2 Stability Monitoring System and Recognition of Vulnerabilities

The tools proposed in this chapter for monitoring the stability of a drone formation are part of the Prediction and REcognition of VULnerabilities in very large socio-technical complex networks (PREVU) toolbox being developed at DRDC Valcartier. This toolbox relies on pattern recognition of possibly disruptive events occurring in networked dynamical systems, including infrastructures and vehicle swarms. In that framework, the vulnerability monitoring system requires a specific feature-based representation of networks, that is, feature extraction and selection is carried out to predict networks' vulnerabilities. This representation solves a classificatory problem consisting in determining the set $\theta(\omega)$ of local representations S_i of the network ω under investigation by generating a classifier ψ such that $\theta(\omega) = \{S_i \in S | \psi(\mathbf{z}_{m+1}^r, \mathbf{x}_i) = V\}$, where $\mathbf{z}_{m+1}^r = (\mathbf{x}_m^r, \mathbf{y}^r)$ is the training dataset. $\mathbf{y}^r = [y^1 \dots y^r]^T$ and \mathbf{x}_m^r are the vector of labels and the table of features representing the set of local representations (nodes, edges, clusters). V represents the vulnerability label, which is determined from a cost function expressing performance degradation, inoperability, or decrease of the quality of service of the network. The pattern recognition leverages features that are extracted from various classes (see Fig. 69.2), namely, structural features, signal and system features, and complexity features (L  chevin et al. 2011a).

Eigenvalues monitoring is proposed next to verify the stability of NA. Eigenvalues are identified as system features and are expected to complement the structural features used to analyze and predict the behavior of complex networks (L  chevin et al. 2011a; Joussetme et al. 2012).

69.3 Problem Formulation

The stability monitoring tool is applied to the chain-type formation of five autonomous quad-rotor UAVs shown in Fig. 69.3. Each vehicle is represented by a 6-degree-of-freedom (DOF), nonlinear state-space model in closed loop with a

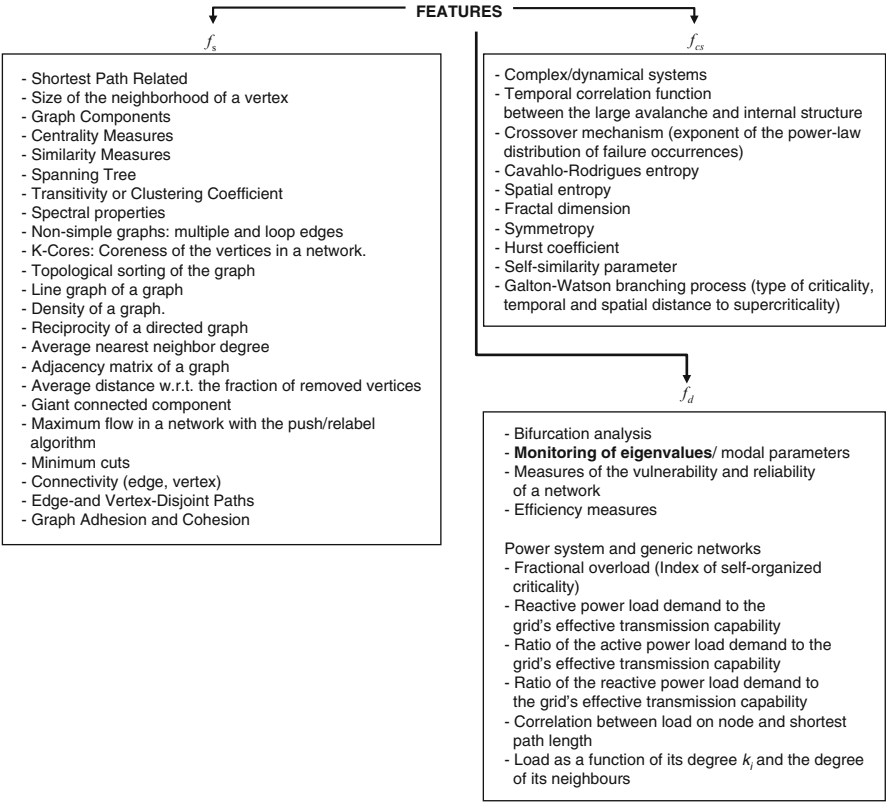


Fig. 69.2 List of features used by the classifier. This list is by no means exhaustive, for example, various measures may define the notion of centrality (Csárdi and Nepusz 2006)

control system. To account for sensor imperfection, white Gaussian measurement noise is added to the state-space vector, which comprises data from GPS and inertial measurement units (IMUs). The reader is referred to Rabbath and Léchevin (2010) for a mathematical description of the model.

The control system consists of two feedback loops (Fig. 69.4a); namely, the autopilot, which leverages the information obtained from the IMUs, and the formation controller, which is a distributed control law using local information about its neighboring vehicles (Léchevin et al. 2006). A neighbor of vehicle i is defined as any vehicle that can be sensed by i . Each of vehicles two to five of the formation illustrated in Fig. 69.3 has exactly one neighbor. Each vehicle may be affected by failures whether intended or not. In this section, the focus is on the effect of the presence of a malicious code that has been inadvertently uploaded on the electronic board of at least one drone. One can think of an infection similar to Stuxnet propagation through removable disks (Falliere et al. 2011). The coding and implementation details of such malware are not considered in this chapter.

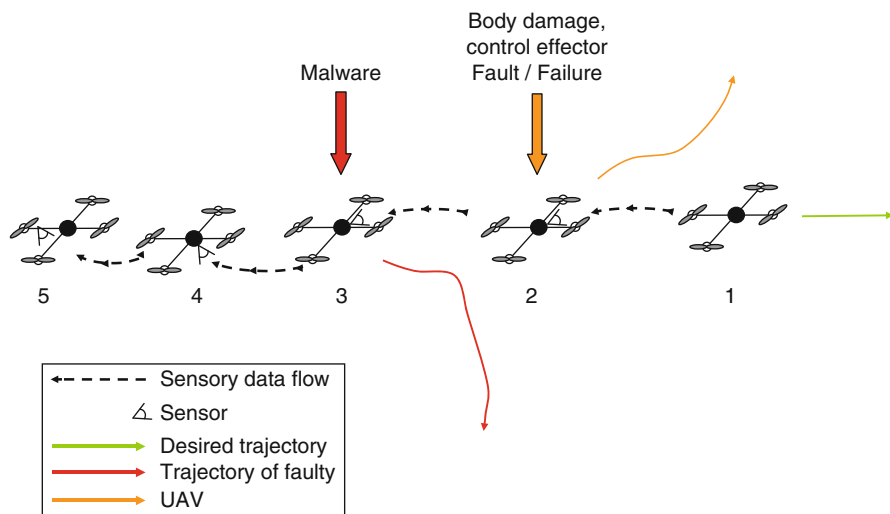


Fig. 69.3 Formation of drone aircraft affected by unexpected events. Possible trajectories are drawn following the occurrence of a malware-induced disturbance (red) and a hardware/software failure (orange). Malware-induced behavior is considered in this chapter. The interested reader may refer to Rabbath and Léchevin (2010) for body damage and control effector fault detection

Instead, a high-level abstraction of the malware effect on the control system, as shown in Fig. 69.4b, is opted for. It is thus assumed that the high-level effect can be coded by some malicious individual or organization. The reader may refer to Rabbath and Léchevin (2010) for other types of failures that might affect teams of autonomous vehicles. It should be noted that the fault detection algorithms proposed in the aforementioned references are based on the derivation of residues, which are designed to be sensitive to actuator errors. Other algorithms are developed to be sensitive to sensor faults. In the remainder of this chapter, it is shown that monitoring the spectrum of some appropriately designed matrix is useful to detect anomalies originating from the activation of a malware designed to target the control system of a team of networked autonomous vehicles.

The malware consists, as suggested in Fig. 69.4b, of a simple process that multiplies the data x_i of vehicle i by $+1$ with a probability p^* and by -1 with a probability $1-p^*$. The formation controller operating in normal condition feeds back x_i . Therefore, it is expected that using the corrupted data $-x_i$ has a destabilizing effect on the vehicle's dynamics and thus on the formation's dynamics.

Interestingly, it should be noted that this simple malicious mechanism may be interpreted as a model for intermittent losses of communications should this mechanism be applied to data x_{i-1} while assuming that x_{i-1} is obtained by means of a communications network rather than by onboard sensors. The malware could disable receiver and emitter components onboard the drones.

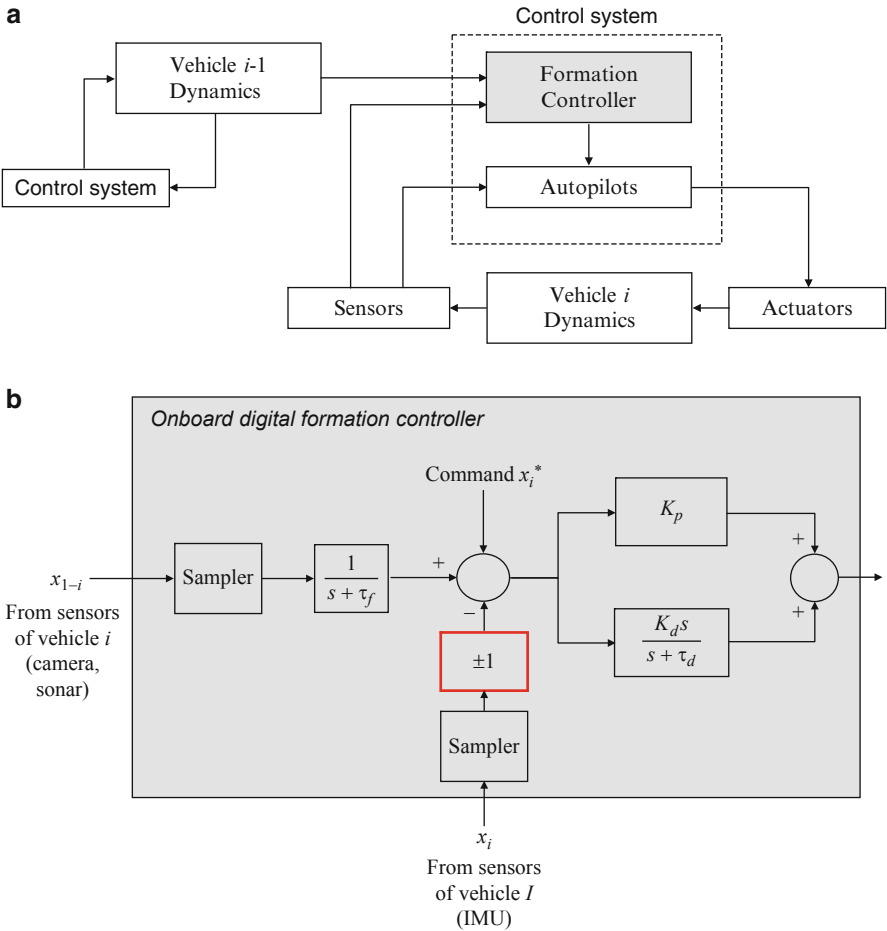


Fig. 69.4 Control system of vehicle i . **(a)** The inner-loop controller (autopilot) stabilizes the vehicle around the commanded trajectory provided by the outer-loop controller (formation controller). **(b)** Based on state vectors x_i and x_{i-1} measured by means of vehicle i 's sensors, the formation controller computes the trajectory tracked by the autopilot (Rabbath et al. 2011). The effect of the malicious code on the formation controller, implemented on vehicle i 's electronic board, is modeled by a sign change ± 1 , whose occurrence is ruled by probability p^* . An uncorrupted control system would operate with $+1$ instead of -1

Achieving system stability is a hard constraint that networked systems must satisfy. Therefore, the following stability monitoring problem is defined.

Problem Objective: To conduct online monitoring of possible stability-instability transitions of NA, the following two tasks should be carried out: (i) conditions under which the trajectories of the formation of UAVS, whose controllers are possibly corrupted by a malware of the type depicted in Fig. 69.4, converge to zero with

probability 1 (almost-sure convergence), and (ii) derive approximations of (i) that uses the information available to each UAV such as its state.

It is assumed that the ground control station is able to access at least those elements of information so that the monitoring technique can be applied. However, it is shown in the simulation section that decentralized monitoring of stability, whereby each asset monitors its health status and that of its neighbors as a means to anticipate critical situations to the NA, is possible although it does not result from a rigorous derivation.

69.4 System Model

An approximate model of each vehicle's nonlinear dynamics is used. A linearized, second-order model is selected. Parameter identification of this model is carried out from data obtained by simulating the full 6-DOF nonlinear dynamics (Rabbath and L  chevin 2010). Vehicle i 's dynamics along the x- and y-axes are decoupled in the model expressed as follows:

$$\begin{aligned} \begin{bmatrix} \dot{X}_i \\ \dot{Y}_i \end{bmatrix} &= \begin{bmatrix} A_{ix} & 0 \\ 0 & A_{iy} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} + u + W, \\ Z_i &= \begin{bmatrix} X_i \\ Y_i \end{bmatrix} + N, \\ X_i &= \begin{bmatrix} x_i - x_i^* \\ \dot{x}_i \end{bmatrix}, Y_i = \begin{bmatrix} y_i - y_i^* \\ \dot{y}_i \end{bmatrix} \end{aligned} \quad (69.1)$$

where W and N stand for the process and observation noises, respectively, which are assumed white and Gaussian. x_i^* and y_i^* are piecewise constant trajectories to be tracked by the leader. A_x and A_y are the state-space matrices of a standard second-order, linear-time, invariant system. u represents the formation controller of vehicle i . u equals zero at equilibrium.

Following (Eq. 69.1), the UAV formation equipped with the local-information-based controller (L  chevin et al. 2006) is represented by a network of assets (NA), comprised of n nodes each of which is modeled by the following discrete-time, jump, linear system

$$x_{i,k+1} = \mathbf{A}_{i,k}(\omega_{i,k})x_{i,k} + \sum_{j \in N_i^a} \mathbf{B}_{ij,k}(\omega_{i,k})x_{j,k}, \quad (69.2)$$

where N_i^a , $x_{i,k} \in \mathbb{R}^{n_i}$, and $\omega_{i,k}$ are the set of neighbors of node i , the state-space vector of asset i at t_k , and the random variable that expresses the asset i 's health status at t_k , respectively. The process and measurement noises are disregarded in (Eq. 69.2) but will be considered in simulations.

The health state of a node depends on its possible infection by the malicious code, which may affect the asset through its control system as explained in the last section.

The random matrix $\mathcal{A}_{i,k}(\omega_{i,k})$ is defined as follows. $\mathcal{A}_{i,k}(\omega_{i,k} = 0) = \bar{A}_i$ when the corresponding node i in NA is infected, which occurs with probability $p_{i,k} = P(\omega_{i,k} = 0)$, and $\mathcal{A}_{i,k}(\omega_{i,k} = 1) = A_i$ with probability $1 - p_{i,k} = P(\omega_{i,k} = 1)$, when node i is healthy. Similarly, $\mathcal{B}_{ij,k}(\omega_{i,k} = 0) = \bar{B}_{ij}$ with probability $p_{i,k}$, and $\mathcal{B}_{ij,k}(\omega_{i,k} = 1) = B_{ij}$ with probability $1 - p_{i,k}$. The interconnection matrix $\mathcal{B}_{ij,k}(\omega_{i,k})$ in (Eq. 69.2) represents the dependency of i on j . Bidirectional dependency between i and j is thus characterized by the matrix pair $(\mathcal{B}_{ij,k}(\omega_{i,k}), \mathcal{B}_{ji,k}(\omega_{j,k})) \in \{B_{ij}, \bar{B}_{ij}\} \times \{B_{ji}, \bar{B}_{ji}\}$, which is determined as a function of the state (infective or susceptible) of corresponding nodes, i on j , in NA. From (Eq. 69.2), the overall dynamics of the network of assets is given by the jump linear system

$$x_{k+1} = \mathbf{A}_k(\omega_k)x_k, \quad (69.3)$$

where $x_{k+1} = [x_{1,k}^T, \dots, x_{n,k}^T]^T$ and $\omega_k = [\omega_{1,k}, \dots, \omega_{n,k}]^T \in \{0,1\}^n$. $x_{i,k}$, and $\omega_{i,k}$ are given in (Eq. 69.2). Matrix $\mathcal{A}_k(\omega_k)$ is straightforwardly obtained from $\mathcal{A}_{i,k}(\omega_{i,k})$ and $\mathcal{B}_{ij,k}(\omega_{i,k})$ in (Eq. 69.2) for all $i \in \{1, \dots, n\}$ and $j \in N_i^a$, and is characterized by a graph Laplacian-like structure (Godsile and Royle 2001).

A nonlinear probabilistic model of malware propagation (Chakrabarti et al. 2008) is applied, in Léchevin et al. (2011b), to a communications network (CN) connected to a network of assets, where the concept of assets can be generalized to such systems as infrastructures. A contraction analysis is then carried out to derive a sufficient stability condition of CN. Here, it is assumed that the presence of the malware within NA is endemic that is the probability $p_{i,k}$ that node i is infected converges, as $k \rightarrow \infty$, to a nonzero probability $p_{i,k}^*$.

69.5 Stability Condition

69.5.1 Stability Analysis of NA

Select the following positive definite function:

$$V(x_k) = x_x^T x_k. \quad (69.4)$$

V is instrumental in the definition of the stochastic Lyapunov function candidate used for the stability analysis. Then,

$$\begin{aligned} E[V(x_{k+1})|\mathbf{F}_n] &= E[x_x^T \mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k) x_k] \\ &= x_x^T \bar{p}_k A^T A x_k + \sum_{\omega_k \in I_k \setminus \mathbf{1}_n} \prod_{i=1}^n \rho_{i,k} x_x^T \mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k) x_k, \end{aligned} \quad (69.5)$$

where \mathcal{F} stands for a sequence of nonincreasing σ -algebras measuring $\{x_i, i \leq n\}$, $A = \mathcal{A}_k(\omega_k = \mathbf{1}_n)$, $\rho_{i,k} \in \{p_{i,k}, 1 - p_{i,k}\}$, $I_{i,k} = \{0,1\}$, $I_i = I_{1,k} \times \dots \times I_{n,k}$, $\mathbf{1}_n$ is the $1 \times n$ unitary vector, and $\bar{p}_k = \prod_{i=1}^n (1 - p_{i,k})$.

Letting $\rho_{i,k} = \rho_{i,k}^* + \tilde{\rho}_{i,k}$, where $\rho_{i,k}^*$ and $\tilde{\rho}_{i,k}$ correspond to the endemic state and to deviation with respect to $\rho_{i,k}^*$, respectively, and $\bar{p}_k = \bar{p}_k^* + \tilde{p}_k$, where the definition of \bar{p}_k^* and \tilde{p}_k is similar to that of $\rho_{i,k}^*$ and $\tilde{\rho}_{i,k}$; (Eq. 69.5) can be expressed as

$$E[V(x_{k+1})|\mathbf{F}_n] = x_x^T (\bar{p}_k^* A^T A + \sum_{\omega_k \in I_k \setminus I_n} \prod_{i=1}^n \rho_{i,k}^* \mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k)) x_k + \tilde{V}_k. \quad (69.6)$$

In (Eq. 69.6), \tilde{V}_k is defined as follows:

$$\begin{aligned} \tilde{V}_k = x_x^T & \left(\sum_{\omega_k \in I_k} M_k(\rho_{i,k}^*, \tilde{\rho}_{j,k}, i \in I, j \in J) \right. \\ & \left. \times \mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k) \right) x_k, \end{aligned} \quad (69.7)$$

where $I \subset \{1, \dots, n\}$ and $J \subset \{1, \dots, n\} \setminus I$.

M_k in (Eq. 69.7) is the sum of all products expressed as $\prod_{i \in I} \rho_{i,k}^* \prod_{j \in J} \tilde{\rho}_{j,k}$, therefore $\lim_{\tilde{\rho}_k \rightarrow 0} M_k = 0$, implying that

$$\lim_{\tilde{\rho}_k \rightarrow 0} \tilde{V}_k = 0. \quad (69.8)$$

The stability analysis will thus consist in deriving a stochastic Lyapunov function candidate V_k , to be determined in the sequel. V_k is based on V and on a perturbation term, δV_k , that compensates for \tilde{V}_k in (Eq. 69.7) and that approaches zero as k goes to infinity by exploiting the fact that $p_{i,k}^*$ is attracting.

Letting

$$\begin{aligned} \bar{\lambda}(\tilde{\rho}_k) = \sup_j & \left| \lambda_j \left(\sum_{\omega_k \in I_k} M_k(\rho_{i,k}^*, \tilde{\rho}_{j,k}, i \in I, j \in J) \right. \right. \\ & \left. \left. \times \mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k) \right) \right|, \end{aligned} \quad (69.9)$$

where $\lambda_j(P)$ denotes the j th eigenvalue of matrix P ; \tilde{V}_k in (Eq. 69.7) can be bounded as

$$\tilde{V}_k \leq \bar{\lambda}(\tilde{\rho}_k) \|x_k\|^2. \quad (69.10)$$

$\|x\|$ stands for the Euclidean norm of vector x . It should be noted that $\lim_{\tilde{\rho}_k \rightarrow 0} \bar{\lambda}(\tilde{\rho}_k) = 0$ since $\lim_{\tilde{\rho}_k \rightarrow 0} M_k = 0$.

The stochastic Lyapunov function candidate is now selected as follows

$$V_k(x_k) = V(x_k) + \delta V_k, \quad (69.11)$$

where $V(x_k)$ is defined in (Eq. 69.4) and

$$\delta V_k = \bar{\lambda}(\tilde{\rho}_k) \sum_{i=k}^{+\infty} \|x_i\|^2 I(\|x_i\|^2 < \|x_k\|^2 + K), \quad (69.12)$$

for some $K > 0$. δV_k is positive and can be bounded by

$$\delta V_k \leq \sup_{i \geq k} (\bar{\lambda}(\tilde{\rho}_i)) \underbrace{\sum_{i=k}^{+\infty} (\|x_k\|^2 + K)}_{S_k}. \quad (69.13)$$

The supremum in (Eq. 69.13) exits from the definition of $\bar{\lambda}(\tilde{\rho}_k)$ in (Eq. 69.9) and from the fact that $p_{i,k}^*$ is attracting. Since $S_{k+1} - S_k = -K - \|x_k\|^2 < 0$ and S_k is positive, S_k is convergent which, in turn, implies that δV_k is convergent. The convergence of $\bar{\lambda}(\tilde{\rho}_k)$ to zero as $\tilde{\rho}_k$ approaches zero, that is, as k goes to infinity, implies that

$$\lim_{k \rightarrow +\infty} \delta V_k = 0. \quad (69.14)$$

Employing (Eq. 69.5), (Eq. 69.10), (Eq. 69.11), and the fact that

$$\delta V_{k+1} - \delta V_k = -\bar{\lambda}(\tilde{\rho}_k) \|x_k\|^2 \quad (69.15)$$

yields the difference

$$\begin{aligned} E[V_{k+1}(x_{k+1})|\mathbf{F}_n] - V_k(x_k) &= x_k^T \left(\bar{p}_k^* A^T A \right. \\ &\quad \left. + \sum_{\omega_k \in I_k \setminus 1_n} \prod_{i=1}^n \rho_{i,k}^* \mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k) - I \right) x_k + \tilde{V}_k \\ &\quad - \bar{\lambda}(\tilde{\rho}_k) \|x_k\|^2. \end{aligned} \quad (69.16)$$

Provided that there exists a symmetric, positive definite matrix L such that

$$\bar{p}_k^* A^T A + \sum_{\omega_k \in I_k \setminus 1_n} \prod_{i=1}^n \rho_{i,k}^* \mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k) + L < I, \quad (69.17)$$

(Eq. 69.16) is simplified as

$$E[V_{k+1}(x_{k+1})|\mathbf{F}_n] - V_k(x_k) \leq -\kappa \|x_k\|^2, \quad (69.18)$$

where $\kappa > 0$ is the smallest eigenvalue of L . The main result of this chapter is now stated.

Applying Theroem 4.2 in Kushner and Yin (1997, p. 81) to (Eq. 69.11), whose time difference is expressed in (Eq. 69.18), taking account of the convergence property of δV_k established in (Eq. 69.14), and assuming that LMI in (Eq. 69.17) is satisfied for all $k \in \mathbb{N}$, then the state x_k of NA in (Eq. 69.3) converges to zero with probability 1.

The analysis can be extended to a class of jump systems, which are excited by an additive disturbance, yielding an almost-sure boundedness of the trajectories (L  chevin et al. 2011b).

69.5.2 Approximation for Monitoring

The way online monitoring is carried out depends on the type of information that is available to the control-and-command center.

Case 1: When the identity of infected hosts is known, empirical frequency f_k^* is substituted for p_k^* in (Eq. 69.17), recalling that $\rho_{i,k}^* \in \{p_{i,k}^*, 1 - p_{i,k}^*\}$. The stability condition is monitored by direct inspection of the eigenvalues of the left-hand side of LMI in (Eq. 69.5).

Case 2: Should the estimate of the limit set p_k^* be unavailable, the left-hand side of (Eq. 69.17) can be approximated by leveraging the state x_k , which is assumed known either from measurement or from state estimation. Noting that

$$E[\mathbf{A}_k^T(\omega_k)\mathbf{A}_k(\omega_k)] = \bar{p}_k^* A^T A + \sum_{\omega_k \in I_k \setminus I_n} \prod_{i=1}^n \rho_{i,k}^* \mathbf{A}_k^T(\omega_k)\mathbf{A}_k(\omega_k), \quad (69.19)$$

and selecting $N \in \mathbb{N}$ sufficiently large, the following approximation is used:

$$E[\mathbf{A}_k^T(\omega_k)\mathbf{A}_k(\omega_k)] \cong \frac{1}{N} \sum_{j=k-N}^N A_{(j)}^T A_{(j)}, \quad (69.20)$$

where $A_{(j)}$ denotes the realization of $\mathbf{A}_j(\omega_j)$ at t_j . A least-square estimate of the average of $A_{(j)}^T A_{(j)}$ in (20) can be computed as follows. Concatenating the following expression: $x_{j+1}^T x_{j+1} = x_j^T A_{(j)}^T A_{(j)} x_j$ for all $j \in [k, k + N - 1]$ where $N \gg \text{size}(x_k)$ yields

$$X_{k,N} = Y_{k,N-1} \text{vec}_2(E_{k,N}) + \varepsilon_k, \quad (69.21)$$

where

$$\begin{aligned}
 X_{k,N-1}^T &= [x_k^T x_k, \dots, x_{k+N-1}^T x_{k+N-1}], \\
 Y_{k,N-1}^T &= [\text{vec}_1(x_k x_k^T) \dots \text{vec}_1(x_{k+N-1} x_{k+N-1}^T)] \\
 E_{k,N} &= \frac{1}{N} \sum_{j=k}^{k+N-1} A_{(j)}^T A_{(j)}, \\
 \tilde{E}_j &= A_{(j)}^T A_{(j)} - E_{k,N} \\
 \varepsilon_k^T &= [\varepsilon_{k,1}, \dots, \varepsilon_{k,j}, \dots, \varepsilon_{k,N}], \varepsilon_{k,j} = x_j^T \tilde{E}_j x_j.
 \end{aligned} \tag{69.22}$$

$\text{vec}_1(E)$ denotes the column vector formed by concatenating the rows of the upper triangular block of matrix E . $\text{vec}_2(E) = \text{vec}_1(F)$, where entries f_{ii} and e_{ii} of F and E , respectively, are such that $f_{ii} = e_{ii}$ for all i and $f_{ij} = 2e_{ij}$ for all $i \neq j$. N is such that $E(\mathcal{A}^T(\omega_i) \mathcal{A}(\omega_i)) \cong E_{k,N}$. The least-square estimate of $E_{k,N}$, which is a symmetric matrix, is given by

$$\text{vec}_2(\hat{E}_{k,N}) = (Y_{k,N-1}^T Y_{k,N-1})^{-1} Y_{k,N-1}^T X_{k,N}, \tag{69.23}$$

The approximate in (Eq. 69.23) can thus be expressed as follows:

$$E[\mathbf{A}_k^T(\omega_k) \mathbf{A}_k(\omega_k)] \cong \hat{E}_{k,N}^T. \tag{69.24}$$

The online monitoring thus consists in verifying that

$$\sup_i \lambda_i(\hat{E}_{k,N}) < 1, \tag{69.25}$$

where $\hat{E}_{k,N}$ is obtained from $\text{vec}(\hat{E}_{k,N})$ in (Eq. 69.23).

69.6 Simulation Results

A formation of quadrotor drones, whose speed is about 1 m/s, is required to track a straight line trajectory of 100 m. At time instant 75 s, the malicious mechanism is triggered with probability p^* . Intuitively, the chance of facing instabilities increases as p^* decreases. This idea is confirmed in Fig. 69.5 where it is shown that, when $p^* = 0.5$, the prescribed straight line trajectory is unstable for followers two to five. The formation remains stable for larger values of p^* , while vehicles two to five follow the leader with a nonzero steady-state error.

The simulation time step is 10 ms. $N = 3,000$, that is, a time window of 30 s is adopted. The largest eigenvalue of $\hat{E}_{k,N}$, given in (Eq. 69.23) and computed from

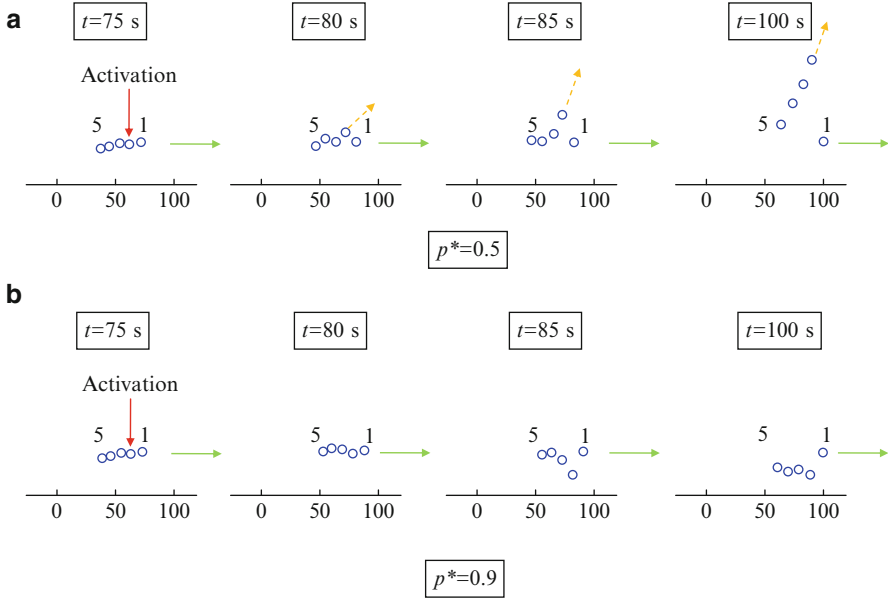


Fig. 69.5 Stability-instability transition as a function of p^* . The malicious code is activated at $t=75$ s. When $p^* \leq 0.5$, the formation's dynamics is unstable. It remains stable, although with a nonzero steady-state error, otherwise. *Green arrows* denote the desired trajectory. *Orange arrows* denote the speed vector of vehicle 2, which is affected by the malware over (75 s, 100 s)

the measurement vector $Z=[Z_1, Z_2, Z_3, Z_4, Z_5]^T$, is shown in Fig. 69.6a. It should be noted that, owing to the presence of noisy signals, a hypothesis testing is implemented to verify whether or not the maximum eigenvalue of matrix $\hat{E}_{k,N}$ is greater than one. The cumulative sum (CUSUM) algorithm is implemented (Basseville and Nikiforov 1993). The decision function is shown in Fig. 69.6b.

Setting the threshold h to 150 gives a detection time equal to 4 s, that is, the anomaly is detected at $t = 79$ s. The process by which the threshold is selected depends on the level of noise and aims to minimize the detection time while maintaining the false alarm rate below some prescribed limit.

Interestingly, the largest eigenvalue of $\hat{E}_{k,N,i}$, which is computed for each vehicle i from Z_i and not for the whole formation, is also informative. Indeed, this decoupling property could be exploited to implement a distributed version of (Eq. 69.23), which requires the knowledge of the whole formation. Applying a first-order, low-pass filter with a time constant of 10 s to the noisy signals entering the expression in (Eq. 69.23), which is now applied to each follower, yields the eigenvalue time variations in Fig. 69.6c–f, each of which corresponds to a vehicle $j \in \{2, 3, 4, 5\}$ whose unstable dynamics is detected locally whenever $\max(\hat{E}_{k,N,i})$ exceeds one. Since the malware is activated in vehicle 2, the devious motion of this vehicle is detected first. The dynamic couplings that result from the implementation of the distributed formation controllers propagate the unstable behavior of vehicle 2 to the remaining followers in the formation.

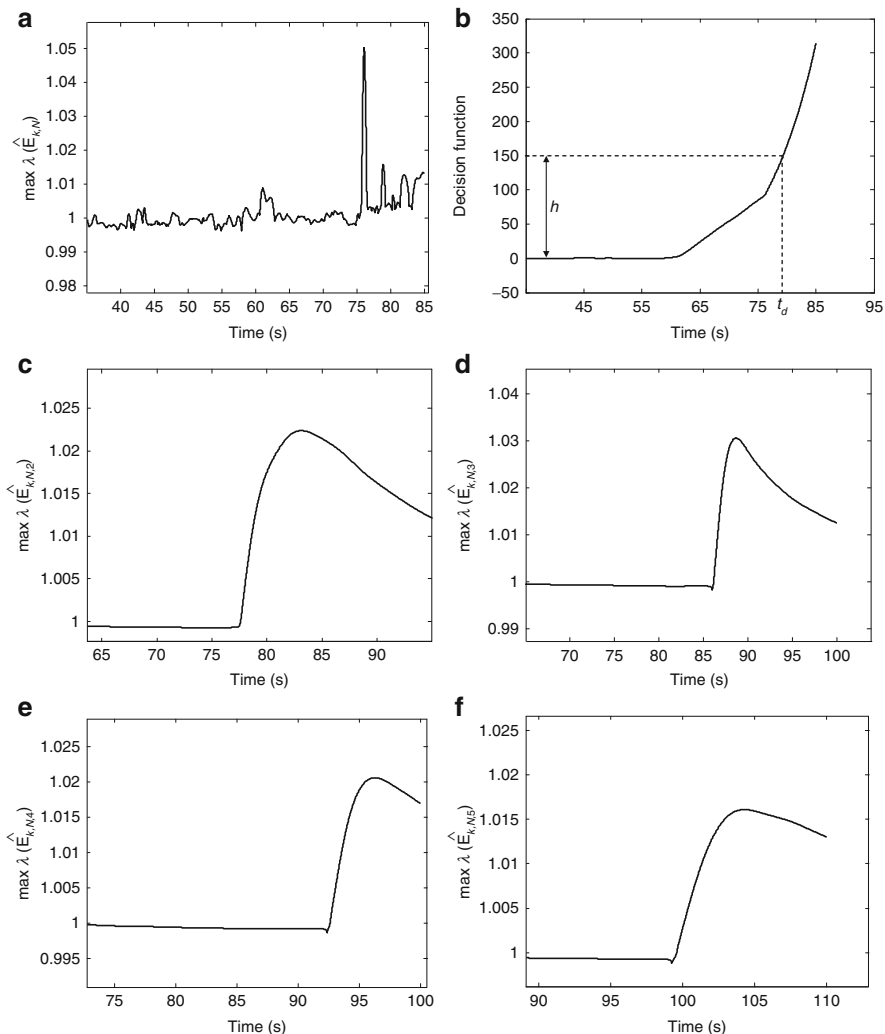


Fig. 69.6 Largest eigenvalues of $\hat{E}_{k,N}$. (a) Largest eigenvalue of the matrix $\hat{E}_{k,N}$ corresponding to the formation of five vehicles; (b) decision function of the CUSUM algorithm compared with a threshold h ; (c), (d), (e), and (f) show the largest eigenvalue of matrices $\hat{E}_{k,N,2}$, $\hat{E}_{k,N,3}$, $\hat{E}_{k,N,4}$, and $\hat{E}_{k,N,5}$, respectively

69.7 Conclusion

A great deal of effort has been devoted to the detection, identification, and mitigation of hardware-software faults and failures on critical systems such as aircraft. It is also highly desirable to analyze and possibly predict the vulnerabilities of such asset when integrated in networks, which are particularly prone to loss of efficiency and

even to potential disruption caused by the propagation of malicious code. Indeed, the activation of a malware, once it has reached some of its targets, may jeopardize the proper operation of the networked assets. Consequences range from the occurrence of stable, off-nominal operating condition to the transition to unstable operating condition, possibly leading to catastrophic events, such as mission failure and avalanche, and placing civilian and military personnel at risk in case of a network of mobile platforms.

Sufficient stability conditions are derived in this chapter, suggesting an approach for the monitoring of a network of UAVs, provided the entire state of the model is available. The model is expressed as a jump linear system. The proposed approach relies on the monitoring of the eigenvalues of a matrix expressed as a function of the measured signals.

The stability monitoring reports satisfactory results when applied to a formation flight of drone aircraft, which are stabilized by means of distributed controllers. Feedbacks are based on local information such as the relative position between two neighboring vehicles. A simple malicious mechanism corrupts the digital realization of the controller in such a way that the formation becomes unstable provided the probability of malware occurrence is less than a threshold.

Extensions of the work presented in this memorandum include (i) real-time implementation of a decentralized estimator whereby the stability monitoring is carried out with local signals, (ii) the prediction of stability-instability transitions rather than just monitoring it, and (iii) the study of best possible actions or drone adaptation techniques following the monitoring process.

References

- S. Ade  , The hunt for the kill switch, *IEEE Spectrum* **45**(5), 35–39 (2008)
- M. Basseville, I.V. Nikiforov, *Detection of Abrupt Changes: Theory and Applications* (Prentice-Hall, Englewood Cliffs, 1993)
- S. Bhattacharya, S. Ba  ar, Game-theoretic analysis of an aerial jamming attack on a UAV communication network, in *Proceedings of the 2010 American Control Conference*, Baltimore, MD, 2010
- D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, C. Faloutsos, Epidemic thresholds in real networks, *ACM Trans. Inf. Syst. Secur.* **10**(4), Article 13 (2008)
- G. Cs  r  di, T. Nepusz, The igraph software package for complex network research, *InterJournal Complex Syst.* **1695** (2006)
- N. Falliere, L.O. Murchu, E. Chien, W32.Stuxnet Dossier, Version 1.4, Symantec Corporation, February (2011)
- E.B. Felstead, Overview of Statcom Jamming and Detection, TTCP Workshop on the Robustness and Vulnerability of Network Centric Warfare, DRDC Ottawa (2007)
- R. Ford, Malcode mysteries revealed. *IEEE Secur. Priv.* **3**(3), 72–75 (2005)
- D. Fulghum, R. Wall, A. Butler, Cyber-combat’s first shot. *Aviation Week and Space Technology* **167**(21) (2007)
- C. Godsil, G. Royle, *Algebraic Graph Theory* (Springer, New York, 2001)
- A.-L. Joussetme, K. Huggins, N. L  chevin, P. Maupin, D. Larkin, Vulnerability-aware architecture for a tactical, mobile cloud, in *3rd Workshop on Complex Networks (CompleNet)*, Melbourne, FL (2012)

- H.J. Kushner, G.G. Yin, *Stochastic Approximation Algorithms and Applications* (Springer, New York, 1997)
- N. Léchevin, C.A. Rabbath, P. Sicard, Trajectory tracking of leader-follower formations characterized by constant line-of-sight angles. *Automatica* **42**, 2131–2141 (2006)
- N. Léchevin, A.-L. Joussemme, P. Maupin, Pattern recognition framework for the prediction of network vulnerabilities, in *IEEE Network Science Workshop*, West Point, NY (2011a)
- N. Léchevin, C.A. Rabbath, P. Maupin, *Toward a Stability and Risk Monitoring System of an Asset-Communications Network Exposed to Malicious Attacks*, 2011 American Control Conference, San Francisco (2011b)
- W. Lohmiller, J.-J.E. Slotine, On contraction analysis for nonlinear systems. *Automatica* **34**(6) 683–696 (1998)
- C.A. Rabbath, N. Léchevin, *Safety and Reliability in Cooperating Unmanned Aerial Systems* (World Scientific, Hackensack, New-Jersey 2010)
- C.A. Rabbath, N. Léchevin, J. Apkarian, *Experiments with a Passivity-based Formation Control System for Teams of Small Robotic Drones*, AIAA Guidance, Navigation and Control Conference and Exhibit, Portland, Oregon (2011)
- P. Seiler, R. Sengupta, Analysis of communication losses in vehicle control problems, in *Proceedings of the American Control Conference*, Arlington, VA (2001)
- N. Shachtaman, Exclusive: computer hits U.S. drone fleet, *Wired* (2011), <http://www.wired.com/dangerroom/2011/10/virus-hits-drone-fleet/>. Accessed 12 Oct 2011
- Y. Wang, I.I. Hussein, Awareness coverage control over large-scale domains with intermittent communications, in *Proceedings of the American Control Conference*, Seattle, WA (2008)
- A.D. Wood, J.A. Stankovic, G. Zhou, Deejam: defeating energy efficient jamming in ieee 802.15.4 based wireless networks, in *Proceedings of the 4th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 07)* San Diego, CA (2007), pp. 60–69

Section XIII

UAV Autonomy

***Christos Cassandras, David Castañón
and LoraWeiss***

Kimon P. Valavanis and George J. Vachtsevanos

UAV Autonomy addresses perhaps the most challenging and important topic that is central to unmanned aircraft performance, autonomy. Autonomous flight is a major target goal of all technologists. The ability of a UAV to take off, execute a mission, and return to its base without significant human intervention (this is the human-on-the-loop concept rather than the human-in-the-loop current requirement) promises to enhance UAV deployment in many application domains. Contributions in this part of the handbook address levels of autonomy and those specific technical challenges that must be overcome if one aims at approaching eventually “full UAV autonomy.” Hardware and software requirements for increased autonomy are discussed with emphasis on algorithms that will bestow to the UAV features of autonomy.

► **Integrated Hardware/Software Architectures to Enable UAVs for Autonomous Flight** by Pippin describes hardware and software architecture components that enable autonomous flight operation, also presenting some recommended architectural approaches and best practices. This is deemed essential as UAVs become increasingly more autonomous and ubiquitous and system hardware and software designers need to consider system requirements that contribute to increased levels of autonomy. Autonomous UAV architectures need to be modular and open and provide visibility for operators and testers, also embracing open and extensible architectures that can be configured and deployed for robust and safe autonomous operation. Future UAVs will operate with varying levels of human interaction

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

and will require the ability to process sensor data onboard and to operate using high-level, autonomous behaviors. The chapter concludes with a case study of the hardware and software architecture for an autonomous UAV research platform.

► **Distributed Optimization of Autonomous UAVs with Event-Driven Communication** by Zhong and Cassandras considers problems where multiple autonomous UAVs cooperate to control their individual state so as to optimize a common objective while communicating with each other to exchange state information. Since communication costs can be significant, especially when the UAVs operate with limited energy, conditions are sought under which communication of state information among nodes can be restricted while still ensuring that the optimization process converges. An asynchronous, event-driven scheme is described which limits communication to instants when some state estimation error function at a node exceeds a threshold. This scheme guarantees convergence to an optimal even in the presence of communication delays as long as they are bounded. This approach is applied to a UAV network coverage control problem where the objective is to maximize the probability of detecting events occurring in a given region, and it is shown that it may significantly reduce communication costs, hence also prolonging the system's lifetime, without any performance degradation.

► **UAV Guidance Algorithms via Partially Observable Markov Decision Processes** by Ragi and Chong presents a path-planning algorithm to guide UAVs to track multiple ground targets based on the theory of *partially observable Markov decision processes* (POMDPs). This method shows how to exploit the generality and flexibility of the POMDP framework by incorporating a variety of features of interest naturally into the framework, which is accomplished by plugging in the appropriate models. Specifically, this study shows how to incorporate the following features by appropriately formulating the POMDP action space, state transition law, and objective function: (1) control UAVs with both forward acceleration and bank angle subject to constraints, (2) account for the effect of wind disturbance on UAVs, (3) avoid collisions between UAVs and obstacles and among UAVs, (4) track targets while evading threats, (5) track evasive targets, and (6) mitigate track swaps.

Collectively, chapters in this section pave the way to addressing the road map to unmanned aircraft full autonomy.

Integrated Hardware/Software Architectures to Enable UAVs for Autonomous Flight

71

Charles Pippin

Contents

| | | |
|--------|---|------|
| 71.1 | Introduction | 1726 |
| 71.2 | Architecture | 1727 |
| 71.2.1 | System Architecture | 1727 |
| 71.2.2 | UAV Software Architecture | 1728 |
| 71.2.3 | Test and Evaluation Considerations | 1730 |
| 71.2.4 | On-Board Versus Off-Board Computation | 1731 |
| 71.2.5 | Interoperability and Standards | 1732 |
| 71.3 | Behavior-Based Autonomy | 1732 |
| 71.3.1 | Mission-Level Autonomy | 1732 |
| 71.3.2 | Behavior-Based Architecture | 1734 |
| 71.4 | Case Study: A UAV Cooperative Research Platform | 1736 |
| 71.4.1 | Introduction | 1736 |
| 71.4.2 | UAV-UGV Teaming | 1738 |
| 71.4.3 | UAV Platform | 1738 |
| 71.4.4 | UGV Platform | 1743 |
| 71.4.5 | JAUS Standard | 1744 |
| 71.4.6 | Ground Station | 1744 |
| 71.4.7 | Cooperative Target Detection and Surveillance | 1745 |
| 71.4.8 | Case Study Summary | 1746 |
| 71.5 | Conclusion | 1746 |
| | References | 1746 |

Abstract

As unmanned aerial vehicles become increasingly more autonomous and ubiquitous, system hardware and software designers will need to consider the additional system requirements that come with increased levels of autonomy. From system design stages through testing and deployment, autonomous UAV architectures

C. Pippin

Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, GA, USA

e-mail: Charles.Pippin@gtri.gatech.edu; <http://UnmannedSystems.gtri.gatech.edu>.

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_58,

© Springer Science+Business Media Dordrecht 2015

1725

will need to be modular, open, and provide visibility for operators and testers. These future UAVs will need to embrace open and extensible architectures that can be configured and deployed for robust and safe autonomous operation. In addition, the systems will operate with varying levels of human interaction and will require the ability to process sensor data onboard and to operate using high-level, autonomous behaviors. This chapter describes hardware and software architecture components that enable autonomous flight operation. It also presents some recommended architectural approaches and best practices. Finally, a case study is presented of the hardware and software architecture for an autonomous UAV research platform.

71.1 Introduction

Imagine a scenario in the not too distant future, in which there is a natural disaster and first responders from several different organizations arrive on the scene. They each launch their own unmanned aerial systems to perform search missions over the disaster area. Rather than being tele-operated, each UAV launched is able to operate autonomously and accept mission requests from the operators, through their ground control stations. This frees the first responders to focus more of their energies on the jobs that they do best. The UAVs from the various groups have different capabilities, sensors, and ownership. Yet, as the UAVs are assigned new search tasks by their operators, they are able to use standard protocols to coordinate with the other UAVs in the area to perform their missions more quickly, efficiently, and effectively.

Now, imagine another future scenario, in which UAVs are able to perform surveillance missions with fewer numbers of operators than are required today. Rather than tele-operating the UAV or entering detailed waypoint by waypoint routes, the mission planners input higher-level missions, such as to survey a specific area, along with operational constraints, such as no-fly zones. The operators rely on the autonomous behaviors of the UAV for the low-level flight planning and control. In addition, the UAV has the ability to perform processing of the sensor data onboard and therefore relieves the need for multiple ground-based data analysts and high-bandwidth communications.

Today, there are thousands of fielded UAVs in use by the United States alone. They come from various manufacturers and have a wide range of capabilities and uses. Future unmanned systems will need increased levels of autonomy, be able to perform processing onboard, utilize open standards for interoperability, and be extensible and allow for introspection. While these scenarios are very different from the current operational standards for UAVs, much of the technology exists today to support them. Consider just some of the benefits that could be available with these improved levels of autonomy: Fewer operators would be required to operate a UAV, making them cheaper and easier to deploy. This also frees the operators to focus on high-level tasks, rather than low-level mundane jobs.

1. With increased autonomy and planning capabilities, UAVs will be able to respond faster to meet operational needs, rather than requiring teams of analysts.

2. With more of the sensor data being processed onboard, fewer analysts will be required to support each UAV. Rather, the system would point out the areas of interest to the analysts, allowing them to focus on the most important aspects of their job and reducing the communication requirements.
3. With increased autonomy, UAVs would be less susceptible to operator error and communication link failures and therefore safer to operate.

However, to support these scenarios, fundamental components of the UAV software and hardware will be required:

1. The onboard mission level autonomy will need to support high-level, behavior-based planning rather than just low-level waypoint or tele-operated control.
2. The autonomous UAVS will need to perform more processing of sensor data onboard the aircraft and have the capability to notify operators and other processes when necessary.
3. The system architecture should be extensible and loosely coupled, so that additional sensors, components, and behaviors can be easily introduced.

Both of these scenarios will be possible in the near future. However, both will require additional autonomy, interoperability, and extensibility in the UAV system architecture. This chapter will discuss design considerations that will support these capabilities and present a case study of a UAV research platform.

71.2 Architecture

71.2.1 System Architecture

The recommended architecture to support autonomous UAV operations leverages the *frontseat-backseat* driver paradigm that is used in many autonomous robotic systems (Benjamin et al. 2009). In essence, the *frontseat* driver is considered to be the flight control system, which includes the autopilot and avionics and related controller hardware related to direct flight control. The *backseat* driver is represented by the mission computer, which is more deliberative and concerned with longer-range planning. The central idea of this paradigm is to decouple the low-level control from the higher-level autonomous operations. A high-level view of this paradigm is shown in Fig. 71.1.

In this architecture, the flight control system can operate standalone, through tele-operation or predefined waypoint routes, or be driven by the mission processor. In principle, the flight control system can operate as a “black box” and it does not matter where the commands originate, while the mission processor also operates as a separate entity and has little knowledge of the internal details of the flight control system.

The flight control system (FCS) typically includes an autopilot, GPS system, gyros, IMS, and a dedicated link to a ground station computer which can be used to send waypoint and control commands to the autopilot. The FCS also directly generates the control signals for aircrafts control surfaces and throttle. Note that the FCS should be able to operate independently and isolated from the mission

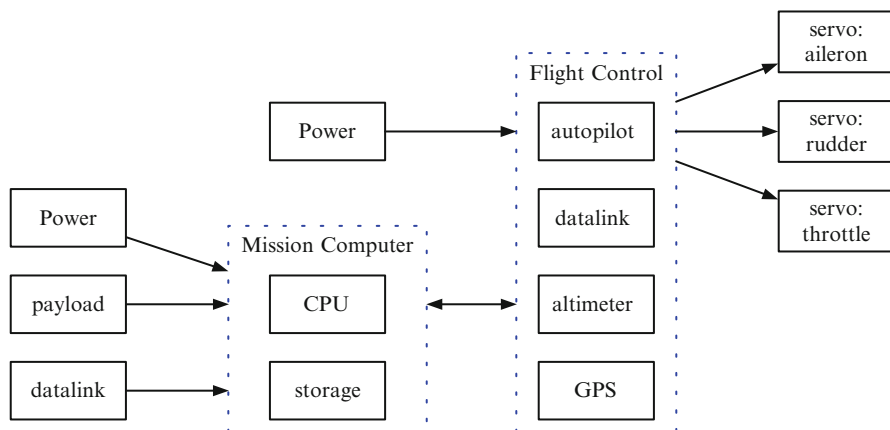


Fig. 71.1 UAV system architecture. The recommended UAV system architecture uses the *frontseat-backseat* paradigm to logically and physically separate low-level control from high-level autonomy

computer by having one or more dedicated datalinks and power sources. This provides for a level of fault tolerance in the system, as well as enabling it to be easily extensible.

For the mission processor, the design goals for extensibility dictate the use of readily available processors and components that can be easily extensible to accommodate future processing needs, autonomous behaviors, and sensor payloads. Furthermore, the processor should have multiple input/output options and relatively low power requirements and be lightweight. An example of this type of processor is the PC-104 mini computer form factor, which can be stacked in parallel to increase processing power as mission requirements dictate.

Note that the mission processor should have its own power source and datalinks for communications to the ground. In addition, it will be connected to the sensor payloads directly to allow the behaviors to incorporate the perception data. The perception data could be processed directly on the mission computer or processed on a separate payload computer, with the relevant output being passed to the mission computer. With this design, future sensor payloads can be easily added to the system and consumed by the autonomous behaviors.

71.2.2 UAV Software Architecture

The purpose of the mission processor is to run high-level mission planning and behavior processes and to exchange messages between processes, other vehicles, and the ground station. Furthermore, a key design goal of the system should be that it is flexible in regard to the architecture and potential expansion. In the future, additional autonomous behaviors, sensors, and communication capabilities may be added.

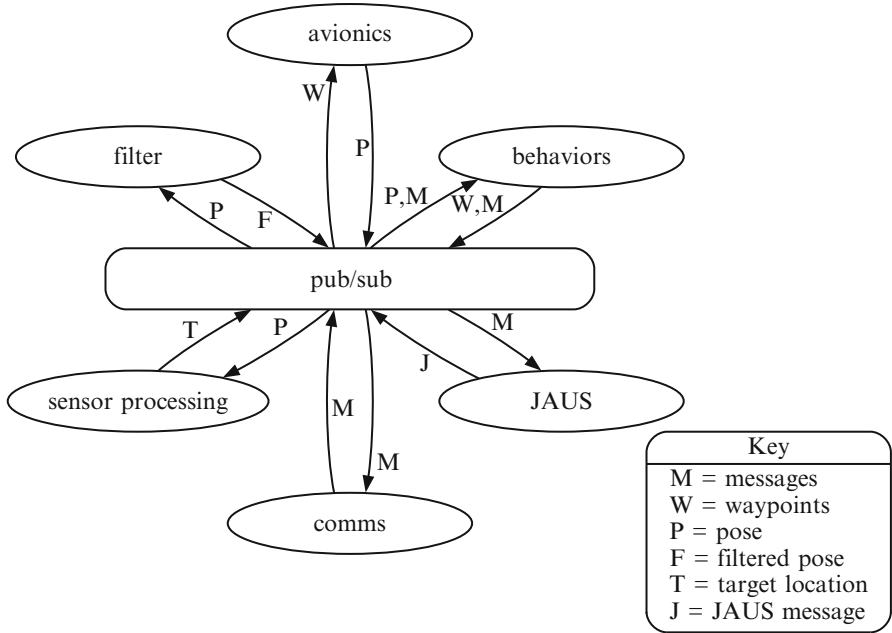


Fig. 71.2 UAV logical software architecture. The recommended UAV software architecture uses a publish/subscribe mechanism for passing messages between processes and is easily extensible

This functionality may even run across multiple processors on separate payload computers. Therefore, it is important that the software components be loosely coupled and communicate using open and flexible architectures.

One approach to providing for an extensible and open architecture is through the use of message-oriented middleware, in which separate processes send messages to each other and are loosely coupled. A typical software pattern to enable this type of flexibility is the publish-subscribe pattern. In this pattern, each process can define messages that it publishes and can subscribe to other messages that will be sent by other processes. Typically, these processes can be written in different languages and even on different computers (as long as they can communicate over a network). The messages are posted to and pulled from a logical data bus. Depending on the actual publish-subscribe implementation, the data bus could be a separate process or may be implemented in a distributed fashion. An implementation of this pattern on a UAV research platform is presented in Tisdale et al. (2006). A similar diagram of the pattern is reproduced here, in Fig. 71.2.

Consider the example of a loosely coupled UAV architecture using a publish-subscribe mechanism as shown in Fig. 71.2. Here, several processes encapsulate the details of interacting with various sensors, the autopilot system, the datalinks, and the autonomous behaviors. Each process can publish messages or subscribe to known messages from others. The design allows for new processes to be added

in the future without changing the existing system. Several sample processes and message types are described below:

Avionics The avionics process interacts with the flight control computer and receives messages from the autopilot and avionics components related to the aircraft's status. The process could publish *Pose*-related messages that include information about the aircraft's position and orientation. Similarly, the process might subscribe to *waypoint* or other command messages from the autonomous behaviors. These messages would be checked for accuracy and forwarded on to the flight control computer.

Sensor processing Multiple sensor processes can operate to process perception data from cameras, radar, or other sensors, onboard the aircraft. These processes could send specialized messages related to their sensors that could be consumed by others. For example, a process might send a message indicating that a target was detected at a given location or the level of a pollutant in the atmosphere. These processes might also subscribe to information from other sensors (to perform sensor fusion) or to messages from the avionics process to include *Pose* information.

Behaviors The high-level autonomy processes operate as sets of behaviors that perform high-level autonomous operations and planning. As an example, a search behavior might generate a pattern of *Waypoint* messages that would be consumed by the avionics process and sent to the flight control computer. The behaviors could subscribe to messages from the sensor processes and change behaviors based on the messages received. The behaviors also publish mission commands and status messages for the communications and other processes to consume.

Comms The communications process encapsulates access to the mission datalink and serves as a gatekeeper between the physical data-link and the onboard processes.

Filter The filter processes could subscribe to any of the messages such as *Pose* or sensor data and perform additional filtering or data fusion and publish the results.

JAUS JAUS or other standards-related processes could be used to transform messages between standard formats to support interoperability.

This approach supports extensibility and encapsulates lower-level components from other processes. For instance, in the future, additional sensors could be added to the system that subscribe to the *Pose* messages and publish *Target Found* messages. In another example, a new standards process could be added to support the translation from one standard to another to further support interoperability.

71.2.3 Test and Evaluation Considerations

Perhaps the most important consideration related to autonomous UAV flight is safety, and any discussion of software architectures for enabling autonomous operation should include this consideration. Autonomous UAVs will need to pass through rigorous test and evaluation procedures before they can be certified and trusted.

To facilitate this, UAV designers should consider how to incorporate safety and test harnesses at each step of the process.

71.2.3.1 Introspection

Rather than operating as a “black box,” autonomous behaviors need to provide an ability for designers and testers to evaluate what the system perceives as the state of the world. For instance, perception components need the ability for operators to visualize data as well as labeled classifications. Autonomous behaviors should regularly report and log their own state and steps toward a particular decision process. The level of logging and introspection should be configurable to allow for additional “debugging” information to be available when needed.

71.2.3.2 Health Monitoring

As part of the overall system reliability and safety, a separate health monitor process should also run on the mission computer. The purpose of this process is to monitor other processes and ensure that they are running within expected design parameters (i.e., they have not crashed, consumed too much memory or CPU, and are responding to messages.) Part of certifying each new process or autonomous behavior is to integrate it with the health monitor and validate that it handles failure gracefully. In addition, the ground station component should include separate, parallel health monitor capabilities that monitor the state of each autonomous behavior, the communications links, batteries, CPU usage, and disk space.

71.2.3.3 Software-in-the-Loop Simulations

The use of the autopilot software-in-the-loop (SIL) simulation environment allows for real-time feedback from the autopilot, as well as control using low-level bank angle commands in a simulation environment. Ideally, each component in the system should have the capability to run as part of a simulation. This would allow for the testers to exercise autonomous behaviors under different conditions. The simulation architecture should allow for the same autonomous behaviors to run either on the platform in real flight or in simulation. New behaviors and code can be tested in simulation first, before being certified for tests in autonomous flight.

71.2.4 On-Board Versus Off-Board Computation

Current and future unmanned systems will generate massive amounts of unstructured data that must be cataloged, stored, analyzed, and shared. Currently, there are more hours of video being generated than analysts are able to watch and UAV operators are facing the problem of a “data deluge” (Dillow 2011). As the number of unmanned systems increases, along with the number of sensors being carried, current procedures will not be sufficient to handle the data, analysis, and visualization requirements.

While there are efforts underway to analyze this data automatically, UAV designs will need to support this requirement as well: UAVs should process as much of the data onboard as possible because of security risks, lack of analysts to process the data, and lack of bandwidth; moreover, sensor data could be fused in real time where it makes sense. Finally, this would allow the autonomous behaviors to consume the output of this data and to respond quickly.

71.2.5 Interoperability and Standards

As unmanned vehicles (including air, ground, and sea) proliferate in dynamic domains, it will become necessary for autonomous, heterogeneous vehicles to coordinate with various ground stations and other manned and unmanned platforms as well. Unmanned vehicles may come from different manufacturers, may have different operators and owners, and may also have overlapping goals. However, as these vehicles become more autonomous, there will be opportunities for vehicle-to-vehicle cooperation and sharing, in order to improve efficiencies. Furthermore, as vehicles may be built by different manufacturers and be under the control of different command centers, interoperability will be important. As STANAG-4586 is becoming more mature in the unmanned systems community, there is a considerable push for new systems to be STANAG-4586 compliant. The SAE AS-4/Joint Architecture for Unmanned Systems (JAUS) standard was developed to provide an open architecture for unmanned ground systems. JAUS defines a message set and data format for passing messages between systems and also between individual components within a system. However, it may make sense for some UAVs to be able to send JAUS messages to communicate with certain ground stations or unmanned ground vehicles. The software architecture described above easily supports the incorporation of such standards.

71.3 Behavior-Based Autonomy

71.3.1 Mission-Level Autonomy

Mission-level autonomy is designed to work in a behavior-based framework, in which behavioral autonomy components are encapsulated into reusable modules or behaviors. Various sets of behaviors can be developed and unit tested before being incorporated into the system. Behaviors can encapsulate a basic and isolated action profile that represents the current state of the mission, such as navigating a search pattern or commanding waypoints, or higher-level operations, such as performing a cooperative task. For an excellent overview of behavior-based autonomy, see Arkin's overview (Arkin 1998).

In this architecture, behaviors are sequenced using a finite state machine that defines the states or behaviors that are possible and the events or triggers that cause

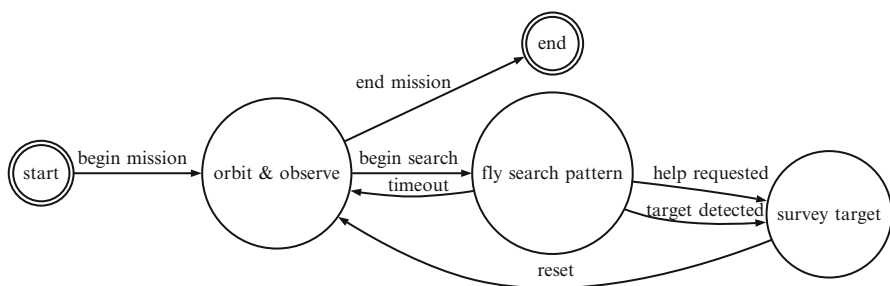


Fig. 71.3 Behavior-based autonomy. A behavior-based autonomy approach is used to encapsulate higher-level autonomy into reusable components. Behaviors are connected together with events to form a finite state machine

the robot to change from one behavior to another. Different inputs available to the behaviors could be used by the triggers to force a behavior change, such as the location of targets or through other events such as receiving a message from team members.

Consider the example behavior state machine, as shown in Fig. 71.3, which is used on this platform to perform a cooperative target search and detection mission. After the UAVs are safely at altitude and switched to autonomous operation, the UAVs enter the *Fly-Orbit* behavior. In this behavior, the UAV is in a “standby” mode and the mission processor simply gathers and logs data from the sensors but does not send commands to the autopilot. When the ground station mission operator sends a *Start-Mission* command, the behavior state machine switches to the *Fly-Search-Pattern* behavior. At this point the behavior commands the autopilot to follow a predetermined search pattern (defined by a set of waypoints.) However, a similar behavior could be written to provide low-level bank angle commands if desired. While the UAV performs the search pattern, the behavior is listening for messages from the other UAVs and UGV on the team (the *Help-Requested* message). It is also listening for messages from the sensors indicating that a target was found (*Target-Detected* message). If the computer vision process locates the target or one of the other UAVs locates the target and sends a request for help over the communications link, the current behavior becomes the *Survey-Target* behavior. At this point, the behavior will send new waypoints to the autopilot, dynamically commanding it to fly a new path around the target location.

Now, consider another example in which UAVs cooperatively negotiate task assignments with each other. In task assignment scenario, each available platform is assumed to be in a state where it can accept tasks. In some cases, these platforms may be autonomous vehicles or they may be manned or remotely operated assets. In either case, the vehicles may be in a holding pattern state, ready to be tasked with new requests. Once a request is assigned, the vehicle begins performing tasks, entering a *Perform Tasks* state, until either all tasks are complete or another event causes the vehicle to be removed from further consideration (such as a

low-fuel condition). If a vehicle has existing tasks to perform when it becomes unavailable, the system will reassign those tasks using the same agent-based negotiation mechanism.

71.3.1.1 High-Level Mission Planning

An important concept behind behavior-based autonomy is that these behaviors can be packed into reusable components that can be tested individually and combined into higher-level missions, such as the example described above. Therefore, for these systems to be useful, operators will need an interface for planning high-level missions, instead of low-level waypoint by waypoint routes.

Behavior-based autonomous systems will require a mission editor which allows operators to choose behaviors from a library of reusable behaviors and configure them into high-level missions. Each behavior is incorporated into a “state machine” plan that represents the event-based transitions between behaviors. There could be separate plans for multiple different missions, and these could be reused or combined into higher-level plans as well. An example of a graphical editor for allowing users to plan behavior-based missions is the MissionLab editor (Endo et al. 2004).

The behavior-based approach to mission planning and execution is a paradigm shift from the current approach to operations. Rather than sending a UAV to a particular location or set of locations, an operator would request that the UAV perform search and surveillance operations over a given area by loading the related behavior-based plan and inputting the specific areas to cover and flight constraints. The operator need not bother with low-level turn and waypoint commands, but allow the onboard planner to calculate them and notify the operator if there is a constraint violation. In addition, during flight, the operator would observe which state the UAV was operating in and be informed of any transitions between states. The next section describes how the behavior framework fits into an overall software architecture.

71.3.2 Behavior-Based Architecture

The recommended UAV autonomous control architecture consists of a layered control system that includes three layers, as shown in Table 71.1. These layers are (1) a deliberative, high-level mission executive, (2) a set of behaviors and state machines, and (3) a low-level reactive controller layer. The mission executive layer performs high-level planning and generates trajectories that can be executed by the UAV. The generated trajectories are segmented into local waypoints that are sent to the behaviors. The behaviors and state machine reflect the current state of the UAV autonomous operation. The behavior layer can request the generation of new plans and implement the behavior logic.

The reactive controller provides low-level control signals for the UAV, regardless of the mode of operation or current mission assignment, and provides for a fast response to ensure vehicle safety. The reactive layer should be tightly integrated with the autopilot and, in some cases, may fully reside within the autopilot. This architecture allows for a fast and responsive set of low-level reactive behaviors

Table 71.1 Control layers

| Layer | Description | Frequency | Outputs |
|-----------------------------------|--|----------------------------|----------------------------|
| Mission executive | Deliberative, high-level global planner | Low frequency | Candidate paths |
| Behaviors and state machine | Processes sensors and executes behaviors | Medium frequency | Waypoints, speed, altitude |
| Reactive controller and autopilot | Generates control signal | High frequency (fast loop) | Low-level control signals |

that allow the vehicle to navigate safely as perception is updated in a dynamic environment, while allowing for full deliberation and higher-level planning. The following sections discuss each of these modules in more detail.

71.3.2.1 Mission Executive

A low-frequency loop, the mission executive, generates trajectories based on mission goals, mission constraints (such as no-fly zones), and a priori knowledge. Global map information is used as an input to this layer. Additionally, regular updates from the perception modules can be incorporated to reflect changes in a dynamic environment. The planner also considers additional constraints for the vehicle, such as flight characteristics, fuel levels, and wind vectors to further refine the candidate path plans.

The high-level deliberative planner, running in a slower executing loop, considers the global environment of the robot to generate candidate path plans for the robot to follow. The planner uses the fused sensory information, stored in a global map as input. The global map is built from Perception module and contains far and near obstacles as well as mission waypoints and goals. The deliberative module runs planning algorithms to generate a candidate path plan that can be executed by the low-level reactive behaviors. The use of a high-level optimistic planner is necessary because perception data will change with additional exploration and is often less accurate from a distance. The high-level, optimistic planner generates candidate paths that encourage exploration and allow for a margin of error before closing off any path. As the robot navigates closer to obstacles, the mid-level deliberative planner will take additional perception inputs and constraints into account and refine the candidate paths. This level segments the candidate plan into a set of local waypoints that the low-level controllers can follow and these can be sent in sequence to the reactive module.

71.3.2.2 Behaviors and State Machine

This module takes as input dynamic obstacles and information about other vessels, such as vessel intent and heading vectors, and adjusts the candidate path to meet these constraints. It runs at a higher frequency than the planning loop and segments the candidate path into a series of local waypoints, which are sent to the low-level controller, for the UAV to follow.

Behavior-based architectures allow for robot developers to create separate and standalone behaviors for autonomous operations. Each behavior is encapsulated as a single unit and can be tested fully before being integrated into more complex sets of behaviors. In addition, behaviors can be extended and designed to operate together. Behavior arbitration schemes are used to determine which behaviors are allowed to run at a given time and which behaviors control the robot in the situation when there are conflicts. There are many existing behavior-based robot behavior frameworks and arbitration mechanisms that can be leveraged, including DAMN (Rosenblatt 1995), 4D/RCS (Albus 2002), MOOS IvP (Benjamin et al. 2009), and ROS (Quigley et al. 2009).

As described above, the behaviors and state machine reflect the current state of the system, and events trigger transitions between states. The behaviors may request that the planner generate a new plan when new information arrives or a state change occurs. In addition, the behaviors encapsulate the “mission logic.” In a cooperative UAV scenario, the behaviors would contain the task assignment and negotiation logic for cooperating with other UAVs and operators. The behaviors plan in the full configuration space of the robot and veto any operations that would cause the robot to violate safety constraints (such as navigating too close to obstacles or performing unsafe maneuvers).

Depending on the mission requirements and autopilot capabilities, it may be sufficient in the architecture to allow the behaviors to output higher-level commands including waypoints, altitude, and speed to the autopilot, while the autopilot sends the low-level, high-frequency control signals to the vehicle servos. However, in some mission scenarios, it may be desirable to exercise more low-level control over vehicle flight. An example of this might be when the vehicle is navigating in environments with obstacles or changing wind conditions. In that case, the lower-level reactive controller would exist external to the autopilot and send direct turn or bank commands.

71.3.2.3 Reactive Controller

The reactive controller consists of the final interface to the autopilot, which provides the control signal. It may also exist as part of the autopilot in some implementations. It should run in a fast loop, to ensure safe operations of the aircraft. It serves as the safety interface to the autopilot and ensures that if the state of the system has changed since the last behavior update that the UAV will react by sending the correct response to the autopilot to avoid a collision or other system failure.

71.4 Case Study: A UAV Cooperative Research Platform

71.4.1 Introduction

This case study describes an unmanned systems architecture and platform for performing cooperative autonomy research. The primary research aim of this system is to explore distributed, cooperative, and autonomy algorithms for controlling



Fig. 71.4 UAV-UGV team. The system includes multiple small UAVs and a UGV

multiple unmanned systems vehicles. The system consists of multiple small, unmanned aerial vehicles (UAVs) and an unmanned ground vehicle (UGV), shown in Fig. 71.4, that can operate autonomously and accept missions from a ground station. The system leverages open standards and off-the-shelf technologies and can be configured to carry different mission payloads. The vehicles communicate with each other and collaborate in a distributed manner. Real-time telemetry and sensor updates are displayed on the FalconView-based ground station. Over 60 flight demonstrations as well as UAV-UGV collaborative experiments have been performed using this system.

The Georgia Tech Research Institute (GTRI) Collaborative Unmanned Systems Technology Demonstrator (CUSTD) program involved multiple UAVs and a UGV working together in a collaborative, autonomous team. The research platform consists of multiple, tactical UAVs that perform onboard computer vision tasks and run autonomous behaviors. The UGV performs waypoint navigation and obstacle avoidance over a road network. The vehicles all operate autonomously and can exchange mission messages with each other using the JAUS standard. The vehicles perform air to ground teaming in a target detection and surveillance mission as well as distributed task allocation using market-based technologies. For a similar UAV to UGV cooperative mission, see the example in Grocholsky et al. (2006).

In a directed cooperation mission, multiple aerial vehicles search for a target, and once it is found, direct ground vehicles and other air vehicles to the target location to perform autonomous surveillance. This platform is part of our ongoing commitment to Unmanned Systems research at GTRI and can be used both internally and in externally sponsored research to study improvements to Unmanned Systems sensors, operations, and collaborative autonomy.

The CUSTD program also serves as a research platform for demonstrating autonomous task assignment using market-based algorithms. Ongoing work includes the ability for heterogeneous vehicles (vehicles with different capabilities

and different sensor characteristics) to collaborate on a shared mission to distribute tasks in dynamic and decentralized manner. In domains where there are a large number of tasks to be shared between a large number of vehicles with different characteristics, it will be important for vehicles to work together in a distributed and dynamic way to allocate tasks among themselves without (or perhaps in addition to) human direction.

71.4.2 UAV-UGV Teaming

In distributed, heterogeneous, multi-agent teams, agents may have different capabilities and types of sensors. Many real-world scenarios could benefit from having teams of robots with different capabilities working in a cooperative manner. An example is after an earthquake or hurricane disaster, in which multiple locations need to be searched and surveyed for survivors. Detection of forest fires or chemical spills is another example. In such dynamic environments, teams of heterogeneous agents will need to cooperate in real time to perform tasks efficiently.

A common theme explored in air-ground teaming missions is how to best leverage the differences between the UAV and UGV platforms. Small-sized UAVs may be flying at an altitude of 1,000 ft. or more and therefore have a broad field of view. They can cover areas quickly and are not constrained to road networks. However, it is often challenging to localize targets exactly from a moving airframe. Furthermore, because they are moving quickly and at altitude, they cannot get a detailed view of the target as easily as a ground vehicle can.

A UGV, on the other hand, can utilize better localization capabilities and get up close to take sensor readings of suspected targets. Furthermore, a UGV can carry additional payloads and sensors, beyond the weight and power limits of the UAV, and can easily perform long missions. However, the UGV in this case is restricted to navigation over road networks and does not have a full view of the environment at any given time.

The vehicles in this system work as a team and are able to leverage the characteristics of the different platforms: The UAVs are used for searching and surveillance, and the UGV is able to get a close-up view of a target. This work describes the architecture for the UAV research platform and mechanisms for communication between air and ground vehicles and the ground station. An example search and surveillance demonstration is also presented.

71.4.3 UAV Platform

The UAV platform, shown in [Fig. 71.5a](#), leverages off-the-shelf readily available components and is based on a quarter-scale Piper Cub airframe with a base model Piccolo avionics and autopilot system from Cloud Cap Technology (Vaglianti et al. 2009). The airframe has a wingspan of 104 in. Two airframes have been tested,



Fig. 71.5 UAV platform. (a) The UAV platform is configurable for different mission requirements, payloads, and sensors. (b) The basic research payload (on the *right*) includes a mission processor, datalink, and digital camera. The mission processor connects to the autopilot (small box on the *left*) over the serial port

one with a four-stroke 20 cc gasoline engine, and the other with a two-stroke 26 cc gasoline engine. The different engine configurations allow for either more power or endurance as the mission dictates. The flight duration is up to 90 min, depending on mission tasks and payload. There is room available onboard for scientific payloads and sensors, configurable to the missions; see [Fig. 71.5b](#). In addition to the autopilot package, each airframe can carry up to seven pounds of payload or sensors. Over 60 field tests of this platform were performed in 2010, including multi-UAV cooperative autonomy and UAV-UGV teaming demonstrations.

71.4.3.1 Flight Operations and Safety

During flight operations, a human pilot manually takes off and lands each UAV. A human operator also monitors the UAV health (GPS status, signal strength, speed, engine RPMs, etc.) using the autopilot operator interface component of the ground station. For safety, each UAV is flown at a separate altitude with sufficient separation. Once the vehicle is at the safe altitude, the human pilot switches the vehicle into autonomous operation. At this point the UAV will begin to track toward the first waypoint in its current route. At any time, the flight operator can change the waypoints and upload them to the vehicle. The operator can also send a command to

the autopilot to shut down any onboard payload should the need arise. Additionally, the pilot can take over manual control at any point during the flight. In the event that the autopilot system loses communication with the ground station, the UAV will automatically return to the safe waypoint and orbit. Lastly, the platform also includes a homing beacon for emergency recovery.

71.4.3.2 UAV System Architecture

The UAV system architecture consists of the autopilot system for low-level flight control and the autonomy payload for high-level mission control. The flight control modules and high-level autonomy modules have been logically and physically separated to provide for flight safety and ease of configuration. The autopilot system provides for low-level control of the platform and can be given a series of waypoints or control loops using the corresponding operator interface in the ground station. The UAV platform can be flown without the autonomy or other sensor payload, using only the autopilot to follow waypoints. This may be desirable if there is a mission requirement to carry a bulky sensor and gather data for offline analysis.

The autopilot module is self-contained and includes a dedicated communications link, power supply, and charging circuit. In addition the platform also carries a dedicated power supply for the engine's ignition. The autopilot system communicates with the ground station via a dedicated 2.4 GHz data-link. The avionics components include a flight control system (flight computer), a GPS/IMU and barometric altimeter sensor for localization, and a waypoint following autopilot for control. Position updates are sent to the ground station at 4 Hz and the core autopilot processes sensor data and controls airframe servos at a rate of 20 Hz. In addition a magnetometer is connected to the autopilot to provide for accurate heading estimates.

Higher-level mission control is provided by the autonomy payload and mission processor. The mission processor receives sensor data and sends new waypoint and turn rate commands to the autopilot. The mission processor communicates with the autopilot's flight control system over a serial port. The autonomy payload includes an onboard PC-104 Intel Atom 1.6 GHz mission processor running the Linux operating system, a USB digital camera with a 1/3" CCD sensor and interchangeable lenses (8, 16, and 25 mm for wide angle and zoom views), and a Digi 900 MHz radio. This radio serves as the mission-level communications link for and agent-agent communications. Each of the UAVs and UGV can use this link to send messages directly to each other or to the ground station. The autonomy payload is self-contained and can easily be removed to perform experiments with different sensor or payload configurations. For example, in other experiments an infrared camera payload was carried to gather sensor data from two different IR cameras for offline analysis.

The vehicle is also equipped with an analog color video camera that transmits video to the ground over a separate link. This video is captured by the ground station and stored in digital format for use by the base station operators and for after-action reports. An example video frame is shown in [Fig. 71.6a](#).



Fig. 71.6 Analog video feeds. (a) The UAV platform captures video during flights and transmits it to the ground station. (b) The UGV platform also streams video to the ground station operators

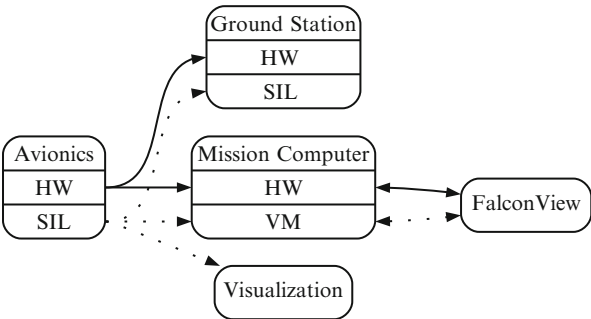


Fig. 71.7 The UAV simulation architecture can support the same autonomous behaviors in real flight or simulation

71.4.3.3 Simulation

The software architecture allows for full simulation of the flight control and mission processor as shown in Fig. 71.7. The Piccolo autopilot provides hardware-in-the-loop (HIL) and software-in-the-loop (SIL) simulation capabilities. This allows for the full simulation of the autopilot and groundstation over a given area. The mission computer is simulated on a Linux virtual machine (VM) and communicates with the simulated autopilot processes over the network. The same autonomous behaviors that execute during a real flight are used during simulation. The datalinks are also simulated by exchanging the hardware radios with a software process that sends the packets over the network. The mission messages and vehicle telemetry are also sent to FalconView™ over the simulated radio. Finally, the output from the autopilot simulation can be fed to a visualization and rendering engine, such as the MetaVR™ tool.

The UAV mission processor’s software architecture is open and extensible and includes the use of open-source technologies and tools. Software processes are loosely coupled and communicate using a publish-subscribe messaging architecture,

LCM (Huang et al. 2010). Some of the lower-level processes and libraries are written in C++, such as the computer vision processes, and high-level autonomous behaviors are written in Java. Both custom and 3rd party low-level libraries and drivers are used to encapsulate access to hardware and standards.

As an example of interprocess communication using LCM, the image processing software component uses pose information to stamp images with position and to perform target localization. The avionics process handles messages coming from the autopilot and publishes a pose message (containing position and attitude information). Other processes, such as filter or image processing, can subscribe to these messages and use them, while posting messages of their own. The use of a publish subscribe mechanism allows new processes and sensors to be more easily integrated onto the platform. For example, a new sensor could include a process that reads data from the sensor and consumes the pose messages from the existing system. The sensor could post new messages that additional behaviors could consume or replace existing message types (such as a target-found message).

71.4.3.4 Aerial Target Detection and Localization

An important component of the system is aerial target localization. The UAV platform currently uses a computer vision-based mechanism for target detection; although additional techniques or sensors could easily be integrated into the system. Aerial Target Detection and Localization combines the ability to capture images from the air, process them using computer vision to find a target in the image, and to calculate the location of that target on the ground, given the position of the UAV.

The UAV's onboard digital camera is affixed to the underside of the fuselage and points down at a target on the ground. The UAV and the camera each have their own coordinate frame. The vision system combines the ability to (1) capture images while in the air, (2) process them in real time using computer vision for target detection, and (3) compute the estimated GPS ground location of the detected target. Figure 71.8 displays a sample image of ground features from Ft. Benning. The image contains the colored tarps that are used as a sample target. Target detection is achieved using color-based recognition methods that rely on a trained model of color thresholds. The UAV's pose is recorded when the image is snapped and is defined by the UAV's position (latitude, longitude, elevation) and attitude (roll, pitch, yaw). In combination with the position and orientation of the camera's coordinate frame with respect to the UAV (extrinsic parameters) and knowledge of the camera's sensor size, lens configuration, and pixel resolution (intrinsic parameters), along with the detected target's pixel coordinates in the image, the target's estimated coordinates in the real world may be computed, i.e., localized.

71.4.3.5 Behavior-Based Autonomy

Mission-level autonomy is designed to work in a behavior-based framework, in which behavioral autonomy components are encapsulated into reusable modules



Fig. 71.8 UAV onboard computer vision. The mission processor processes images from the onboard digital camera in real time. The target location is at the colored tarps in the center of the images. The UGV navigated to the target location after the target was detected from the UAV. Images are shown from each of the 2 UAVs, with different camera lenses and flying at different altitudes (a) 8 mm lens at 300 m. (b) 16 mm lens at 250 m

or behaviors. Various sets of behaviors can be developed and unit tested before being incorporated into the system. Behaviors can encapsulate simple low-level control, such as flying a search pattern or commanding waypoints, or higher-level operations, such as performing a cooperative task.

71.4.4 UGV Platform

The UGV platform consists of a 2006 Porsche Cayenne SUV automobile that has been modified for autonomous operation, shown in Fig. 71.4. This vehicle was developed as Georgia Tech's entry into the DARPA Urban Challenge. (http://www.darpa.mil/grandchallenge/TechPapers/Sting_Racing.pdf). The vehicle is equipped with stereo vision for lane navigation and LIDAR and radar sensors for obstacle detection. The vehicle performs autonomous waypoint navigation using the filtered GPS and IMU sensor output. The system uses a standard D* algorithm for graph-based path planning over a list of ordered waypoints. The vehicle starts with a known map of the navigable environment, a Route Network Definition File (RNDF), which contains metadata about points that define road segments. The vehicle can be given a set of waypoints to visit in advance or can be sent a current waypoint using the JAUS standard messages. The LIDARs and radar are used to detect static and moving obstacles in the environment and provide 360° coverage around the vehicle. The UGV also is equipped with the same Digi 900 MHz radio as the UAVs for agent-agent communications. The vehicle also carries an analog video camera and transmitter. A sample video frame is shown in Fig. 71.6b. This platform can be configured to carry different sensors as mission requirements dictate. For example,

in one experiment, the UGV carried a chemical sensor as part of an ammonia detection mission and sent sensor readings back to the ground station.

The vehicle navigates using a hybrid approach of behavior-based reactive control which references an internal, local map, and a graph-based planner that references the RNDF road network. The reactive controllers, similar to the behavior-based controllers described in Wooden et al. (2007), keep the vehicle clear of obstacles, while the high-level planner provides a route to the waypoint locations.

71.4.5 JAUS Standard

In support of collaborative autonomy, the vehicles must have a common capability to request assistance from each other when a target of interest is located. The Joint Architecture for Unmanned Systems (JAUS) is a standard that allows for greater interoperability between unmanned systems. As an example, each of the vehicles in the system and also the base station can send JAUS GO-TO-WAYPOINT messages to the team or a specific vehicle to request that they navigate to a new area. The standard includes a message set, protocol, and was originally an initiative of the U.S. Department of Defense for facilitating open architectures for unmanned systems. Interoperability is an important focus area for future unmanned systems, as vehicles of different types and from different vendors will need to cooperate with each other.

The vehicles in this system can each exchange JAUS messages with the other vehicles and the ground station. The benefit of the JAUS messaging capability on this platform is that vehicles from other vendors that also implement JAUS may be easily integrated into this system for performing joint field tests. This project uses the OpenJAUS (<http://www.openjaus.com>) implementation for creating JAUS standard messages between the air and ground platforms.

71.4.6 Ground Station

The system also includes a FalconViewTM-based ground station, as shown in Fig. 71.9. FalconViewTM is widely used by the U.S. Department of Defense for its aircraft mission planning and mapping capabilities. The open-source version of FalconViewTM is now freely available (<http://www.falconview.org>). FalconViewTM provides for application extensions through a plug-in framework. The vehicles in this system communicate to the FalconViewTM application plug-in through a ground station mission server to display vehicle position and telemetry information in real time, using JAUS standard messages. The ground station can also be used to send JAUS messages to the vehicles and to send other waypoint and mission-based commands. As the vehicles in the system perform the mission, the vehicles' behavioral state (such as *Search* or *Survey*) is displayed on the FalconViewTM map, along with the vehicle positions. Detected target locations are updated on the map as well.



Fig. 71.9 Using the FalconView™-based ground station, multiple UAVs and UGVs can be displayed, with video overlays and system status. The FalconView™ plugin can also send mission-specific commands to the mission processor

71.4.7 Cooperative Target Detection and Surveillance

In one type of a cooperative scenario, all vehicles participate in a search, detection, and surveillance mission. For example, a team of UAVs and a UGV can perform target surveillance over a wide area by flying a set of predefined waypoints, with the UGV waiting in standby. When one of the UAVs locates the target, it sends a message to team members with the target location. At that time, the team members can all navigate to the target location and perform surveillance.

This field experiment was performed as part of the Army's Robotics Rodeo in October 2010, at the McKenna site at Ft. Benning, GA, shown in Fig. 71.6a. The test location includes a network of roads and a long runway, along with a village or Military Operations on Urban Terrain (MOUT) setting. The purpose of this experiment was to demonstrate air-ground cooperation. Two UAVs autonomously flew search patterns covering the road networks and the village, with the UGV in standby mode at the far end of the test site. A fixed target, consisting of colored tarps, was placed inside the village. The UAV mission computers processed 2–3 image frames per second onboard, using basic color thresholds, to detect the target, as shown in Fig. 71.8.

When the first UAV spotted the target on the ground, the mission processor sent new waypoint commands to the autopilot and it autonomously changed course to survey the target. It also requested help from the other UAV and UGV by sending a JAUS message to each vehicle. In this case, the vehicles were setup to resend the JAUS messages until an acknowledgement was received from each team member. The other UAV received the message and also autonomously changed course to survey the target. The UGV was given the target location and navigated autonomously to the target area to perform more detailed surveillance. After all of the vehicles jointly surveyed the target for several minutes, the mission was completed.

71.4.8 Case Study Summary

This case study presented the design of a reusable and configurable UAV-UGV research platform for performing joint air-to-air and air-to-ground cooperative missions. The system can process sensor data onboard and run high-level autonomy algorithms for performing collaborative, autonomous missions. This platform has been field tested in multiple flight configurations, including a demonstration of a cooperative detection and surveillance task involving two UAVs and a UGV.

71.5 Conclusion

This chapter presented desiderata for systems designers to consider when building autonomous UAV systems. These desiderata help support the increasing levels of autonomy in UAV systems through behavior-based autonomy, modular and loosely coupled components, and distributed architectures.

Also important to the design of autonomous UAVs is the ability to perform high-level mission planning and to observe the state of autonomous behaviors. In addition, a core concept is the ability to process and store as much of the sensor data onboard as possible to make it immediately available to the autonomous behaviors and to reduce the reliance on ground station operators and analysts.

From a testing and software maintenance viewpoint, by incorporating introspection, open standards, and architectures into the design, these approaches will make the certification process easier and decrease maintenance costs. Finally, these approaches will free operators to focus on high-level tasks and help to make UAV architectures more scalable as a result.

References

- J.S. Albus, 4D/RCS: a reference model architecture for intelligent unmanned ground vehicles, in *Proceedings of SPIE Aerosense Conference*, Orlando, 2002, pp. 1–5
- R.C. Arkin, *Behavior-Based Robotics* (MIT, Cambridge, 1998)

- M.R. Benjamin, J.J. Leonard, H. Schmidt, P.M. Newman, A tour of moos-ivp autonomy software modules. Technical report MIT-CSAIL-TR-2009-006, MIT Computer Science and Artificial Intelligence Laboratory, 2009
- C. Dillow, Can technology save the military from a data deluge? Pop. Sci. (2011). <http://www.popsoci.com/technology/article/2011-11/can-technology-save-military-its-technology>
- Y. Endo, D.C. MacKenzie, R.C. Arkin, Usability evaluation of high-level user assistance for robot mission specification. *IEEE Trans. Syst. Man Cybern. C: Appl. Rev.* **34**(2), 168–180 (2004). ISSN:1094-6977. doi:10.1109/TSMCC.2004.826285
- B.P. Grocholsky, J. Keller, V. Kumar, G. Pappas, Cooperative air and ground surveillance. *IEEE Robot. Autom. Mag.* **13**(1), 16–25 (2006)
- A.S. Huang, E. Olson, D.C. Moore, LCM: lightweight communications and marshalling, in *Proceedings of the International Conference Intelligent Robots and Systems (IROS)*, Taipei, Oct 2010, <http://code.google.com/p/lcm/>
- M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, ROS: an open-source robot operating system, in *ICRA Workshop on Open Source Software*, Kobe, Japan, 2009
- J.K. Rosenblatt, DAMN: a distributed architecture for mobile navigation, in *Proceedings of the AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents*, Stanford, 1995
- J. Tisdale, A. Ryan, M. Zennaro, X. Xiao, D. Caveney, S. Rathinam, J.K. Hedrick, R. Sengupta, The software architecture of the Berkeley UAV platform, in *IEEE Conference on Control Applications*, Munich, Oct 2006
- B. Vaglienti, R. Hoag, M. Niculescu, J. Becker, D. Miley, *Piccolo User's Guide (v2.1.0)*. Cloud Cap Technology, www.cloudecaptech.com, 14 Oct 2009
- D. Wooden, M. Powers, D. MacKenzie, T. Balch, M. Eggerstadt, Control-driven mapping and planning, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 07)*, San Diego, Oct 2007, pp. 3056–3061

Minyi Zhong and Christos G. Cassandras

Contents

72.1 Introduction 1750

72.2 Asynchronous Distributed Optimization Framework 1753

72.3 Convergence Analysis 1757

 72.3.1 Optimization with State Update Noise 1759

 72.3.2 Optimization with Non-negligible Communication Delays 1760

72.4 Asynchronous Distributed Coverage Control..... 1764

 72.4.1 Coverage Control Problem Formulation 1764

 72.4.2 Asynchronous Versus Synchronous Optimal Coverage Control 1766

72.5 Conclusion 1771

References 1772

Abstract

This chapter considers problems where multiple autonomous UAVs cooperate to control their individual state so as to optimize a common objective while communicating with each other to exchange state information. Since communication costs can be significant, especially when the UAVs operate with limited energy, conditions are sought under which communication of state information among nodes can be restricted while still ensuring that the optimization process converges. An asynchronous, event-driven scheme is described which limits communication to instants when some state estimation error function at a node exceeds a threshold. This scheme guarantees convergence to an optimal

M. Zhong (✉)
Server and Tools Division, Microsoft Corporation, Redmond, WA, USA
e-mail: mizhon@microsoft.com; dominozhong@gmail.com

C.G. Cassandras
Division of Systems Engineering, Center for Information and Systems Engineering,
Boston University, Brookline, MA, USA
e-mail: cgc@bu.edu

even in the presence of communication delays as long as they are bounded. This approach is applied to a UAV network coverage control problem where the objective is to maximize the probability of detecting events occurring in a given region, and it is shown that it may significantly reduce communication costs, hence also prolonging the system's lifetime, without any performance degradation.

72.1 Introduction

In this chapter, teams of autonomous UAVs cooperating toward a common objective are considered. What makes the UAVs “autonomous” is their ability to optimize their own actions while limiting communication with the remaining team members as much as possible. The cooperating UAVs define a dynamic system which may be thought of as a network with each UAV corresponding to a node maintaining its own state s_i , $i = 1, \dots, N$. The goal of each node is to control its state so as to optimize some system-wide objective expressed as a function of $\mathbf{s} = [s_1, \dots, s_N]$ and possibly the state of the environment. Clearly, to achieve such a goal in a dynamic and uncertain environment, the nodes must share, at least partially, their state information. However, this may require a large amount of information flow and becomes a critical issue when the UAVs operate with limited resources. Aside from energy required to move, communication is known to be by far the largest consumer of the limited energy of a node (Shnayder et al. 2004), compared to other functions such as sensing and computation. Moreover, every communication among nodes offers an opportunity for corruption or loss of information due to random effects or adversarial action. Therefore, it is crucial to reduce communication between UAVs to the minimum possible. This in turn imposes a constraint on the optimization task performed by each node, since it requires that actions be taken without full knowledge of other nodes' states. Standard synchronization schemes require that nodes exchange state information frequently, sometimes periodically, which can clearly be inefficient and, in fact, often unnecessary since it is possible that (i) system inactivity makes the periodic (purely time driven) exchange of information unnecessary, (ii) occasional state information is adequate for control and/or optimization mechanisms which do not require perfect accuracy at all times, and (iii) state information of other nodes can be estimated reasonably well and explicit communication is thus redundant. This motivates an effort to seek *asynchronous* optimization mechanisms in which a node communicates with others only when it considers it indispensable; in other words, each node tries to reduce the cost of communication by transmitting state information only under certain conditions and only as a last resort. This poses questions such as “what should the conditions be for a node to take such communication actions?” and “under what conditions, if any, can we guarantee that the resulting optimization scheme possesses desirable properties such as convergence to an optimum?”

The general setting described above applies to problems where the nodes are UAVs controlling their locations and seeking to maintain some desirable formation (Ögren et al. 2004; Lawton et al. 2000) while following a given trajectory.

The UAV team may also be operating as a sensor network with UAVs located so as to achieve objectives such as maximizing the probability of detecting events in a given region or maintaining a desired distance from targets or data sources that ensures high-quality monitoring (Meguerdichian et al. 2001; Cortes et al. 2004; Ganguli et al. 2006; Zou and Chakrabarty 2003; Mihaylova et al. 2002; Cassandras and Li 2005; Hussein and Stipanovic 2007; Hokayem et al. 2007); this is often referred to as a “coverage control” problem. In some cases, the state of a UAV may not be its location but rather its perception of the environment which changes based on data directly collected by this UAV or communicated to it by other UAVs; consensus problems fall in this category (DeGroot 1974; Vicsek et al. 1995; Jadbabaie et al. 2003; Moreau 2005; Nedić and Ozdaglar 2009).

In this chapter, a UAV team is viewed as a network of N cooperating nodes. The team’s goal is to minimize an objective function $H(\mathbf{s})$ known to all nodes with every node controlling its individual state $s_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$. The state update scheme employed by the i th node is of the general form

$$s_i(k+1) = s_i(k) + \alpha_i d_i(\mathbf{s}(k)), \quad k = 0, 1, \dots \quad (72.1)$$

where α_i is a constant positive step size and $d_i(\mathbf{s}(k))$ is an *update direction* evaluated at the k th *update event* (see also Bertsekas and Tsitsiklis 1997). We often use

$$d_i(\mathbf{s}(k)) = -\nabla_i H(\mathbf{s}(k))$$

where $\nabla H(\mathbf{s}(k))$ is the gradient of $H(\mathbf{s}(k))$ and $\nabla_i H(\mathbf{s}(k)) \in \mathbb{R}^{n_i}$. In general, each state is characterized by dynamics of the form $\dot{s}_i(t) = f_i(s_i, u_i, t)$ where $u_i \in \mathbb{R}^l$ is a control vector; for our purposes, however, we treat s_i as a directly controllable vector. Thus, in (72.1) we view $s_i(k+1)$ as the *desired* state determined at the k th update event and assumes that the control u_i is capable of reaching $s_i(k+1)$ from $s_i(k)$ within a time interval shorter than the time between update events.

A key difficulty in (72.1) is that $\mathbf{s}(k)$ is in fact not fully known to node i . Thus, $d_i(\mathbf{s}(k))$ has to be evaluated by synchronizing all nodes to provide their states to node i at the time its k th update event takes place. This is extremely costly in terms of communication and assumes no delays so that the state information is accurate. Alternatively, node i can evaluate $d_i(\mathbf{s}(k))$ using estimates of s_j for all $j \neq i$ relying on prior information from node j and possibly knowledge of its dynamics. A key concern is to determine instants when a node j may communicate its state to other nodes through what are termed *communication events*. It should be noted that such communication events occur at different times for each node, as do each node’s state update events, so that the resulting mechanism is asynchronous. A distributed optimization framework is necessitated by the fact that nodes operate in a dynamic and uncertain environment and each node generally has local information not available to others. An example is the appearance of a random obstacle in the coverage control setting mentioned earlier, detected only by a single node.

A scheme is presented through which a node j maintains an *error function* of its actual state relative to its state as estimated by other nodes (which node j can evaluate). The node then transmits its actual state at time t only if this error function at t exceeds a given *threshold* δ_j . In other words, a node does not incur any communication cost unless it detects that the deviation of its state from the other nodes' estimate of its state becomes too large; this may happen due to the normal state update (72.1) accumulating noise, imperfect state estimation or through unexpected state changes (e.g., if a mobile node encounters an obstacle). First, assuming negligible communication delays, it was shown in Zhong and Cassandras (2010) that by varying this threshold appropriately and under certain rather mild technical conditions, the resulting optimization scheme converges and leads to a minimum of $H(\mathbf{s})$; this minimum may be local or global depending on the nature of the objective function. Our approach controls the asynchronous occurrence of communication events through the threshold-based scheme outlined above in a way that may drastically reduce the number of such events while still guaranteeing convergence. Our analysis can also be extended to cases where an explicit noise term is included in (72.1), as well as when we allow communication delays to be non-negligible, provided there exists an upper bound in the number of state update events that occur between the time of a communication event initiated by a node and the time when all nodes receive the communicated message.

In the second part of the chapter, this approach is applied to a coverage control problem in which a distributed optimization scheme based on (72.1) is used in order to deploy UAVs in a region (possibly containing polygonal obstacles) so as to maximize the probability of detecting events (e.g., unknown data sources) in this region. In earlier work (Cassandras and Li 2005), it was assumed that all nodes have perfect state information by synchronizing update events with communication events. This imposed significant communication costs. Here, this synchronization requirement is relaxed and communication events are limited to occur according to the new proposed event-driven policy leading to convergence to the optimum. Simulation results are included to show that the same performance is attained with only a fraction of the original communication costs.

Our work on asynchronous distributed optimization with event-driven communication was introduced in Zhong and Cassandras (2008a) with no communication delays in the convergence analysis. Such event-driven communication is also used in collaborative estimation (as opposed to optimization) (Shima et al. 2007) where a node transmits data to other nodes only when a computable estimation error exceeds some fixed threshold. In Wan and Lemmon (2009), an augmented Lagrangian method based on event-triggered message passing is developed to obtain an approximate solution of the network utility maximization problem.

The remainder of the chapter is organized as follows. Section 72.2 describes our asynchronous distributed optimization framework and the proposed scheme for communication events. The convergence analysis is reviewed in Sect. 72.3, including the case where communication delays are present. In Sect. 72.4, it is shown how our approach applies to a coverage control problem for UAV teams, and conclusions are given in Sect. 72.5.

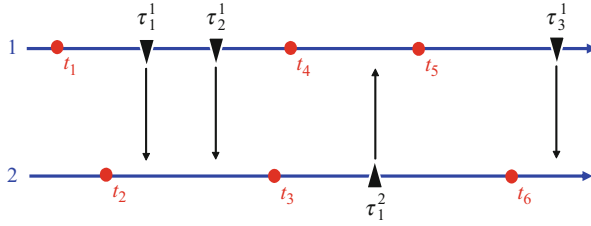


Fig. 72.1 State update and communication processes for two nodes. *Dots* represent state update events at a node, and *triangles* represent state communication events

72.2 Asynchronous Distributed Optimization Framework

In a setting where N cooperating nodes (the UAVs) seek to optimize a common objective function, there are two processes associated with each node: a *state update process* and a *state communication process*. A discussion of the state update process comes first.

Let $t_k, k = 1, 2, \dots$, denote the time when any one node performs a state update, that is, it takes an action based on (72.1). No constraint is imposed on when precisely such an update event occurs at a node, and it is allowed to be either periodic or according to some local node-based policy. However, it will be assumed that every node performs an update with sufficient frequency relative to the updates of other nodes (this assumption will be stated precisely later).

Let C^i be the set of indices in $\{t_k\}$ corresponding to update events at node i . As an example, in Fig. 72.1, where nodes 1 and 2 perform state updates at $\{t_1, t_4, t_5\}$ and $\{t_2, t_3, t_6\}$, respectively, one can see that $C^1 = \{1, 4, 5\}$ and $C^2 = \{2, 3, 6\}$. For all $k \notin C^i, d_i(s(k)) = 0$ in (72.1), that is,

$$s_i(k+1) = s_i(k) \quad \text{if } k \notin C^i$$

Any state update at such $k \notin C^i$ is referred to as a *null step* at node i .

Next, the state communication process is discussed. Let τ_n^j be the n th time when node j broadcasts its true state to all other nodes, $n = 1, 2, \dots$ and $\tau_0^j = 0$ (a more general scheme which allows point-to-point communication will be introduced later in this section). Depending on the network connectivity at that time, it is possible that only a subset of nodes is reached. It is initially assumed that at all communication event times the state information broadcast by node j can reach any other node with negligible delay either directly or indirectly (in the latter case, through a sequence of transmissions), that is, we assume that the underlying network is connected. The negligible communication delay assumption will be relaxed in the sequel. The network connectivity assumption is preserved throughout, although it may be replaced by weaker conditions when the network has nodes that can be decoupled, that is, the local decision of a node is not affected by the state information of some other nodes in the network. For the range of applications that involve

UAV teams, there is usually a “base station” which maintains communication with all autonomous UAVs in the network, thus ensuring full connectivity. However, the existence of such a base station is not required for the general optimization framework proposed. If connectivity is lost by some node, then a related problem is to ensure that the node can control its state so as to reestablish such connectivity. This problem is discussed and resolved elsewhere (Zhong and Cassandras 2011).

Consider now a state update time t_k at node i , that is, $k \in \mathcal{C}^i$. Of interest is the most recent communication event from a node $j \neq i$. Define

$$\tau^j(k) = \max\{\tau_n^j : \tau_n^j \leq t_k, n = 0, 1, 2, \dots\} \quad (72.2)$$

as the time of the most recent communication event at node j up to a state update event at t_k . As an example, in Fig. 72.1 node 1 communicates its state to node 2 twice in the interval (t_2, t_3) ; in this case, $\tau^1(3) = \tau_2^1$. However, no further communication event takes place from node 1 until after the next state update event at node 2 at time t_6 , so that $\tau^1(6) = \tau^1(3) = \tau_2^1$. The policy used by node j to generate communication events is crucial and will be detailed later in this section, but it is emphasized that it is in no way constrained to be synchronized with update events or with the communication events of any other node.

In order to differentiate between a node state at any time t and its value at the specific update times t_k , $k = 0, 1, \dots$, $x_i(t)$ is used to denote the former and observe that

$$s_i(k) = x_i(t_k)$$

Thus, the state of node j communicated to other nodes at time $\tau^j(k)$ may be written as $x_j(\tau^j(k))$. Returning to the state update process, consider some t_k with $k \in \mathcal{C}^i$, and let $s^i(k)$ be a vector with node i 's estimates of all node states at that time, that is, an estimate of $\mathbf{s}(k)$. There are various ways for node i to estimate the state of some $j \neq i$. The simplest is to use the most recent state information received at time $\tau^j(k)$ as defined in (72.2), that is,

$$s_j^i(k) = x_j(\tau^j(k)) \quad (72.3)$$

Alternatively, node i may use a dynamic linear estimate of the form

$$s_j^i(k) = x_j(\tau^j(k)) + \frac{t_k - \tau^j(k)}{\Delta_j} \cdot \alpha_j \cdot d_{j,\tau^j(k)} \quad (72.4)$$

where Δ_j is an estimate of the average time between state updates at node j (e.g., a known constant if node j performs periodic updates) and $d_{j,\tau^j(k)}$ is the update direction communicated by node j at time $\tau^j(k)$ along with its state. Note that $[t_k - \tau^j(k)]/\Delta_j$ is an estimate of the number of state updates at j since its last communication event. More generally, if the precise local decision-making process of j is known to i , then i can evaluate $s_j^i(k)$ using this information with initial condition $x_j(\tau^j(k))$. In this case, the estimate is error-free except for noise that

may have affected the actual state evolution of node j in the interval $[\tau^j(k), t_k]$. In general, the value of an estimate $s_i^j(k)$ used by node i to estimate node j 's state depends on t_k , the most recent communication event time $\tau^j(k)$, and the actual state $x_j(\tau^j(k))$ of node j at that time.

Now consider what criterion a node i might use to generate its communication events, recalling that the goal is to reduce communication costs. If node i knows that node j uses a specific method to estimate its state, then node i can evaluate that estimate and hence the error in it at any time. If $x_i^j(t)$ is the estimate of $x_i(t)$ evaluated by node $j \neq i$ at time t , an estimation *error function* $g(x_i(t), x_i^j(t))$ is defined which measures the quality of the state estimate of node i with the requirement that

$$g(x_i(t), x_i^j(t)) = 0 \text{ if } x_i(t) = x_i^j(t) \quad (72.5)$$

Examples of $g(x_i(t), x_i^j(t))$ include $\|x_i(t) - x_i^j(t)\|_1$ and $\|x_i(t) - x_i^j(t)\|_2$. Let $\delta_i(k)$ be an error *threshold*, determined by node i after the k th state update event such that $k \in C^i$. Thus, $\delta_i(k) = \delta_i(k-1)$ if $k \notin C^i$. Let \tilde{k}_t^i be the index of the most recent state update time of node i up to t , that is,

$$\tilde{k}_t^i = \max \{n : n \in C^i, t_n \leq t\} \quad (72.6)$$

If different nodes use different means to estimate i 's state, then generally $x_i^j(t) \neq x_i^k(t)$ for nodes $j \neq k$, and communication may be limited to a node-to-node process. Let τ_n^{ij} be the n th time when node i sends its true state to node j . Let us also set $\tau_0^{ij} = 0$ for all i, j . Then, the communication event policy at node i with respect to node j is determined by

$$\tau_n^{ij} = \inf \left\{ t : g(x_i(t), x_i^j(t)) \geq \delta_i(\tilde{k}_t^i), t > \tau_{n-1}^{ij} \right\} \quad (72.7)$$

When a communication event is triggered by (72.7) at τ_n^{ij} , assuming negligible communication delay, x_i^j is instantaneously set to $x_i(\tau_n^{ij})$, that is, $x_i^j((\tau_n^{ij})^+) = x_i(\tau_n^{ij})$. Therefore, the error measure is reset to zero, that is, $g(x_i((\tau_n^{ij})^+), x_i^j((\tau_n^{ij})^+)) = 0$.

If, on the other hand, all nodes use the exact same estimation method, then $x_i^j(t)$ will have the same value for all $j \neq i$, and τ_n^{ij} in (72.7) can be replaced by τ_n^i . In other words, node i communicates its state to all other nodes only when it detects that its true state deviates from the other nodes' estimate of it by at least the threshold $\delta_i(\tilde{k}_t^i)$. Figure 72.2 shows an example of a trajectory of $g(x_i(t), x_i^j(t))$ where the threshold is decreasing over time and the error function is continuous over all intervals $(\tau_{n-1}^{ij}, \tau_n^{ij})$, $n = 1, 2, \dots$. Observe that the negligible communication delay assumption allows the value of $g(x_i(t), x_i^j(t))$ to be instantaneously reset to 0 at τ_n^{ij} , $n = 1, 2, \dots$.

Next, the way in which the threshold $\delta_i(k)$ should be selected is discussed. The basic idea is to use a large value at the initial stages of the optimization process and later reduce it to ultimately ensure convergence. One of the difficulties is in

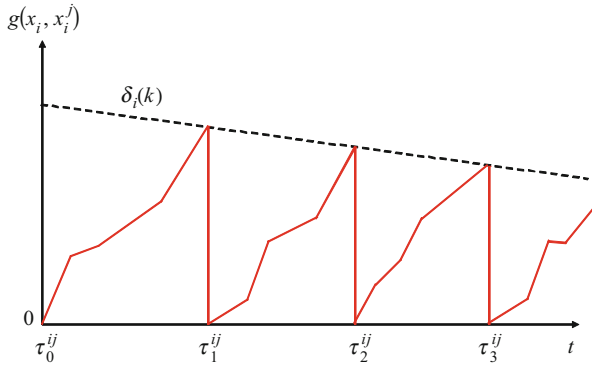


Fig. 72.2 Trajectory of the error function $g(x_i(t), x_i^j(t))$ when the communication delay is negligible

selecting an appropriate initial value for $\delta_i(k)$ which, if too large, may prevent any communication. The approach followed is to control $\delta_i(k)$ in a manner which is proportional to $\|d_i(s^i(k))\|_2$, the Euclidean norm of the update direction at the k th update event henceforth denoted by $\|\cdot\|$. Thus, let

$$\delta_i(k) = \begin{cases} K_\delta \|d_i(s^i(k))\| & \text{if } k \in C^i \\ \delta_i(k-1) & \text{otherwise} \end{cases} \quad (72.8)$$

where K_δ is a positive constant. An initial condition is also imposed such that

$$\delta_i(0) = K_\delta \|d_i(s^i(0))\|, \quad i = 1, \dots, N \quad (72.9)$$

and

$$s_j^i(0) = x_j(0) \quad (72.10)$$

Note that (72.10) can be readily enforced by requiring all nodes to share their initial states at the beginning of the optimization process, that is, $\tau_0^{ij} = 0$ for all i, j ; since generally $g(x_i(0), x_i^j(0)) \geq \delta_i(0)$, this triggers (72.7) and results in $g(x_i((0)^+), x_i^j((0)^+)) = 0$ for all $i = 1, \dots, N$. Also note that since $s^i(k)$ is node i 's local estimate of $s(k)$ at t_k , the computation in (72.8) requires only local information.

Finally, it should be pointed out that the quality of the estimation method used may be critical for effective communication reduction in our event-driven optimization framework. Poor quality state estimation can potentially trigger communication events at a frequency that is higher than that of state updates, especially close to convergence when nodes have small $d_i(s^i)$. Interestingly, when the simplest method, that is, static estimation, is used, this issue does not arise because triggering estimation errors coincide with state update events. Therefore, a node will communicate its real state at most as often as it updates its state.

72.3 Convergence Analysis

In this section, the convergence properties of the asynchronous distributed state update scheme are discussed. For $k = 0, 1, \dots$,

$$s_i(k+1) = s_i(k) + \alpha d_i(s^i(k)) \quad (72.11)$$

is the scheme used by nodes $i = 1, \dots, N$, where $d_i(s^i(k))$ is an update direction which satisfies $d_i(s^i(k)) = 0$ for all $k \notin \mathcal{C}^i$. For simplicity, a common step size α is used, but each node may easily adjust its step size by incorporating a scaling factor into its own $d_i(s^i(k))$. Recall that $s^i(k)$ is the state estimate vector evaluated by node i at the k th update event using the most recent state updates from other nodes at times $\tau^j(k)$ defined by (72.2). The framework in Bertsekas and Tsitsiklis (1997) will be followed. The distinctive feature in our analysis is the presence of the controllable state communication process defined by (72.7)–(72.9) which imposes a requirement on the constant K_δ in order to guarantee convergence. Further, our analysis provides means to select this constant in conjunction with the step size parameter α in (72.11) in a way that may potentially drastically reduce the number of communication events while still guaranteeing convergence.

A number of assumptions are stated below, most of which are commonly used in the analysis of distributed asynchronous algorithms (Bertsekas and Tsitsiklis 1997). Recall that, for the time being, it is assumed that the communication delay is negligible whenever a node informs other nodes of its current states.

Assumption 1. There exists a positive integer B such that for every $i = 1, \dots, N$ and $k \geq 0$ at least one of the elements of the set $\{k - B + 1, k - B + 2, \dots, k\}$ belongs to \mathcal{C}^i .

This assumption imposes a bound on the state update frequency of every node in order to guarantee that the entire state vector will be iterated on. It does not specify a bound in time units but rather ensures that each node updates its state at least once during a period in which B state update events take place. It is worth pointing out that an update event time t_k may correspond to more than one node performing updates.

Assumption 2. The objective function $H(\mathbf{s})$, where $\mathbf{s} \in \mathbb{R}^m$, $m = \sum_{i=1}^N n_i$, satisfies the following:

- (a) $H(\mathbf{s}) \geq 0$ for all $\mathbf{s} \in \mathbb{R}^m$.
- (b) $H(\cdot)$ is continuously differentiable, and $\nabla H(\cdot)$ is Lipschitz continuous, that is, there exists a constant K_1 such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, $\|\nabla H(\mathbf{x}) - \nabla H(\mathbf{y})\| \leq K_1 \|\mathbf{x} - \mathbf{y}\|$.

In what follows, all vectors shall be taken to be column vectors and $'$ will be used to denote a transpose. Let

$$d(k) = [d_1(s^1(k))', \dots, d_N(s^N(k))']'$$

For simplicity, $d_i(k)$ instead of $d_i(s^i(k))$ will henceforth be written.

Assumption 3. There exist positive constants K_2 and K_3 such that for all $i = 1, \dots, N$ and $k \in \mathcal{C}^i$,

$$(a) \quad d_i(k)' \nabla_i H(s^i(k)) \leq -\|d_i(k)\|^2 / K_3$$

$$(b) \quad K_2 \|\nabla_i H(s^i(k))\| \leq \|d_i(k)\|$$

Here, $\nabla_i H(s^i(k))$ denotes a vector with dimension n_i . Its j th component, denoted by $\nabla_{i,j} H(s^i(k))$, is the partial derivative of $H(\cdot)$ with respect to the j th component of s_i . This assumption is very mild and is immediately satisfied with $K_2 = K_3 = 1$ when an update direction given by $d_i(k) = -\nabla_i H(s^i(k))$ is used.

Assumption 4. The error function $g(x_i(t), x_i^j(t))$ satisfies the following:

$$(a) \quad \text{There exists a positive constant } K_4 \text{ such that } \|x_i(t) - x_i^j(t)\| \leq K_4 g(x_i(t), x_i^j(t)) \text{ for all } i, j, t.$$

$$(b) \quad g(x_i(t), x_i^j(t)) \leq \delta_i(\tilde{k}_i^j) \text{ where } \tilde{k}_i^j \text{ was defined in (72.6).}$$

In the common case where $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|$, part (a) of this assumption is obviously satisfied with $K_4 = 1$. Part (b) may be violated if there is a discontinuity when $g(x_i(t), x_i^j(t))$ exceeds $\delta_i(\tilde{k}_i^j)$, in which case one can simply redefine the error function to take the value $\delta_i(\tilde{k}_i^j)$ at all such points, that is, an error function $\bar{g}(\cdot)$ is used instead such that $\bar{g}(x_i(t), x_i^j(t)) = \delta_i(\tilde{k}_i^j)$ if $g(x_i(t), x_i^j(t)) > \delta_i(\tilde{k}_i^j)$, otherwise $\bar{g}(\cdot) = g(\cdot)$. Alternatively, observe that if such a discontinuity occurs at t , then, by (72.7), node i generates a communication event, and, under the negligible delay assumption, node i receives node j 's state and sets $x_i^j(t) = x_i(t)$, hence $g(x_i(t), x_i^j(t)) = 0$ by (72.5); that is, the error is instantaneously reset to zero. Therefore, defining $g(\cdot)$ to be right-continuous guarantees that part (b) is always satisfied. With this understanding, the notation $g(x_i(t), x_i^j(t))$ will still be used, and this assumption is taken to be satisfied, though it is later relaxed.

Theorem 72.1. Under Assumptions 1–4, the communication event policy (72.7) and the state update scheme (72.11), if the error threshold $\delta_i(k)$ controlling communication events is set by (72.8) and (72.9), then there exist positive constants α and K_δ such that $\lim_{k \rightarrow \infty} \nabla H(s(k)) = 0$.

The proof of this theorem and all subsequent theoretical results in this chapter are omitted and can be found in Zhong and Cassandras (2010).

Corollary 72.1. If $0 < \alpha < 2/K_1 K_3$, then

$$K_\delta < \frac{1}{(1+B)K_4\sqrt{m}} \left(\frac{2}{K_1 K_3} - \alpha \right) \quad (72.12)$$

guarantees that $\lim_{k \rightarrow \infty} \nabla H(s(k)) = 0$.

Note that (72.12) provides an upper bound for K_δ that guarantees convergence under the conditions of Theorem 72.1. Obviously, there may be larger values of K_δ under which convergence is still possible.

As already mentioned, one often sets $d_i(s^i(k)) = -\nabla_i H(s^i(k))$ in (72.11) and uses $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|$, in which case Assumption 3 is satisfied with $K_2 = K_3 = 1$ and Assumption 4(a) with $K_4 = 1$. It follows from Corollary 72.1 that $\alpha < 2/K_1$ can be chosen and (72.12) leads to a K_δ arbitrarily close from below to $(2/K_1 - \alpha)[(1 + B)\sqrt{m}]^{-1}$. Observe that, if node states are scalar, this value is inversely proportional to $\sqrt{m} = \sqrt{N}$. Thus, large networks require a smaller value of K_δ , implying that convergence is less tolerant to a node's state estimates evaluated by other nodes and communication needs to be more frequent. For vector node states, the same is true since $m = \sum_{i=1}^N n_i$. Along the same lines, note that K_δ is inversely proportional to B , which means that when there is a larger difference in the state update frequency between the fastest node and the slowest node (larger B), more communication is necessary in order to preserve convergence. Finally, a smaller step size α (slower change of states) allows one to choose a larger K_δ , which means greater tolerance to estimation errors.

72.3.1 Optimization with State Update Noise

As already mentioned, it has been assumed that each node executing (72.1) is capable of reaching a desired new state $s_i(k+1)$ from $s_i(k)$ within a time interval shorter than the time between update events. Thus, any noise present in this update step affects only the time required for $s_i(k+1)$ to be attained. If, however, a noise term $w_i(k)$ is explicitly included in (72.1), then a new state update process must be considered:

$$s_i(k+1) = s_i(k) + \alpha d_i(s^i(k)) + w_i(k) \quad (72.13)$$

The following assumption will be made regarding this added noise process:

Assumption 5. There exists a positive constant $K_w < 2/K_1 K_3$ such that $w_{i,j}(k) \leq K_w d_{i,j}(k)$, where $d_{i,j}(k)$ and $w_{i,j}(k)$ are the j th scalar component of $d_i(k)$ and $w_i(k)$, respectively.

This assumption requires the noise incurred during a state change to generally decrease with the magnitude of the state changes in the sense that it is bounded by $K_w d_{i,j}(k)$, which decreases with k . Then, Theorem 72.1 can be extended to the following result.

Theorem 72.2. Under Assumptions Assumption 1–5, the communication event policy (72.7) and the state update scheme (72.13), if the error threshold $\delta_i(k)$ controlling communication events is set by (72.8) and (72.9), then there exist positive constants α and K_δ such that $\lim_{k \rightarrow \infty} \nabla H(s(k)) = 0$.

72.3.2 Optimization with Non-negligible Communication Delays

Thus far, negligible communication delays have been assumed, that is, the transmission and reception of a message in a communication event has been treated as instantaneous. Under this assumption, when a communication event is triggered by node i at t according to (72.7), a copy of the true state of node i is sent to node j , and the error function $g(x_i(t), x_i^j(t))$ is reset to zero immediately. This was illustrated by the trajectory of $g(x_i(t), x_i^j(t))$ in Fig. 72.2. As a result, the upper bound $\delta_i(\tilde{k}_t^i)$ for $g(x_i(t), x_i^j(t))$ is at our disposal, which, due to Assumption 4(a), leads to a bound for $\|x_i(t) - x_i^j(t)\|$. This upper bound turns out to be instrumental in the proof of Theorem 72.1.

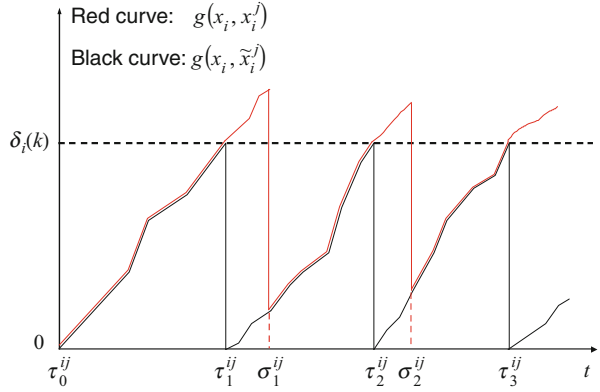
The case where communication delays are not negligible is now discussed, and the communication event policy is modified accordingly, leading to a convergence result similar to Theorem 72.1. The key to achieving this is to find another bound for $g(x_i(t), x_i^j(t))$ and $\|x_i(t) - x_i^j(t)\|$ without resorting to a negligible delay assumption. The first step is to observe that, as a consequence of non-negligible communication delays, after a communication event originates from nodes i to j at τ_n^{ij} , node j cannot set x_i^j to $x_i(\tau_n^{ij})$ immediately. Therefore, the error function $g(x_i(t), x_i^j(t))$ continues to grow until the message containing $x_i(\tau_n^{ij})$ is received by node j . This point is illustrated in Fig. 72.3 where the red curve is an example of the trajectory of $g(x_i(t), x_i^j(t))$ and σ_n^{ij} denotes the time when a message sent by node i at τ_n^{ij} is received at node j . Obviously, if messages sent by a node can take infinitely long to reach other nodes, there can be no cooperation among these nodes, and a distributed algorithm may never reach an optimal solution. If this happens, then the underlying communication network used to control the system is likely to be unstable, a case that must be excluded. In particular, if $\sigma_n^{ij} - \tau_n^{ij}$ for some $n = 1, 2, \dots$ in Fig. 72.3 is unbounded, then $g(x_i(t), x_i^j(t))$ may be unbounded, possibly leading to an infinitely large estimation error (for simplicity, the error bound $\delta_i(k)$ is treated as constant in all figures in this section). Thus, as a first step, the following assumption is added:

Assumption 6. There exists a nonnegative integer D such that if a message is sent before t_{k-D} from any node i to any other node j , it will be received before t_k .

In other words, it is assumed that at most D state update events can occur between a node sending a message and all destination nodes receiving this message.

An additional complication caused by non-negligible delays is in the triggering of communication events. Specifically, if the policy in (72.7) is still used, then between τ_n^{ij} and σ_n^{ij} the value of $g(x_i(t), x_i^j(t))$ generally remains above δ_i (as shown in Fig. 72.3). As a result, node i will continuously trigger communication events until σ_n^{ij} . To suppress this redundancy over the interval $[\tau_n^{ij}, \sigma_n^{ij}]$, it is necessary to modify (72.7). Doing so requires carefully differentiating between the way in which node j estimates the state of node i in the presence of communication delays as opposed to no delays (as assumed in the previous section). Thus, such a general-purpose estimate is expressed as follows:

Fig. 72.3 Trajectories of error functions with communication delays. In this case, every message sent by node i is received before a new communication event is triggered at node i



$$x_i^j(t) = \begin{cases} h_i^j(t; \tau_{n-1}^{ij}) & \rho_{n-1} < t < \rho_n, \quad n = 1, 2, \dots \\ x_n & t = \rho_n, \quad n = 1, 2, \dots \end{cases} \quad (72.14)$$

where $h_i^j(\cdot)$ denotes the specific functional form of the estimator used by node j to estimate the state of node i and τ_n^{ij} , $n = 1, 2, \dots$, is the time when the n th communication event from i to j is generated and viewed as a parameter of the estimator. The estimate is defined over an interval (ρ_{n-1}, ρ_n) , and ρ_n is the n th time when the estimate is reset to a given value x_n . This provides a formal representation of the estimation process in our optimization framework: At instants ρ_n , $n = 1, 2, \dots$, node j resets its estimate of node i 's state to some value x_n and then repeats the estimation process with this new initial condition over the next interval (ρ_n, ρ_{n+1}) . In the case of negligible communication delays (considered in the previous section),

$$\rho_n = \tau_n^{ij}, \quad x_n = x_i(\tau_n^{ij}) \quad (72.15)$$

that is, the n th reset time is the instant when a communication event is generated from nodes i to j , and the reset value is the corresponding actual state of i at that time. However, in the presence of communication delays, one must set

$$\rho_n = \sigma_n^{ij}, \quad x_n = h_i^j(\sigma_n^{ij}; \tau_n^{ij}) \quad (72.16)$$

where $\sigma_n^{ij} > \tau_n^{ij}$ is the time the n th message sent from node i is received by node j , and the reset value depends on the estimation method used by node j .

In view of this discussion, the new communication event policy is defined as

$$\tau_n^{ij} = \inf \left\{ t : g(x_i(t), h_i^j(t; \tau_{n-1}^{ij})) \geq \delta_i(\tilde{k}_t^i), \quad t > \tau_{n-1}^{ij} \right\} \quad (72.17)$$

with the initial condition $\tau_0^{ij} = 0$. Next, a new variable $\tilde{x}_i^j(t)$ is introduced, defined as

$$\tilde{x}_i^j(t) = \begin{cases} h_i^j(t; \tau_{n-1}^{ij}) & \tau_{n-1}^{ij} < t < \tau_n^{ij}, \quad n = 1, 2, \dots \\ x_i(\tau_n^{ij}) & t = \tau_n^{ij}, \quad n = 1, 2, \dots \end{cases} \quad (72.18)$$

with an initial condition $\tilde{x}_i^j(0) = x_i(0)$. In other words, $\tilde{x}_i^j(t)$ is the estimate of node i 's state used by node j as if there were no communication delays, consistent with (72.15). This variable is maintained by node i . If the estimation process is designed to have the property that $h_i^j(\tau_n^{ij}; \tau_n^{ij}) = x_i(\tau_n^{ij})$, (72.18) can be further simplified to

$$\tilde{x}_i^j(t) = h_i^j(t; \tau_{n-1}^{ij}), \quad \tau_{n-1}^{ij} \leq t < \tau_n^{ij}, \quad n = 1, 2, \dots$$

Figure 72.3 illustrates the difference between a trajectory of $g(x_i(t), \tilde{x}_i^j(t))$ (black curve) and a trajectory of $g(x_i(t), x_i^j(t))$ (red curve) under (72.17). Notice that if the communication delay is zero, then $g(x_i(t), \tilde{x}_i^j(t))$ and $g(x_i(t), x_i^j(t))$ will obviously follow the exact same trajectory. There is an additional reason for introducing $\tilde{x}_i^j(t)$ at node i : If communication delays are not negligible and are typically unknown, then node i does not know the exact value of $x_i^j(t)$ and hence cannot execute (72.7).

Regarding the possible forms that $h_i^j(t; \tau_n^{ij})$ can take, two cases are discussed. First, if node j uses a static state estimation method, that is, $h_i^j(t; \tau_n^{ij}) = x_i(\tau_n^{ij})$, it simply sets its estimate of i to the value contained in the received message:

$$x_i^j(\sigma_n^{ij}) = x_i(\tau_n^{ij}) \quad (72.19)$$

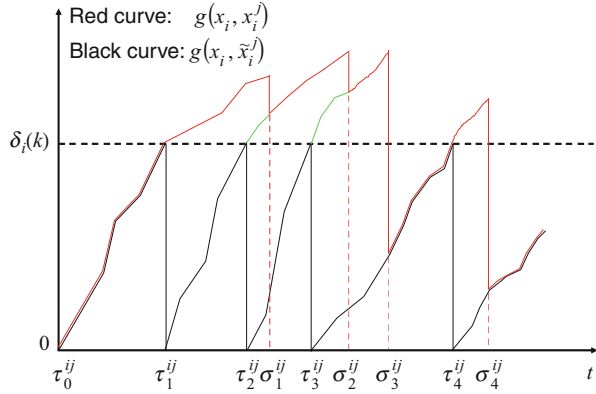
In this case, $\tilde{x}_i^j(t) = x_i(\tau_n^{ij})$ for all $t \in [\tau_n^{ij}, \tau_{n+1}^{ij})$, in accordance with (72.18). On the other hand, if node j uses a dynamic state estimation method, the value of $x_i^j(\sigma_n^{ij})$ depends on τ_n^{ij} as well. For example, if node j uses (72.4), then

$$x_i^j(\sigma_n^{ij}) = x_i(\tau_n^{ij}) + (\sigma_n^{ij} - \tau_n^{ij}) \cdot D_i \quad (72.20)$$

where $D_i = \alpha_i \cdot d_{i, \tau_n^{ij}} / \Delta_i$. Note that in this case evaluating the state estimate in (72.20) requires knowledge of τ_n^{ij} , that is, the message sent by node i must be time stamped (otherwise, in the presence of random delays, node j cannot infer the value of τ_n^{ij} when it receives a message at σ_n^{ij}). This, in turn, requires a clock synchronization mechanism across nodes (which is often not simple to implement).

Finally, the new communication event policy (72.17) and Assumption 6 provide bounds for $g(x_i(t), x_i^j(t))$ and $\|x_i(t) - x_i^j(t)\|$. For simplicity, first consider the case where every message from a node reaches its destination before the next communication event is triggered, that is, $\sigma_n^{ij} < \tau_{n+1}^{ij}$ for all $n = 1, 2, \dots$, as shown in Fig. 72.3. At each σ_n^{ij} , the state estimate $x_i^j(\sigma_n^{ij})$ is synchronized with $\tilde{x}_i^j(\sigma_n^{ij})$ defined in (72.18), and it follows that $g(x_i(\sigma_n^{ij}), x_i^j(\sigma_n^{ij})) \leq \delta_i(\tilde{k}_{\sigma_n^{ij}}^i)$. Due to Assumption 6, at most D state update events occur between τ_n^{ij} and σ_n^{ij} for all

Fig. 72.4 Trajectories of error functions with communication delays. In this case, additional communication events occur before a message is received



$n = 1, 2, \dots$, which guarantees that $g(x_i(t), x_i^j(t))$ will not exceed $\delta_i(k)$, for some k , augmented by the estimation error accumulated over these D state update steps.

If, on the other hand, $g(x_i(t), \tilde{x}_i^j(t))$ reaches $\delta_i(k_i^j)$ during $[\tau_n^{ij}, \sigma_n^{ij}]$, then a new communication message will be sent out by node i before the previous one is received. For example, in Fig. 72.4, $\tau_2^{ij} < \sigma_1^{ij}$ and $\tau_3^{ij} < \sigma_2^{ij}$. Although the trajectory of $g(x_i(t), x_i^j(t))$ becomes more complicated in this case, $g(x_i(t), x_i^j(t))$ (the red curve) still has an upper bound, which is given within the proof of Theorem 3, in Zhong and Cassandras (2010).

In light of this discussion, if an error function is chosen such that $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|/K_4$, it is obvious that $\|x_i(t) - x_i^j(t)\|$ is bounded. Thus, if any error function is chosen such that $\|x_i(t) - x_i^j(t)\| \leq K_4 \cdot g(x_i(t), x_i^j(t))$ (according to Assumption 4(a), $\|x_i(t) - x_i^j(t)\|$ is also bounded. In Theorem 72.3, the static estimation case (72.19) is considered, which is the least costly in terms of estimation complexity. Moreover, static estimation, as already pointed out, does not require a synchronization mechanism which (72.20), for example, does. It is also important to point out that in Theorem 72.3, Assumption 4(b) is no longer used; instead, Assumption 6 allows the derivation of the bounds required to establish the desired convergence result.

Theorem 72.3. *Under Assumptions 1–4(a) and 6, the communication event policy (72.17), the state update scheme (72.11), and the static estimation method (72.19), if the error threshold $\delta_i(k)$ controlling communication events is set by (72.8) and (72.9), then there exist positive constants α and K_δ such that $\lim_{k \rightarrow \infty} \nabla H(s(k)) = 0$.*

In addition, if both the state update noise introduced in Sect. 72.3.1 and non-negligible communication delays are present in the optimization process, as long as Assumptions 5 and 6 are still satisfied, a convergence result (omitted here) similar to Theorem 72.2 can also be derived.

72.4 Asynchronous Distributed Coverage Control

In this section, the proposed distributed optimization framework with event-driven communication is applied to the class of coverage control problems mentioned in Sect. 72.1. Specifically, the problem of deploying a UAV team is considered so that the UAVs are located in a way that maximizes the probability of detecting events occurring in a given two-dimensional mission space. The goal is to show how an asynchronous algorithm may significantly reduce the energy expended on communication among nodes with no adverse effect on performance. The convergence of the objective function is also illustrated without and with communication delays, and the effect of K_δ on the convergence speed and the communication cost is examined.

72.4.1 Coverage Control Problem Formulation

A brief review of the coverage control problem is first given, following the formulation in Cassandras and Li (2005). An event density function $R(x)$ over the mission space $\Omega \subset \mathbb{R}^2$ is defined, which captures the frequency of random event occurrences at some point $x \in \Omega$. $R(x)$ satisfies $R(x) \geq 0$ for all $x \in \Omega$ and $\int_\Omega R(x) dx < \infty$. It is assumed that when an event takes place, it will emit some signal which may be observed by some UAVs equipped with sensing device. The cooperative system consists of N UAV nodes deployed into Ω to detect the random events. Their positions are captured by a $2N$ -dimensional vector $\mathbf{s} = (s_1, \dots, s_N)$.

The probability that node i detects an event occurring at $x \in \Omega$, denoted by $p_i(x, s_i)$, is a monotonically decreasing differentiable function of $\|x - s_i\|$, the Euclidean distance between the event and the UAV node. Since multiple nodes are deployed to cover the mission space, the joint probability that an event occurring at x is detected (assuming independent sensor detection processes), denoted by $P(x, \mathbf{s})$, is given by

$$P(x, \mathbf{s}) = 1 - \prod_{i=1}^N [1 - p_i(x, s_i)]$$

and the optimization problem of interest is

$$\max_{\mathbf{s}} H(\mathbf{s}) = \max_{\mathbf{s}} \int_{\Omega} R(x) P(x, \mathbf{s}) dx$$

in which the locations of the UAV nodes are used as decision variables to maximize the probability of event detection in Ω . As discussed in Zhong and Cassandras (2008b), this objective function aims at maximizing the joint event detection probability without considering the issue of maintaining a balance between a region

within Ω which may be extremely well covered and others which may not. Thus, a more general form of the objective function is

$$H(\mathbf{s}) = \int_{\Omega} R(x) M(P(x, \mathbf{s})) dx \quad (72.21)$$

where $M(P) : [0, 1] \rightarrow \mathbb{R}$ is a (possibly piecewise) differentiable concave nondecreasing function of the joint detection probability P . Clearly, $M(P)$ may be selected so that the same amount of marginal gain in $P(x, \mathbf{s})$ is assigned a higher reward at a lower value of P . A special case is $M(P) = P$ where coverage balance is not taken into account.

A *synchronous* gradient-based solution was obtained in Cassandra and Li (2005) in which the next waypoint on the i th UAV node's trajectory is determined through

$$s_i(k+1) = s_i(k) + \alpha \frac{\partial H(\mathbf{s})}{\partial s_i}, \quad k = 0, 1, \dots \quad (72.22)$$

The gradient is given by

$$\frac{\partial H(\mathbf{s})}{\partial s_i} = \int_{\Omega_i} R(x) \prod_{k \in \mathcal{B}_i} [1 - p_k(x, s_k)] \frac{dp_i(x, s_i)}{dd_i(x)} \frac{s_i - x}{d_i(x)} dx \quad (72.23)$$

where $d_i(x) \equiv \|x - s_i\|$ and $\Omega_i = \{x : d_i(x) \leq D\}$ is the node i 's region of coverage where D denotes the sensing radius of node i (i.e., $p_i(x, s_i) = 0$ for $d_i(x) > D$). In addition, $\mathcal{B}_i = \{k : \|s_i - s_k\| < 2D, k = 1, \dots, N, k \neq i\}$ is the set of neighbor nodes of i .

The state update rule (72.22) allows a fully distributed implementation based on node i 's local information and the state of its neighbors (instead of the state of all nodes). This eliminates the communication burden of transferring information to and from a central controller and the vulnerability of the whole system which would be entirely dependent on this controller. However, (72.23) shows that node i needs the exact locations of all nodes in \mathcal{B}_i in order to carry out (72.22) in a state update. As already mentioned, the communication involved to ensure such state synchronization has a high energy cost which is often unnecessary because the locations of neighboring nodes may be accurately estimated due to minor changes in their locations or update directions (see (72.3) and (72.4)).

Next, the asynchronous method developed above is applied to this problem, and the results with the synchronous approach are compared in terms of communication cost, performance, and convergence behavior. The asynchronous method is also applied to a nonsmooth version of the coverage control problem when the mission space contains obstacles (see also Zhong and Cassandra 2011).

72.4.2 Asynchronous Versus Synchronous Optimal Coverage Control

Numerical results are presented based on simulations of the coverage control setting using an interactive Java-based simulation environment (The interactive simulation environment used in all results shown in this section (along with instructions) is located at <http://codescolor.bu.edu/simulators.html>). We compare three versions of the optimal coverage control solution:

1. Synchronous iterations where all nodes perform state updates using state information from all other nodes
2. Asynchronous iterations performed by node i with fixed error threshold $\delta_i(k)$, that is, $\delta_i(k) = \delta_i$ for all k , where δ_i is a positive constant
3. Asynchronous iterations performed by node i using (72.8) and (72.9)

The coverage control problem considered here involves four UAVs deployed into a rectangular mission space (50 units by 60 units) from the lower left corner. All UAVs update their states at approximately the same frequency (one update in every ten simulation time steps augmented by some uniformly distributed noise) using the same step size in all three schemes. They all have identical sensing capabilities and $p_i(x, s_i) = e^{-0.08\|x-s_i\|}$ for all i , which means that, if deployed individually, a UAV can only effectively cover a relatively small portion of the mission space ($p_i(x, s_i) < 0.50$ when $\|x - s_i\| > 8.67$). For the asynchronous versions, (72.3) is used as a simple static state estimation scheme, and the error function is $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|$.

In Figs. 72.5 and 72.6, these three algorithms are compared under both the negligible and non-negligible communication delay assumptions. When communication costs are compared in Fig. 72.5, every time a node broadcasts its state information, the total number of communications is increased by one. By looking at these figures, it is clear that the asynchronous method can substantially reduce the communication cost (hence, the energy consumption at all UAVs) while performance convergence is virtually indistinguishable from that of the synchronous method. The asynchronous algorithm with fixed $\delta_i(k)$ has the added advantage that it usually stops incurring communication cost earlier than the other two methods. However, it does not guarantee convergence to stationary points. In Figs. 72.5b and 72.6b, results are provided for a case with a fairly large communication delay (50 simulation time steps for every transmitted message). Comparing Fig. 72.6b to a, it is clear that introducing delays does not affect convergence but it does slow it down; in Fig. 72.6b, the optimal value of the objective function is attained approximately Assumption 5 time units later than the time seen in Fig. 72.6a.

Figure 72.7 shows the UAV trajectories for these three methods under the negligible communication delay assumption. Methods 1 and 3 converge to the same node configuration which is indicated by the squares, while method 2 converges to a configuration close to it (it is not possible to mark the configuration method 2 converges to due to the limited resolution of the figure).

Fig. 72.5 Communication cost comparison of three distributed optimization algorithms under different communication delay assumptions. (a) Negligible communication delay. (b) Non-negligible communication delay

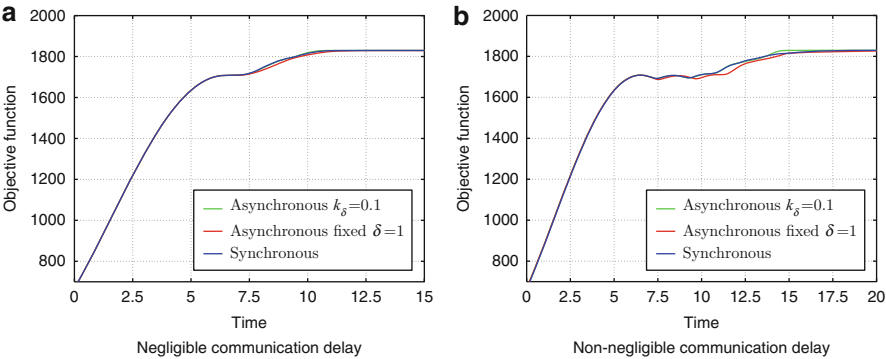
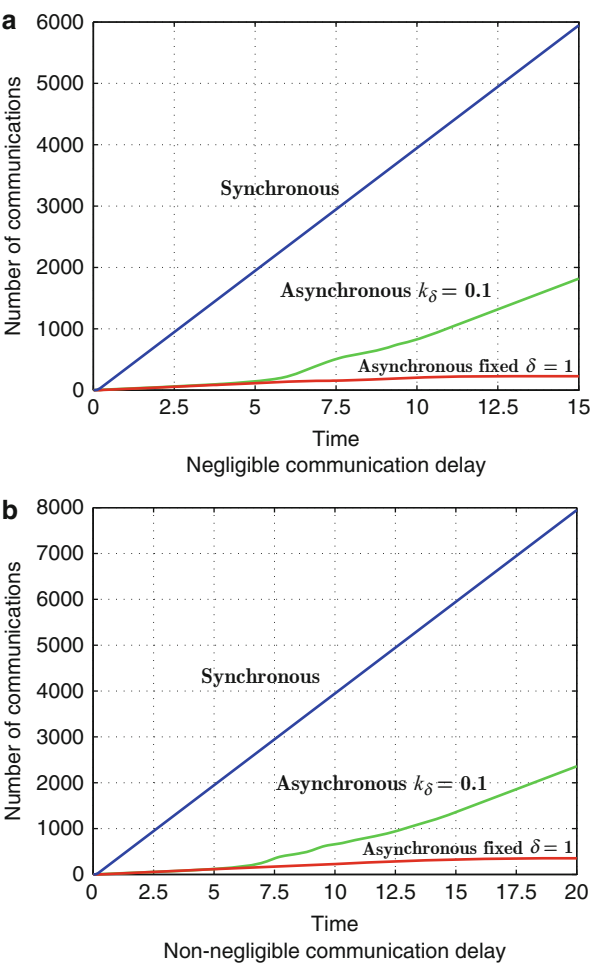


Fig. 72.6 Objective function trajectories of three distributed optimization algorithms under different communication delay assumptions. (a) Negligible communication delay. (b) Non-negligible communication delay

Fig. 72.7 Node trajectory comparison of three distributed optimization algorithms in a coverage control mission. See Fig. 72.6 for legend

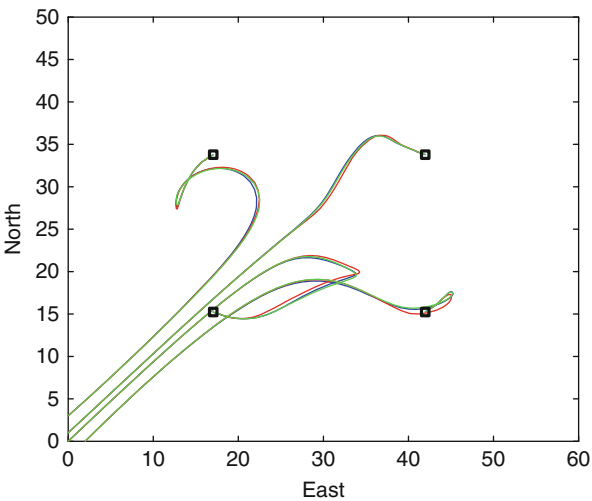
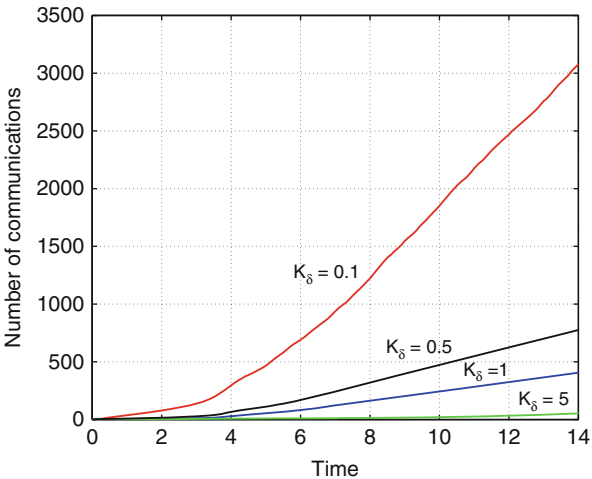


Fig. 72.8 Communication cost comparison of asynchronous distributed algorithm with different K_δ



In Figs. 72.8 and 72.9, under the negligible communication delay assumption, the performance of the asynchronous method is compared under different values of the constant K_δ in (72.8) with the same coverage control setting as before. One can see that a larger K_δ leads to fewer communication events. But one can also notice in Fig. 72.9 that when $K_\delta = 5$, the objective function curve exhibits considerable oscillations before it converges. This suggests that K_δ is set too high and some “necessary” communications between nodes have been omitted at the expense of effective cooperation. In other words, when K_δ is increased, although convergence to a local optimum may still be guaranteed if (72.12) is satisfied, so is the risk of slower convergence.

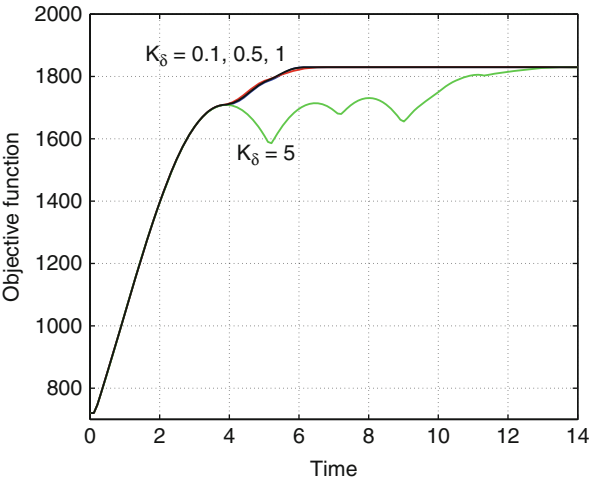


Fig. 72.9 Objective function trajectories of asynchronous distributed algorithm with different K_δ

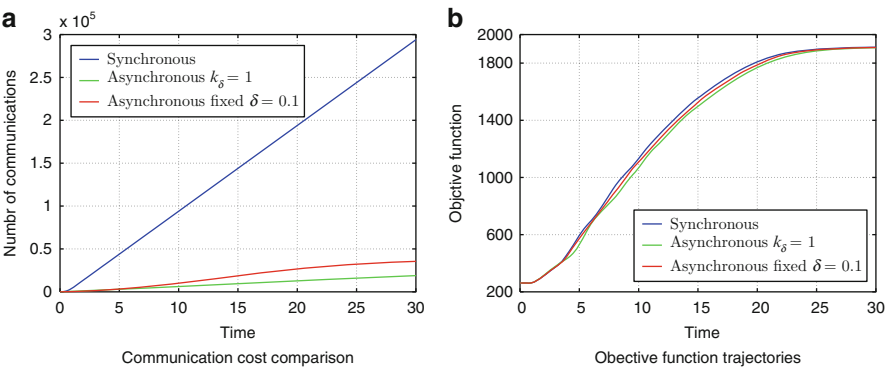


Fig. 72.10 Coverage control simulation result of a large network with 100 nodes. (a) Communication cost comparison. (b) Objective function trajectories

In Fig. 72.10, results are shown from a larger coverage control problem with 100 UAVs with no obstacle or communication delay and reduced sensing range. Similar communication savings are observed as in the smaller problem of Fig. 72.5 with only four UAVs.

Next, the asynchronous algorithm is applied to a problem with polygonal obstacles in the region to be covered and employs (72.4) as the state estimation method. Five UAVs are deployed into a mission space cluttered with obstacles as shown in Fig. 72.11. In this case, a nonsmooth optimization problem arises with gradients that are more complicated than (72.23) in order to account for discontinuities in the sensor detection probability functions $p_i(x, s_i)$; details on their derivation

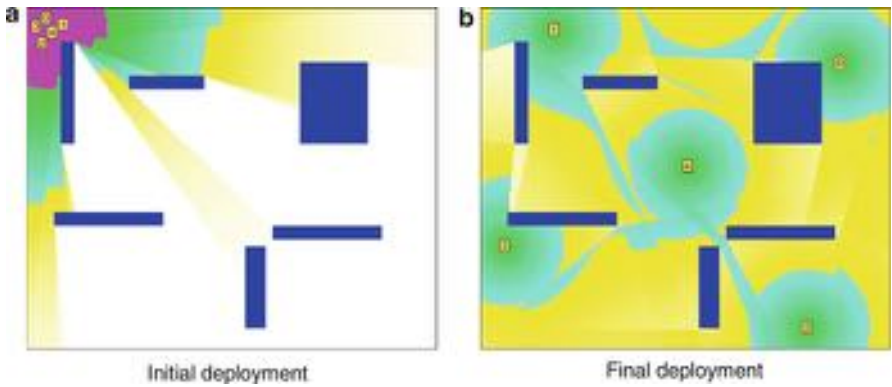
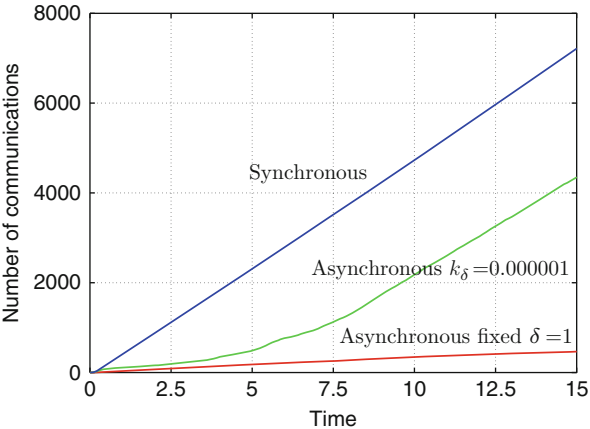


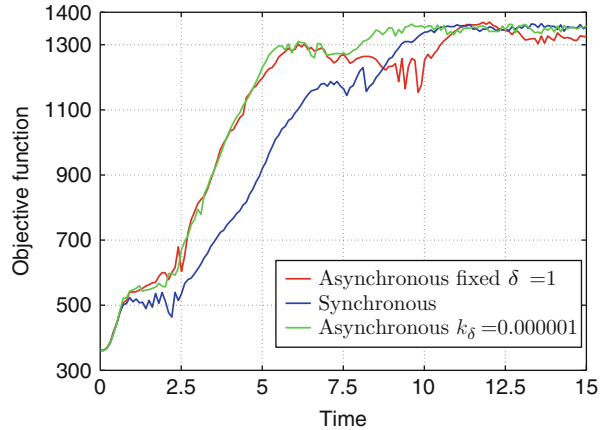
Fig. 72.11 Initial and final deployment of an optimal coverage mission in an environment with polygonal obstacles. The *blue rectangles* represent obstacles. Areas covered with cyan color have joint detection probability $P(x, s) \geq 0.97$. Green areas have $0.50 \leq P(x, s) < 0.97$. Yellow and white areas have $0 \leq P(x, s) < 0.50$. If the image is viewed in *black and white*, a lighter color indicates lower event detection probability in that area. (a) Initial deployment. (b) Final deployment

Fig. 72.12 Communication cost comparison of three distributed optimization algorithms in a nonsmooth problem



are provided in Zhong and Cassandras (2011). Convergence under (72.22) cannot always be guaranteed, and there are multiple local optima which nodes can oscillate around. In order to achieve a more balanced coverage, the objective function in (72.21) is used. In Fig. 72.12, one can once again see the cost-cutting benefits of using asynchronous methods. Figure 72.13 shows that all three methods eventually achieve the same approximate level of coverage.

Fig. 72.13 Objective function trajectories of three distributed optimization algorithms in a nonsmooth problem



72.5 Conclusion

The issue addressed in this chapter is the extent to which communication among autonomous UAVs could be reduced in a cooperative optimization setting. This is crucial when the cooperating UAVs have limited energy resources. In addition, limiting communication reduces the possibility of jamming or other sources of information corruption that can damage the cooperative process. The main result is showing that frequent communication among cooperating UAVs (nodes in a network) is not necessary in order to guarantee convergence of a process seeking to optimize a common objective. Specifically, the scheme that has been developed limits communication events to “last resort only” when some state estimation error function at a UAV exceeds a threshold. Under certain technical conditions, the convergence of a gradient-based asynchronous distributed algorithm employing this communication policy is still provably guaranteed. In addition, the range of the two crucial parameters on which such convergence depends has been quantified. This important property applies to both cases where communication delays are negligible and where they are not. In the latter case, as expected, simulation results show convergence to be generally slower. Further, the convergence property still applies when noise is explicitly included in the UAV state update process, provided that some additional conditions hold which appropriately bound this noise term.

This approach has been applied to a coverage control problem common in the deployment of UAV teams and confirmed through numerical examples that limited asynchronous (event driven) communication results in substantial energy savings which can prolong the life of such a team with no adverse effect on the optimization objective. Thus, from a practical standpoint, the proposed method can significantly benefit teams of autonomous UAVs by reducing their use of energy with no compromise in performance. However, there is no guarantee that an event-driven

communication approach always provides energy savings relative to time-driven mechanisms. What this approach offers is the flexibility to exploit intervals over which there is no need for a UAV to unnecessarily communicate minimal errors in its state trajectory.

Acknowledgments This work is supported in part by NSF under Grants DMI-0330171 and EFRI-0735974; by AFOSR under grants FA9550-04-1-0133, FA9550-04-1-0208, FA9550-07-1-0361, FA9550-09-1-0095; by DOE under grant DE-FG52-06NA27490; and by ONR under grant N00014-09-1-1051.

References

- D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods* (Athena Scientific, Belmont, MA, 1997)
- C.G. Cassandras, W. Li, Sensor networks and cooperative control. *Eur. J. Control* **11**(4–5), 436–463 (2005)
- J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.* **20**(2), 243–255 (2004)
- M. DeGroot, Reaching a consensus. *J. Am. Stat. Assoc.* **69**(345), 118–121 (1974)
- A. Ganguli, J. Cortés, F. Bullo, Distributed deployment of asynchronous guards in art galleries, in *Proceedings of the American Control Conference*, Minneapolis, 2006, pp. 1416–1421
- P. Hokayem, D. Stipanovic, M. Spong, Dynamic coverage control with limited communication, in *Proceedings of the American Control Conference* (IEEE, New York, 2007), pp. 4878–4883
- I. Hussein, D. Stipanovic, Effective coverage control using dynamic sensor networks with flocking and guaranteed collision avoidance, in *Proceedings of the American Control Conference* (IEEE, New York, 2007), pp. 3420–3425
- A. Jadbabaie, J. Lin, S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* **48**(6), 988–1001 (2003)
- J. Lawton, B. Young, R. Beard, A decentralized approach to elementary formation maneuvers, in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation* (Robotics and Automation Society/IEEE Service Center, Piscataway, 2000), pp. 2728–2733
- S. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, in *Proceedings of IEEE INFOCOM* (IEEE, Piscataway, 2001), pp. 1380–1387
- L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, Active sensing for robotics—a survey, in *Proceedings of the 5th International Conference on Numerical Methods and Applications*, Borovets, Bulgaria, 2002, pp. 316–324
- L. Moreau, Stability of multi-agent systems with time-dependent communication links. *IEEE Trans. Autom. Control* **50**(2), 169–182 (2005)
- A. Nedić, A. Ozdaglar, Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control* **54**(1), 48–61 (2009)
- P. Ögren, E. Fiorelli, N.E. Leonard, Cooperative Control of mobile sensor networks: adaptive gradient climbing in a distributed environment. *IEEE Trans. Autom. Control* **49**(8), 1292–1302 (2004)
- T. Shima, S. Rasmussen, P. Chandler, UAV team decision and control using efficient collaborative estimation. *ASME J. Dyn. Syst. Meas. Control* **129**(5), 609–619 (2007)
- V. Shnayder, M. Hempstead, B.R. Chen, G.W. Allen, M. Welsh, Simulating the power consumption of large-scale sensor network applications, in *Proceedings of the International Conference on Embedded Networked Sensor Systems* (ACM, New York, 2004), pp. 188–200
- T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **75**(6), 1226–1229 (1995)

- P. Wan, M.D. Lemmon, Event triggered distributed optimization in sensor networks, *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks* (IEEE Computer Society, Washington, DC, 2009)
- M. Zhong, C.G. Cassandras, Asynchronous distributed optimization with minimal communication, in *Proceedings of 47th IEEE Conference on Decision and Control* (IEEE, Cancun, 2008a), pp. 363–368
- M. Zhong, C.G. Cassandras, Distributed coverage control in sensor networks environments with polygonal obstacles, in *Proceedings of 17th IFAC World Congress, July 6–11, 2008, Seoul* (International Federation of Automatic Control, Seoul, 2008b), pp. 4162–4167
- M. Zhong, C.G. Cassandras, Asynchronous distributed optimization with event-driven communication. *IEEE Trans. Autom. Control* **55**(12), 2735–2750 (2010)
- M. Zhong, C.G. Cassandras, Distributed coverage control and data collection with mobile sensor networks. *IEEE Trans. Autom. Control* **56**(10), 2445–2455 (2011)
- Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in *Proceedings of IEEE INFOCOM* (IEEE, New York, 2003), pp. 1293–1303

Shankarachary Ragi and Edwin K. P. Chong

Contents

73.1 Introduction 1776

73.2 Related Work 1777

73.3 Partially Observable Markov Decision Process 1778

 73.3.1 Definition 1778

 73.3.2 Optimization Objective..... 1780

 73.3.3 Optimal Policy 1780

73.4 Formulation of the UAV Guidance Problem 1781

 73.4.1 Problem Specification 1781

 73.4.2 POMDP Formulation..... 1782

 73.4.3 NBO Approximation Method 1783

 73.4.4 UAV Kinematics 1784

 73.4.5 Empirical Study of NBO 1785

73.5 Model Enhancements and Additional Features 1786

 73.5.1 Wind Compensation 1787

 73.5.2 Collision Avoidance 1789

 73.5.3 Evading Threats 1792

 73.5.4 Tracking Evasive Targets..... 1797

 73.5.5 Track Swap Avoidance 1801

73.6 Conclusion 1805

References 1806

Abstract

The goal here is to design a path-planning algorithm to guide unmanned aerial vehicles (UAVs) for tracking multiple ground targets based on the theory of *partially observable Markov decision processes* (POMDPs). This study shows how to exploit the generality and flexibility of the POMDP framework by

incorporating a variety of features of interest naturally into the framework, which is accomplished by plugging in the appropriate models. Specifically, this study shows how to incorporate the following features by appropriately formulating the POMDP action space, state-transition law, and objective function: (1) control UAVs with both forward acceleration and bank angle subject to constraints, (2) account for the effect of wind disturbance on UAVs, (3) avoid collisions between UAVs and obstacles and among UAVs, (4) track targets while evading threats, (5) track evasive targets, and (6) mitigate track swaps.

73.1 Introduction

This study describes a dynamic path-planning algorithm that guides autonomous UAVs for tracking multiple ground targets based on the theory of *partially observable Markov decision processes* (POMDPs) (Chong et al. 2009; Miller et al. 2009). The algorithm collects measurements from the sensors (mounted on the UAVs), constructs the tracks, and computes the control commands for the UAVs.

In practice, POMDP problems are intractable to solve exactly. There are several approximation methods in the literature to solve a POMDP: heuristic ECTG (Kreucher et al. 2004), parametric approximation (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998), policy rollout (Bertsekas and Castanon 1999), hindsight optimization (Chong et al. 2000; Wu et al. 2002), and foresight optimization (Bertsekas 2007). This study uses an approximation method called the *nominal belief-state optimization* (NBO) (Miller et al. 2009) (explained later), which is computationally efficient. This study shows how to incorporate the following features of interest naturally into the POMDP framework: control UAVs with both forward acceleration and bank angle subject to constraints, account for wind disturbance on UAVs, avoid collisions between UAVs and obstacles and among UAVs, guide UAVs to track targets while evading threats, guide UAVs for tracking evasive targets, and guide UAVs for tracking targets while mitigating track swaps. These issues are addressed as explained below:

- Variable-speed UAVs are considered for the purpose of this study, where the speed and heading direction of a UAV are varied by appropriately controlling the forward acceleration and bank angle. These control variables are subject to certain constraints. These control variables are incorporated into the framework, which is accomplished by appropriately formulating the POMDP action space and state-transition law.
- Wind speeds are often comparable to UAV cruise speeds. As a result, the trajectory of a UAV gets deviated from its desired course (due to wind) leading to tracking performance degradation. The study addresses this problem by appropriately formulating the POMDP state-transition law. More precisely, the speed and direction of wind are incorporated into the UAV kinematic model.
- In a scenario where UAVs are tracking targets in the presence of obstacles and the UAV flying altitude is less than the height of these obstacles, a collision may happen between a UAV and an obstacle. This study designs the algorithm

in such a way that the UAVs track the targets while avoiding collisions with the obstacles. In a scenario where all UAVs fly at a same altitude, a collision may happen between a UAV and another UAV. The goal here is to design a path-planning algorithm while avoiding collisions among UAVs. A penalty metric is incorporated into the objective function used for optimizing the control commands to prevent collisions among UAVs.

- In this present study, a threat is a ground vehicle that pursues UAVs that are tracking it. The goal here is to design a method—in the context of the POMDP framework—to track targets while evading threats. A threat motion model is defined to represent the dynamics of a threat. A penalty metric is incorporated into the objective function to guide the UAVs to evade threats while tracking targets.
- In this present study, an evasive target is a ground vehicle that actively moves away from UAVs that are tracking it. The goal here is to design a method—in the context of the POMDP framework—to track evasive targets. An evasive target motion model is defined to represent the target dynamics. This motion model is incorporated into the NBO method.
- A *track swap* is an undesirable event that causes the swapping of target identities. The likelihood of a track swap depends on the location of the UAVs and the state of the tracker. Therefore, track swaps are mitigated by controlling the UAVs appropriately. This is accomplished by incorporating a metric—that represents the risk of a track swap – into the objective function.

In summary, this study shows how to incorporate a variety of features of interest naturally into the POMDP framework by simply plugging in the appropriate models. The organization of this chapter is as follows: Sect. 73.2 gives a brief overview of UAV guidance algorithms present in the literature and contrasts these methods with the method considered in this study; Sect. 73.3 presents a general description of the POMDP framework; Sect. 73.4 poses the UAV guidance problem as a POMDP and presents an approximation method to solve the POMDP problem; Sect. 73.5 shows how to incorporate a variety of features of interest naturally into the POMDP framework; and Sect. 73.6 summarizes this study.

73.2 Related Work

The literature is replete with motion planning algorithms for autonomous vehicles. A general survey of path-planning algorithms can be found in Dadkhah and Mettler (2012), Goerzen et al. (2010), and LaValle (2006). What makes this path-planning method for UAVs (presented in this chapter) worth studying is the combination of the following features naturally within the POMDP framework:

1. This method involves real-time calculation of guidance commands in response to continual feedback information.
2. This method has a “look-ahead” property, in that the guidance control at each time step takes into account the impact of the control over a time horizon into the future.

3. This method incorporates explicit dynamic constraints on the UAV motion.
4. The cost criterion is based on tracking error.

Many researchers have developed path-planning algorithms in the past while incorporating a few of these features. For example, the path-planning algorithms presented in Geyer (2008), Berger et al. (2009), He et al. (2010a), Tisdale et al. (2007), Park et al. (2009), He et al. (2010b), Kim et al. (2008), Saunders et al. (2005), and Cassandra and Li (2003) incorporated “look-ahead” and can be implemented in real time. However, only Tisdale et al. (2007) and Saunders et al. (2005) incorporated the dynamic constraints on UAV motion, and only He et al. (2010b) incorporated the tracking error as a cost criterion. This POMDP method is an integrated approach where all the above features are fused in a single framework—the guidance commands are calculated in real time (with look-ahead) while incorporating dynamic constraints on UAV motion in response to feedback information, where the cost criterion is tracking error.

There are many other methods in the literature that are not designed specifically for UAVs but are designed for autonomous robots in general, e.g., Kaneshige et al. (2004), Leven and Hutchinson (2002), Sud et al. (2008), Naveed et al. (2011), Bruce and Veloso (2002), LaValle (2000), Kavralu et al. (1996), Bry and Roy (2011), Yong and Barth (2006), Ma and Castanon (2006), McIntyre et al. (2005), Zu et al. (2004), Ousingawatt and Campbell (2004), Lu et al. (2011), Cowlagi and Tsotras (2007), Ren et al. (2005), and Li and Cassandra (2006). In the recent past, POMDPs have been used to design path-planning algorithms for UAVs, but they differ from the method in this study in that either the dynamic constraints on UAV motion are not incorporated (Sarunic et al. 2009) or the cost criterion is different (Sarunic et al. 2009; Bai et al. 2011). There are also methods in the recent literature (e.g., Capitán et al. 2011; Adiprawita et al. 2011; Du et al. 2010; Candido and Hutchinson 2010; Kurniawati et al. 2011) where POMDPs are used to develop path-planning algorithms for autonomous robots in general. There are many schemes in the literature for wind compensation (e.g., Jennings et al. 2008; McGee et al. 2005; Ceccarelli et al. 2007; McGee and Hedrick 2006; Lawrance and Sukkariéh 2011; Techy et al. 2010) and collision avoidance (e.g., Redding et al. 2007; Koren and Borenstein 1991; Kavraki et al. 1996; He et al. 2006; Dong et al. 2005; Khatib 1985; Tilove 1990; Krogh and Thorpe 1986; Borenstein and Koren 1989; Park et al. 2008; Nikolos et al. 2003). This study shows how to account for these naturally in the context of the POMDP framework.

73.3 Partially Observable Markov Decision Process

73.3.1 Definition

A *partially observable Markov decision process* (POMDP) is a controlled dynamical process in discrete time. A POMDP can also be interpreted as the controlled version of a hidden Markov reward process. This section provides a general description of the POMDP. The following are the key components of the POMDP:

- A set of states of the system that possibly evolve over time and an initial distribution specifying the random initial state. Let \mathcal{X} represent the set of states of the system and x_k is the state of the system at time k such that $x_k \in \mathcal{X}$.
- A set of actions, which are controllable aspects of the system. Let \mathcal{U} be the set of all possible actions and u_k is an action at time k such that $u_k \in \mathcal{U}$.
- A state-transition law that specifies the next state distribution given the current state and action, i.e., $x_{k+1} \sim p_k(\cdot|x_k, u_k)$, where p_k is a conditional probability distribution function given the current state x_k and the current action u_k .
- A set of observations and an observation law that specifies the distribution over the observation space given an action taken at a state. Let \mathcal{Z} represent the set of all observations and z_k be the observation at time k , which is given by $z_k \sim q_k(\cdot|x_k, u_k)$, where q_k is a conditional probability distribution function given the current state x_k and the current action u_k .
- A cost function that specifies the cost (real number) incurred given an action taken at a state, i.e., $C(x_k, u_k)$.

Figure 73.1a shows a symbolic representation of the POMDP process. The POMDP process begins at time $k = 0$ at a (random) initial state. As the process evolves, the state transitions to a (random) next state from the current state according to the state-transition law given the action. An action taken at a given state incurs a

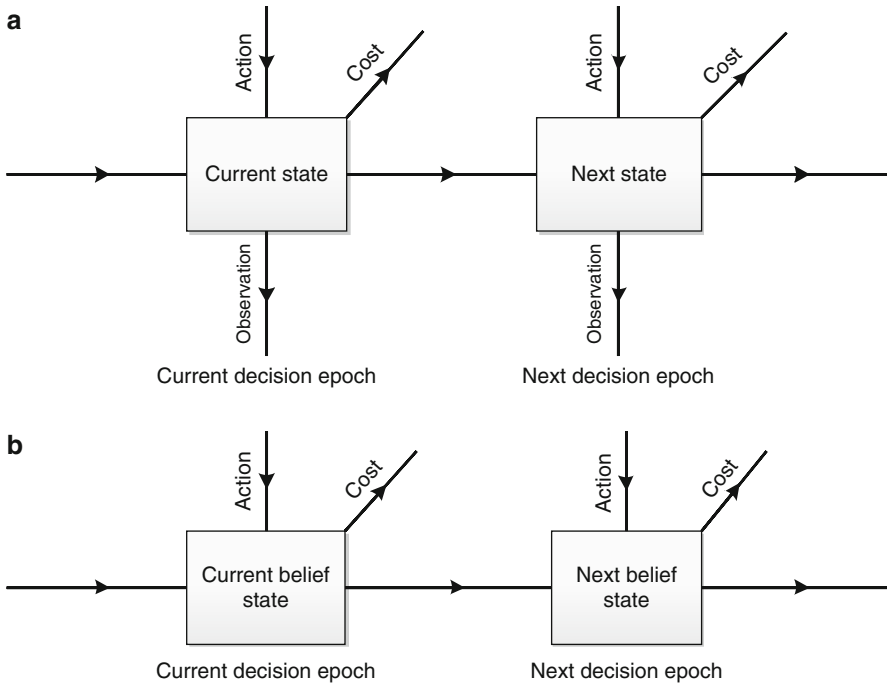


Fig. 73.1 Symbolic representations. (a) POMDP with state space \mathcal{X} and (b) fully observable MDP with state space \mathcal{B}

cost, which is given by the cost function. At every time step, the system generates an observation, which depends on the current state and the current action. At every time step, this observation is used to infer the actual underlying state. However, there will be some uncertainty in the knowledge of the underlying state; this uncertainty is represented by the *belief state*. The belief state is the posterior distribution over the underlying state, which is updated according to the Bayes' rule. Let \mathcal{B} represent the set of distributions over the state space \mathcal{X} . Let $b_k \in \mathcal{B}$ be the belief state at time k . A POMDP can be viewed as a fully observable Markov decision process (MDP) with state space \mathcal{B} as shown in Fig. 73.1b.

73.3.2 Optimization Objective

A POMDP problem boils down to an optimization problem, where the objective is to find actions over a time horizon H such that the expected cumulative cost is minimized. The expected cumulative cost, to be minimized over the action sequence u_0, u_1, \dots, u_{H-1} , is given by

$$J_H = \mathbb{E} \left[\sum_{k=0}^{H-1} C(x_k, u_k) \right]. \quad (73.1)$$

The action chosen at time k should be allowed to depend on the history of all observable quantities till time $k - 1$. If an optimal choice of such actions exists, then there exists an optimal sequence of actions that depend only on the “belief-state feedback” (Chong et al. 2009). Therefore, the objective function J_H can be written in terms of the belief states as follows:

$$J_H = \mathbb{E} \left[\sum_{k=0}^{H-1} c(b_k, u_k) \mid b_0 \right], \quad (73.2)$$

where $c(b_k, u_k) = \int C(x, u_k) b_k(x) dx$ and b_0 is the given initial belief state.

73.3.3 Optimal Policy

Given the optimization problem, the goal is to find, at each time k , the optimal policy $\pi_k^* : \mathcal{B} \rightarrow \mathcal{U}$ such that if the action $u_k = \pi_k^*(b_k)$ is performed at time k , the objective function (73.2) is minimized. According to the Bellman's principle of optimality (Bellman 1957), the optimal objective function value can be written in the following form:

$$J_H^*(b_0) = \min_u \{c(b_0, u) + \mathbb{E} [J_{H-1}^*(b_1) \mid b_0, u]\}, \quad (73.3)$$

where b_1 is the random next belief state, J_{H-1}^* is the optimal cumulative cost over the horizon $H - 1 : k = 1, 2, \dots, H - 1$, and $E[\cdot | b_0, u]$ is the conditional expectation given the current belief state b_0 and an action u taken at time $k = 0$. The Q -value of taking an action u given the current belief state b_0 is as follows:

$$Q_H(b_0, u) = c(b_0, u) + E[J_{H-1}^*(b_1) | b_0, u]. \quad (73.4)$$

The optimal policy (from Bellman's principle) at time $k = 0$ is given by

$$\pi_0^*(b_0) = \arg \min_u Q_H(b_0, u). \quad (73.5)$$

The optimal policy at time k is given by

$$\pi_k^*(b_k) = \arg \min_u Q_{H-k}(b_k, u). \quad (73.6)$$

In practice, the second term in the Q function (73.4) is hard to obtain exactly. There are several methods in the literature to obtain the Q -value approximately, which include heuristic ECTG (Kreucher et al. 2004), parametric approximation (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998), policy rollout (Bertsekas and Castanon 1999), hindsight optimization (Chong et al. 2000; Wu et al. 2002), and foresight optimization (Bertsekas 2007). This study adopts an approximation method called *nominal belief-state optimization* (NBO), which was introduced in Miller et al. (2009) specifically for the purpose of solving the POMDP in the context of the UAV guidance problem.

73.4 Formulation of the UAV Guidance Problem

73.4.1 Problem Specification

This study considers a specific UAV guidance problem, where the goal is to design an algorithm to control a fleet of UAVs for multitarget tracking. The guidance problem is specified as follows:

- *2-D motion*: The targets move in a plane on the ground. A simplified UAV motion model is used, where the altitude of a UAV is constant. The position coordinates of each UAV, i.e., (x, y) , are varied by applying the following UAV controls: forward acceleration and bank angle.
- *Noisy observations*: The UAVs are mounted with sensors that generate the position measurements of the targets. These measurements have random errors that are spatially varying, i.e., the measurement error covariance of a target depends on the locations of sensors (UAVs) and the target.

- *UAV controls*: A UAV is controlled by forward acceleration (controls the speed) and bank angle (controls the heading angle). The values of these control variables lie within certain minimum and maximum limits.
- *Obstacles*: The obstacles are modeled as circular regions in the surveillance region that act as occlusions and also block the path of UAVs.
- *Threats and evasive targets*: A threat is a ground vehicle that pursues the UAVs that are tracking it. An evasive target is a ground vehicle that actively moves away from UAVs to avoid being tracked.
- *Tracking objectives*: The objective is to minimize the mean-squared error between the tracks and the targets. In Sections 73.5.2, 73.5.3, and 73.5.5, the tracking objectives are modified to avoid collisions (between UAVs and obstacles and among UAVs), evade threats, and mitigate track swaps.

73.4.2 POMDP Formulation

This section poses the UAV guidance problem as a *partially observable Markov decision process* (POMDP). The key components of POMDP, in terms of the UAV guidance problem, are as follows:

- *States*. Here, three subsystems are defined as follows: the sensors, the targets, and the tracker. The state at time k is given by $x_k = (s_k, \chi_k, \xi_k, \mathbf{P}_k)$, where s_k represents the sensor state, χ_k represents the target state, and (ξ_k, \mathbf{P}_k) represents the tracker state. The sensor and the target states include the locations and velocities of the UAVs and the targets, respectively. The tracker state is a standard in the Kalman filter (Blackman and Popoli 1999; Bar-Shalom et al. 2001), where ξ_k is the posterior mean vector and \mathbf{P}_k is the posterior covariance matrix.
- *Actions*. The actions include the forward accelerations and the bank angles of all UAVs. More precisely, the action at time k is given by $u_k = [a_k; \phi_k]$, where a_k and ϕ_k are the vectors containing the forward accelerations and the bank angles, respectively, for all the UAVs.
- *State-transition law*. Since there are three subsystems, it is convenient to define the state-transition law for each subsystem separately. The sensor state evolves according to $s_{k+1} = \psi(s_k, u_k)$, where ψ is a mapping function (defined later). The target state evolves according to $\chi_{k+1} = f(\chi_k) + v_k$, where v_k represents an i.i.d. random sequence and f represents the target motion model (also defined later). Finally, the tracker state evolves according to the Kalman filter equations with a data association technique called *joint probabilistic data association* (JPDA) (Bar-Shalom and Fortmann 1988; Blackman and Popoli 1999).
- *Observations and observation law*. Let χ_k^{pos} and s_k^{pos} be the position vectors of a target and a sensor/UAV, respectively. Then the observation of the target's position is given by

$$z_k^\chi = \begin{cases} \chi_k^{\text{pos}} + w_k & \text{if the target is visible,} \\ \text{no measurement} & \text{if an obstacle blocks the} \\ & \text{line of sight,} \end{cases} \quad (73.7)$$

where w_k represents a random measurement error with a distribution that depends on the locations of the UAV (s_k^{pos}) and the target (χ_k^{pos}). The sensor and the tracker states are assumed to be fully observable.

- *Cost function.* The mean-squared error between the tracks and the targets is used as the cost function

$$C(x_k, u_k) = E_{v_k, w_{k+1}} [||\chi_{k+1} - \xi_{k+1}||^2 | x_k, u_k].$$

- *Belief state.* The belief state at time k is given by $b_k = (b_k^s, b_k^\chi, b_k^\xi, b_k^{\mathbf{P}})$, where $b_k^s = \delta(s - s_k)$, $b_k^\xi = \delta(\xi - \xi_k)$, $b_k^{\mathbf{P}} = \delta(\mathbf{P} - \mathbf{P}_k)$ (since the sensor and the tracker states are fully observable), and b_k^χ is the posterior distribution of the target state.

73.4.3 NBO Approximation Method

A linearized target motion model with zero-mean noise is considered, as given below:

$$\chi_{k+1} = \mathbf{F}_k \chi_k + v_k, \quad v_k \sim \mathcal{N}(0, \mathbf{Q}_k), \quad (73.8)$$

and the observations are given by

$$z_k^\chi = \mathbf{H}_k \chi_k + w_k, \quad w_k \sim \mathcal{N}(0, \mathbf{R}_k(\chi_k, s_k)),$$

where \mathbf{F}_k is the target motion model and \mathbf{H}_k is the observation model (73.7). The *constant velocity* (CV) model (Blackman and Popoli 1999; Bar-Shalom et al. 2001) is considered for target dynamics (73.8), which defines \mathbf{F}_k and \mathbf{Q}_k . Since the probability distributions are assumed to be Gaussian, the target belief state can be expressed (or approximated) as $b_k^\chi(\chi) = \mathcal{N}(\chi - \xi_k, \mathbf{P}_k)$, where ξ_k and \mathbf{P}_k evolve according to the JPDA algorithm (Bar-Shalom and Fortmann 1988; Blackman and Popoli 1999).

According to the *nominal belief-state optimization* (NBO) method, the objective function is approximated as follows:

$$J_H(b_0) \approx \sum_{k=0}^{H-1} c(\hat{b}_k, u_k),$$

where $\hat{b}_1, \hat{b}_2, \dots, \hat{b}_{H-1}$ is a *nominal* belief-state sequence and the optimization is over an action sequence u_0, u_1, \dots, u_{H-1} . The nominal target belief-state sequence can be identified with the nominal tracks $(\hat{\xi}_k, \hat{\mathbf{P}}_k)$, which are obtained from the Kalman filter equations (Blackman and Popoli 1999; Bar-Shalom et al. 2001) with exactly zero-noise sequence as follows:

$$\begin{aligned}\hat{b}_k^\chi(\chi) &= \mathcal{N}(\chi - \hat{\xi}_k, \hat{\mathbf{P}}_k), \\ \hat{\xi}_{k+1} &= \mathbf{F}_k \hat{\xi}_k, \\ \hat{\mathbf{P}}_{k+1} &= \left[\hat{\mathbf{P}}_{k+1|k}^{-1} + \mathbf{H}_{k+1}^T \left[\mathbf{R}_{k+1} (\hat{\xi}_{k+1}, s_{k+1}) \right]^{-1} \mathbf{H}_{k+1} \right]^{-1},\end{aligned}$$

where $\hat{\mathbf{P}}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{P}}_k \mathbf{F}_k^T + \mathbf{Q}_k$ and $s_{k+1} = \psi(s_k, u_k)$ (ψ is defined in the next subsection). The cost function, i.e., the mean-squared error between the tracks and the targets, can be written as

$$c(\hat{b}_k, u_k) = \text{Tr} \hat{\mathbf{P}}_{k+1}.$$

Therefore, the goal is to find an action sequence $(u_0, u_1, \dots, u_{H-1})$ that minimizes the cumulative cost function (with truncated horizon Miller et al. 2009)

$$J_H(b_0) = \sum_{k=0}^{H-1} \text{Tr} \hat{\mathbf{P}}_{k+1} = \sum_{k=0}^{H-1} \left(\sum_{i=1}^{N_{\text{targs}}} \text{Tr} \hat{\mathbf{P}}_{k+1}^i \right), \quad (73.9)$$

where N_{targs} represents the number of targets and $\hat{\mathbf{P}}_{k+1}^i$ represents the nominal error covariance matrix of i th target at time $k + 1$. When there are multiple UAVs, the nominal covariance matrix for i th target at time $k + 1$ can be expressed (based on data-fusion techniques) as

$$\hat{\mathbf{P}}_{k+1}^i = \left[\sum_{j=1}^{N_{\text{sens}}} (\hat{\mathbf{P}}_{k+1}^{i,j})^{-1} \right]^{-1},$$

where N_{sens} represents the number of sensors/UAVs and $\hat{\mathbf{P}}_{k+1}^{i,j}$ is the nominal covariance matrix of the i th target computed at the j th sensor. See Miller et al. (2009) for a detailed description of the NBO method.

73.4.4 UAV Kinematics

In this subsection, the mapping function ψ is defined, which was introduced in Sect. 73.4.2 to describe the evolution of the sensor (UAV) state given an action, i.e., $s_{k+1} = \psi(s_k, u_k)$. The state of the i th UAV at time k is given by $s_k^i = (p_k^i, q_k^i, V_k^i, \theta_k^i)$, where (p_k^i, q_k^i) represents the position coordinates, V_k^i represents the speed, and θ_k^i represents the heading angle. Let a_k^i be the forward acceleration (control variable) and ϕ_k^i be the bank angle (control variable) of the UAV, i.e., $u_k^i = (a_k^i, \phi_k^i)$. The mapping function ψ can be specified as a collection of simple

kinematic equations that govern the UAV motion. The kinematic equations of the UAV motion (Geiger et al. 2006) are as follows:

1. Speed update:

$$V_{k+1}^i = [V_k^i + a_k^i T]_{V_{\min}^{V_{\max}}}, \quad (73.10)$$

where $[v]_{V_{\min}^{V_{\max}}} = \max \{V_{\min}, \min(V_{\max}, v)\}$,

2. Heading angle update:

$$\theta_{k+1}^i = \theta_k^i + (gT \tan(\phi_k^i) / V_k^i), \quad (73.11)$$

3. Location update:

$$p_{k+1}^i = p_k^i + V_k^i T \cos(\theta_k^i),$$

$$q_{k+1}^i = q_k^i + V_k^i T \sin(\theta_k^i),$$

where V_{\min} and V_{\max} are the minimum and the maximum limits on the speed of the UAVs, g is the acceleration due to gravity, and T is the length of the time step.

73.4.5 Empirical Study of NBO

The NBO method is implemented in MATLAB, where the optimization problem discussed in Sect. 73.4.3 is solved using the command *fmincon* (gradient-based search algorithm). The measurement error, i.e., w_k , in (73.7) is distributed according to the normal distribution $\mathcal{N}(0, \mathbf{R}_k(\chi_k, s_k))$, where \mathbf{R}_k reflects 10 % range uncertainty and 0.01π radian angular uncertainty. In all the simulations in this study, the time horizon in (73.9) is set to $H = 6$. The goal of this empirical study is to illustrate the flexibility of the POMDP framework in being able to incorporate features of interest simply by “plugging in” the appropriate feature model into the framework and to compare the quantitative performance over varying parameter values.

This subsection provides the simulation of a few scenarios to demonstrate the coordinated behavior among the UAVs and the ability of the planning algorithm to track multiple targets with a single UAV. This subsection also provides empirical results to compare the tracking performance of variable-speed UAVs with that of the fixed-speed UAVs. In these simulations, a sequence of red circles represents the trajectory of a target, and a curve joining the arrows represents the trajectory of a UAV, where the size of an arrow is proportional to the instantaneous speed of the UAV and the arrow points toward the instantaneous heading direction of the UAV.

73.4.5.1 Coordinated UAV Motion

Figure 73.2 shows the simulation of a scenario with three UAVs and two targets, which depicts the scenario at the end of the simulation. Both targets start at the bottom, and as the simulation progresses, one target moves toward the northeast and the other moves toward the northwest. The UAVs start at the bottom and move according to the kinematic equations in Sect. 73.4.4, with controls obtained from the command *fmincon*, which minimize the cumulative cost function (73.9). The UAVs

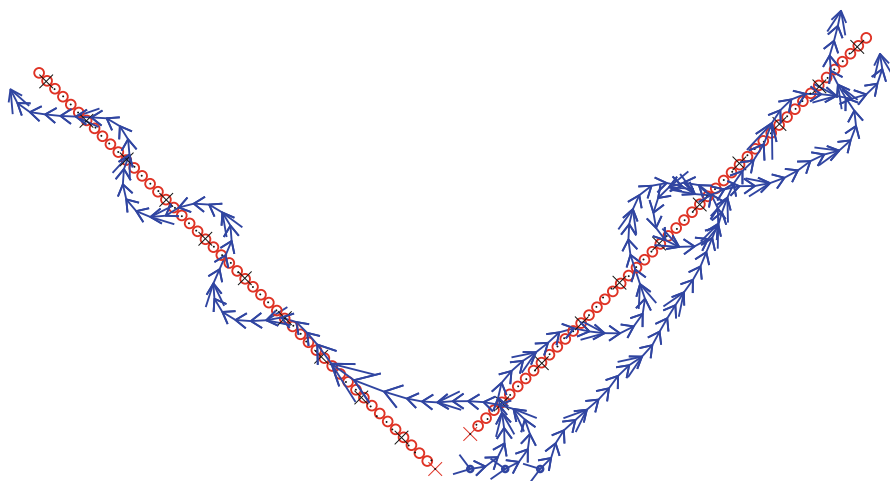


Fig. 73.2 Three UAVs (*blue*) tracking two targets (*red*)

coordinate to maximize the coverage of the targets as shown in Fig. 73.2. To evaluate how the agility in UAV motion affects the quantitative tracking performance, the scenario in Fig. 73.2 is simulated for 1,000 Monte Carlo runs with variable-speed UAVs and then with fixed-speed UAVs. Figure 73.3 shows the cumulative frequency of the average target-location errors for both fixed-speed and variable-speed UAVs. It is evident from Fig. 73.3 that variable-speed UAVs give better performance over fixed-speed UAVs.

73.4.5.2 Weaving Between Targets

Figure 73.4 shows the simulation of a scenario with one UAV and two targets, which depicts the scenario at the end of the simulation. Both targets start from the left and move toward the right. The UAV weaves between the targets as shown in Fig. 73.4, so that the average speed of the UAV toward the right is close to the speed of the targets. Also, the tracks get refined by looking at the targets from different angles, which is achieved by weaving between the targets. This scenario demonstrates the ability of the planning algorithm to maximize the coverage of multiple targets with a single UAV.

73.5 Model Enhancements and Additional Features

This section shows how to incorporate a variety of features of interest naturally into the POMDP framework. Specifically, the following features are incorporated by appropriately formulating the state-transition law and the objective function: (1) account for the effect of wind disturbance on UAVs, (2) avoid collisions between

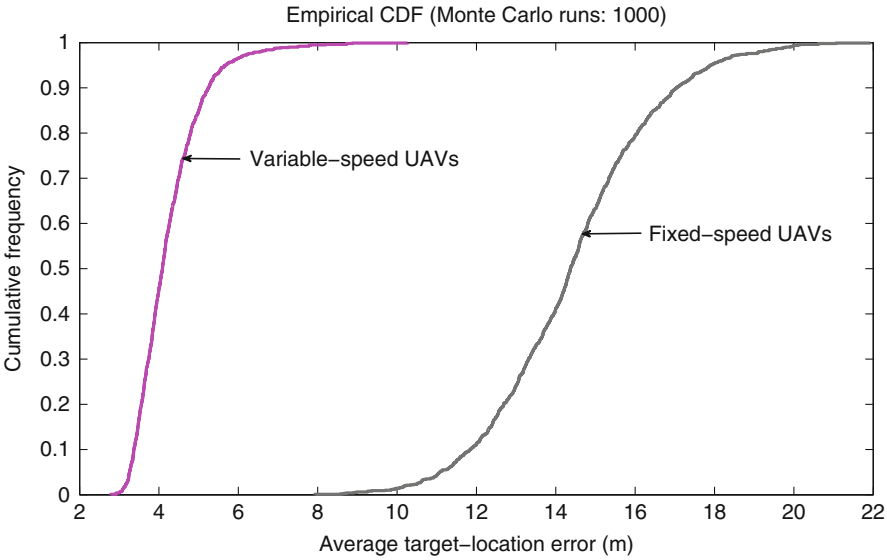


Fig. 73.3 Performance comparison: variable-speed UAVs (speed 11–26 m/s) vs. fixed-speed UAVs (speed: 15 m/s)

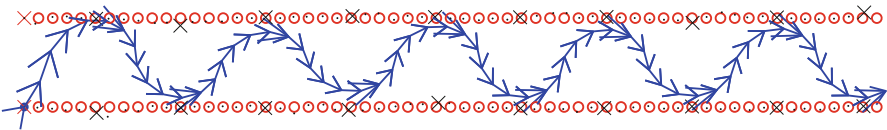


Fig. 73.4 One UAV (*blue*) tracking two targets (*red*)

UAVs and obstacles and among UAVs, (3) track targets while evading threats, (4) track evasive targets, and (5) mitigate track swaps.

73.5.1 Wind Compensation

If unaccounted for, wind drags the UAVs from their planned paths, which results in tracking performance degradation. In this subsection, a wind-compensation method is discussed in the context of the POMDP framework. More precisely, the speed and direction of wind are incorporated into the framework to nullify the effect of wind on UAVs.

The NBO method requires a nominal belief-state sequence of future states of the targets and the sensors (UAVs) given the actions. Since the planning algorithm in this study has a *look-ahead* property, it is important to have a UAV motion model that correctly predicts the trajectory of each UAV. [Figure 73.5a](#) shows the simulation of a scenario with one UAV and one target in the presence of wind. A simple

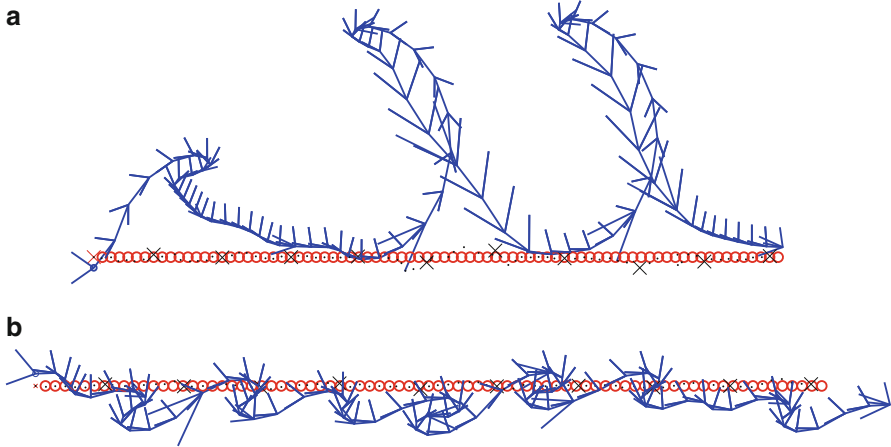


Fig. 73.5 UAV (blue) tracking a target (red) in the presence of wind. (a) No wind compensation; wind speed = 15 knots (north) and (b) wind compensation; wind speed = 15 knots (north)

wind model is used, i.e., wind blowing in the north direction at a constant speed. Figure 73.5a shows a plot at the end of the simulation where wind was unaccounted for, which resulted in drifting of the UAV from its desired course. As a result, the tracking performance (measured in average target-location error) gets deteriorated. To nullify the effect of wind, the speed and direction of wind are incorporated into the state-transition law corresponding to the sensor subsystem, i.e., the kinematics of the UAV motion. The information on wind (i.e., the speed and direction of wind) is assumed to be available (e.g., via measurements from a pitot tube). The modified location update equations are given below:

$$\begin{aligned} p_{k+1}^i &= p_k^i + V_k^i T \cos(\theta_k^i) + w_x T, \\ q_{k+1}^i &= q_k^i + V_k^i T \sin(\theta_k^i) + w_y T, \end{aligned} \quad (73.12)$$

where w_x and w_y are the average wind speeds in x and y directions. The speed and bank angle update equations remain the same as in (73.10) and (73.11). This wind-compensation method is not limited to constant wind scenarios (as in the above example); any wind model can be incorporated into the framework.

Figure 73.5b shows the simulation of the same scenario as in Fig. 73.5a, but with modified UAV kinematic model (73.12). With wind compensation, the UAV trajectory stays close to that of the target's, which improves the tracking performance. These scenarios (with and without wind compensation) are simulated for 2,000 Monte-Carlo runs; at the end of every run, the average target-location error is calculated. Figure 73.6 shows the cumulative frequency of average target-location errors for various wind speeds, which also demonstrates the advantage of having wind compensation. From Fig. 73.6, it is evident that the tracking performance deteriorates with increasing wind speed when wind is not compensated for, and

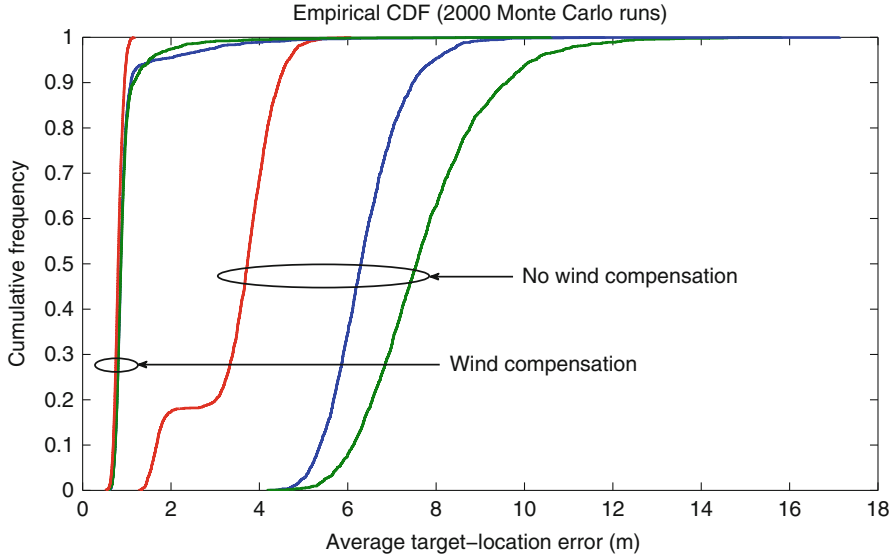


Fig. 73.6 Tracking performance under various wind speeds: 10 knots (*red*), 15 knots (*blue*), and 20 knots (*green*)

with wind compensation, the tracking performance does not deteriorate even with increasing wind speed.

73.5.2 Collision Avoidance

73.5.2.1 Collision Avoidance Between UAVs and Obstacles

Suppose that the targets are moving on the ground in the presence of obstacles where the UAV flying altitude is lower than the height of the obstacles. A collision between the UAVs and the obstacles may happen if the obstacles are unaccounted for. These obstacles also act as occlusions, i.e., a sensor cannot generate the measurement of a target if it is occluded. This section shows how to avoid collisions between UAVs and obstacles in the context of the POMDP framework. To avoid collisions, a penalty term (that increases as the UAVs come close to obstacles) is included in the objective function. Let N_{targs} represent the number of targets and N_{sens} represent the number of UAVs. The new objective function is written as follows:

$$J_H = \sum_{k=0}^{H-1} \left(\sum_{i=1}^{N_{\text{targs}}} \text{Tr} \hat{\mathbf{P}}_{k+1}^i + \gamma \sum_{j=1}^{N_{\text{sens}}} P_{k+1}^{\text{coll},j} \right), \quad (73.13)$$

where $P_{k+1}^{\text{coll},j}$ is a collision penalty function corresponding to the j th UAV and γ is a scaling factor. For the purpose of the 2-D simulation environment, the obstacles

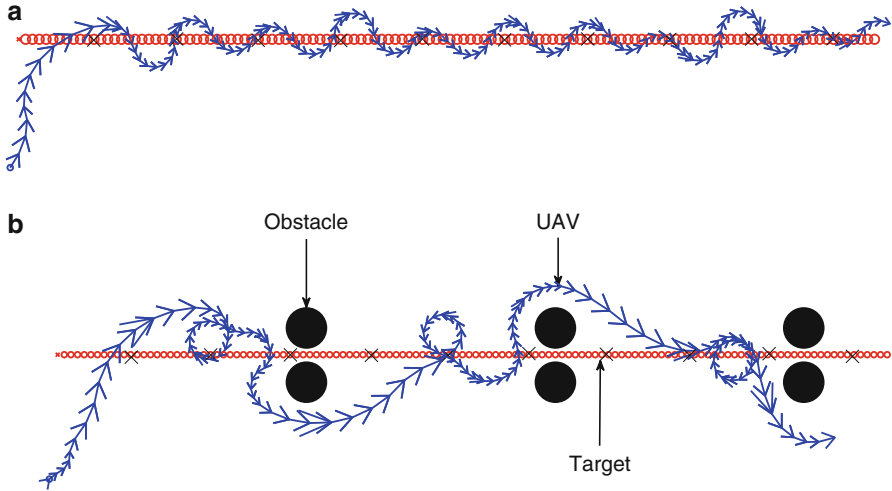


Fig. 73.7 UAV tracking a target while evading obstacles: UAV (blue), target (red), and obstacles (black). (a) No obstacles and (b) obstacles in the path of the UAV

are modeled as circular regions in the plane (surveillance region). Any continuous function can be used as the collision penalty function provided that the function increases as a UAV approaches an obstacle. For the purpose of the simulation, a simple linear function is used as the penalty function as given below:

$$P_{k+1}^{\text{coll},j} = \begin{cases} D - d_{k+1}^j & \text{if } d_{k+1}^j < D, \\ 0 & \text{otherwise,} \end{cases} \quad (73.14)$$

where d_{k+1}^j is the distance between the j th UAV and the closest obstacle from its location at time $k + 1$ and D is a constant such that the penalty function is nonzero only when d_{k+1}^j is less than D (i.e., the obstacles do not affect the planning algorithm when the UAVs are far from the obstacles). The UAV can detect its distance from the surrounding obstacles. Therefore, d_{k+1}^j ($\forall j$) in (73.14) is available at all times.

Figure 73.7b shows the simulation of a scenario with one UAV and one target, where the target moves on the ground in the presence of obstacles. Both the target and the UAV start at the left, and as the target moves toward the right, the UAV tracks the target while evading the obstacles, demonstrating the effectiveness of the inclusion of the penalty term in the objective function (73.13). If there were no obstacles, the UAV trajectory would have been close to that of the target as in Fig. 73.7a. When the UAV goes far from the target (when avoiding obstacles), the trace objective increases. However, the penalty from the collision penalty function is reduced (the parameter γ in (73.13) is set to a sufficiently large value to induce this behavior).

73.5.2.2 Collision Avoidance Among UAVs

Consider a multi-UAV scenario, where all UAVs are flying at the same altitude. To avoid collisions among the UAVs, a penalty term is incorporated into the objective function similar to (73.13), as given below:

$$J_H = \sum_{k=0}^{H-1} \left(\sum_{i=1}^{N_{\text{targs}}} \text{Tr} \hat{\mathbf{P}}_{k+1}^i + \gamma \sum_{j=1}^{N_{\text{sens}}} P_{k+1}^{\text{coll},j} \right), \quad (73.15)$$

where

$$P_{k+1}^{\text{coll},j} = \begin{cases} D - d_{k+1}^j & \text{if } d_{k+1}^j < D, \\ 0 & \text{otherwise,} \end{cases} \quad (73.16)$$

where d_{k+1}^j is the distance between the j th UAV and the closest neighboring UAV at time $k + 1$, i.e.,

$$d_{k+1}^j = \min_{i, i \neq j} d_{k+1}^{ji},$$

where d_{k+1}^{ji} is the distance between the j th UAV and the i th UAV.

The significance of D in (73.16) is as follows: a UAV contributes to the penalty function only if it approaches another UAV such that the distance between them becomes less than D . This penalty function penalizes those actions (over time) that cause any two UAVs to come close to each other, which effectively avoids collisions among the UAVs. Suppose that the safe distance (minimum spacing between two aircrafts, which is maintained to avoid midair collisions) between two aircrafts is 100m. The value of D in (73.16) is set equal to the safe distance, i.e., $D = 100$. Figure 73.8 shows the simulation of a scenario, where two UAVs are tracking one target with enhancement to the objective function (73.15). Figure 73.9 shows the plot of the distance between the two UAVs (in the scenario Fig. 73.8) as a function of time for $\gamma = 10$. From Fig. 73.9, it is evident that, with the enhancement to the objective function (73.15), the distance between the UAVs is greater than D , i.e., the safe distance, at all times. This shows that the enhancement to the objective function enables the planning algorithm to guide the UAVs for tracking a target while avoiding collisions among the UAVs by keeping a safe distance (D) between them.

A performance metric called *collision threat index* (CTI) is introduced for the scenario in Fig. 73.8, which is defined as the fraction of time the distance between the UAVs is less than D . The scenario in Fig. 73.8 is simulated for 1,000 Monte Carlo runs for various values of γ in (73.13). Figure 73.10a shows the plot of the cumulative frequency of CTI for various values of γ . It is evident from Fig. 73.10a that the performance with respect to CTI can be improved by increasing the value of γ . Table 73.1 shows that the inclusion of the collision penalty function in the objective function has significantly improved the performance in terms of CTI. However, increasing the value of γ reduces the tracking performance with respect to the average target-location error, which is evident from Fig. 73.10b. Figure 73.10b

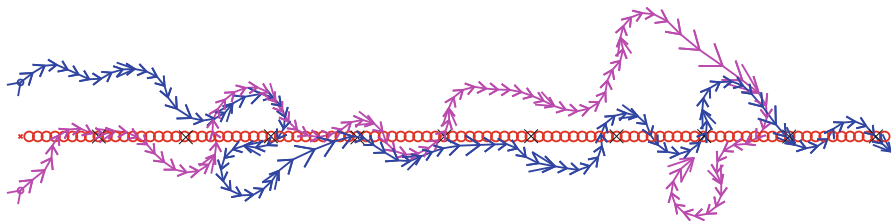


Fig. 73.8 Two UAVs tracking a target while avoiding collisions among themselves: UAVs (blue, magenta) and target (red)

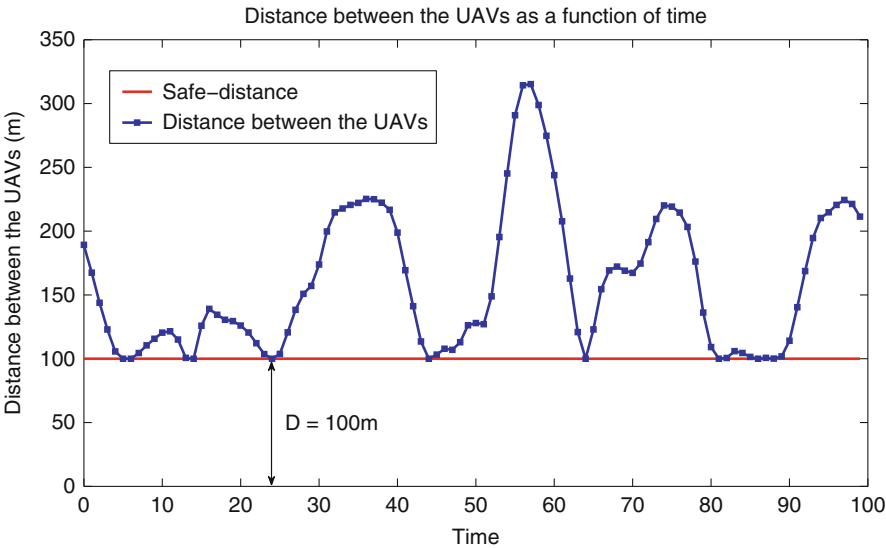


Fig. 73.9 Distance between the UAVs as a function of time (for scenario in Fig. 73.8)

shows the plot of the cumulative frequency of average target-location errors for various values of γ . In summary, γ can be used as a parameter to tune the trade-off between the performance with respect to CTI and the performance with respect to average target-location error.

73.5.3 Evading Threats

For the purpose of this section, a *threat* is a ground vehicle that actively pursues UAVs that are tracking it. Each threat knows the locations of all UAVs at all times

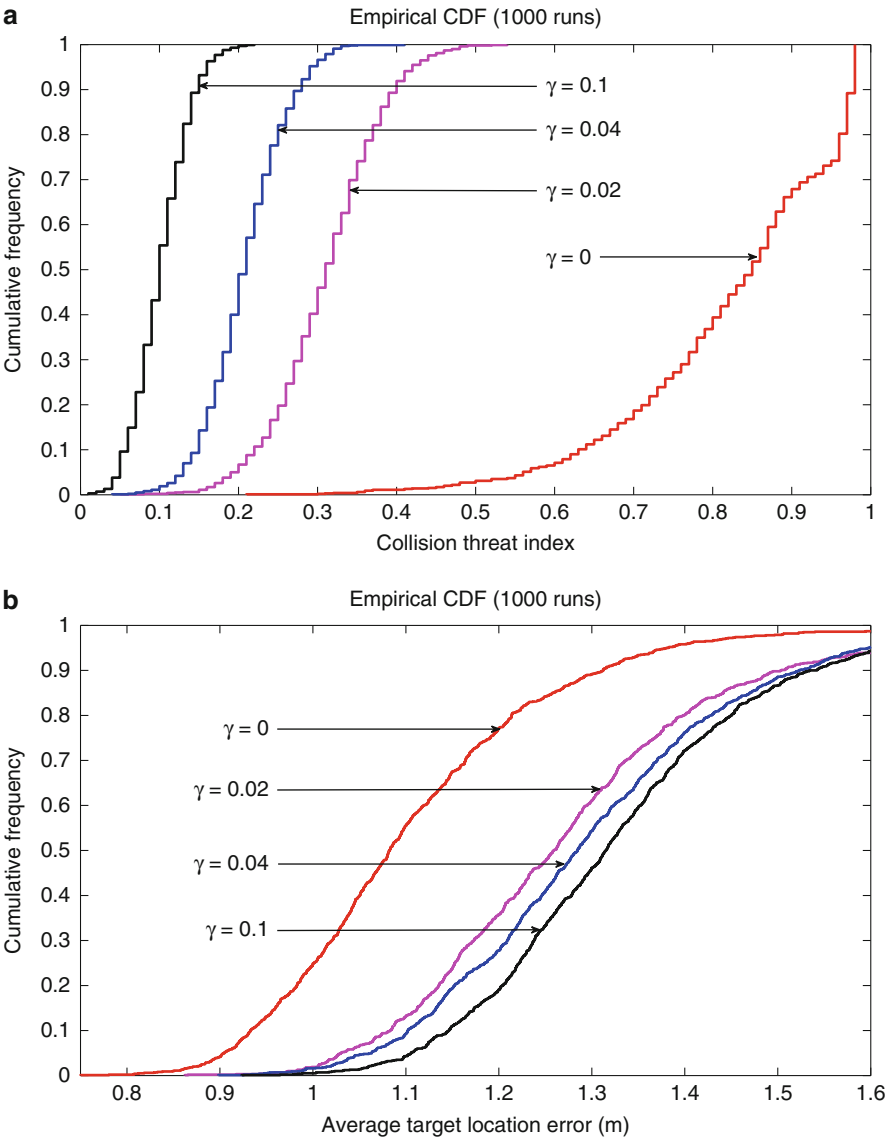


Fig. 73.10 Cumulative frequency of performance measures for various values of γ in (73.13). (a) CTI and (b) Avg. target-location error

and aligns its direction of motion toward the closest UAV at every time step. The goal here is to design the guidance algorithm in such a way that the UAVs track targets while avoiding threats by keeping a safe distance from them. The next subsection introduces a threat motion model that describes the evolution of the threat state.

Table 73.1 Average CTI (averaged over 1,000 runs) with and without the collision penalty function in the objective function

| Objective function | CTI (average) |
|---|---------------|
| $\text{Tr } \mathbf{P}$ | 0.823 |
| $\text{Tr } \mathbf{P} + 0.1 (P^{\text{coll}})$ | 0.102 |

73.5.3.1 Threat Motion Model

Let χ_k^t represent the state of a threat at time k , which includes the position coordinates (x_k and y_k) and speed (v_k) of the threat, i.e.,

$$\chi_k^t = [x_k, y_k, v_k]^T.$$

The threat state evolves according to the following equation:

$$\chi_{k+1}^t = \Phi_k \chi_k^t + n_k, \quad n_k \sim \mathcal{N}(0, \mathbf{Q}_k^t), \quad (73.17)$$

where \mathbf{Q}_k^t represents the process noise covariance matrix and

$$\Phi_k = \begin{bmatrix} 1 & 0 & \cos(h_k)T \\ 0 & 1 & \sin(h_k)T \\ 0 & 0 & 1 \end{bmatrix},$$

where h_k is the heading angle of the threat, which is given by

$$h_k = \tan^{-1} \left(\frac{y_k^{\text{closest UAV}} - y_k}{x_k^{\text{closest UAV}} - x_k} \right), \quad (73.18)$$

where $(x_k^{\text{closest UAV}}, y_k^{\text{closest UAV}})$ represent the position coordinates of the closest UAV from the threat at time k and (x_k, y_k) represent the position coordinates of the threat at time k . The heading angle Eq. (73.18) captures the pursuing property of the threat.

73.5.3.2 Threat Belief-State Evolution

The observation of the threat state is given by

$$z_k^t = \mathbf{H}_k \chi_k^t + w_k, \quad w_k \sim \mathcal{N}(0, \mathbf{R}_k).$$

Since the threat state evolves according to (73.17), i.e., $\chi_{k+1}^t = f(\chi_k^t) + n_k$, the Kalman filter or the extended Kalman filter cannot be used to evaluate the threat state estimate because the function $f(\cdot)$ is nonlinear and not differentiable. Therefore, a heuristic approach is used to write the Kalman filter equations. The threat belief state can be expressed (or approximated) as $b_k^t(\chi) = \mathcal{N}(\chi - \xi_k^t, \mathbf{P}_k^t)$, where ξ_k^t and \mathbf{P}_k^t evolve according to a heuristic Kalman filter (differs from the Kalman filter only in the prediction step), where the prediction step is written as follows:

$$\begin{aligned}\xi_{k|k-1}^t &= \hat{\Phi}_k \xi_{k-1}^t, \\ \mathbf{P}_{k|k-1}^t &= \hat{\Phi}_k \mathbf{P}_{k-1}^t \hat{\Phi}_k^T + \mathbf{Q}_k^t,\end{aligned}$$

where $\hat{\Phi}_k$ is an approximation of Φ_k in (73.17), which is evaluated as follows:

$$\hat{\Phi}_k = \begin{bmatrix} 1 & 0 & \cos(\hat{h}_k)T \\ 0 & 1 & \sin(\hat{h}_k)T \\ 0 & 0 & 1 \end{bmatrix},$$

where

$$\hat{h}_k = \tan^{-1} \left(\frac{\hat{y}_k^{\text{closest UAV}} - \hat{y}_k}{\hat{x}_k^{\text{closest UAV}} - \hat{x}_k} \right), \quad (73.19)$$

where $(\hat{x}_k^{\text{closest UAV}}, \hat{y}_k^{\text{closest UAV}})$ are the position coordinates of the estimated closest UAV from the threat at time k and (\hat{x}_k, \hat{y}_k) are the estimated position coordinates of the threat at time k . The closest UAV from the threat can be found by evaluating the Euclidean distances between the threat and each UAV. The location of the threat is known only with uncertainty, which is given by the posterior distribution of the threat state. Therefore, the exact Euclidean distance between the threat and a UAV cannot be calculated. So, the Mahalanobis distance (Mahalanobis 1936) is evaluated, which is a statistical distance, between the threat and a UAV. The closest UAV from the threat is estimated as follows:

1. The Mahalanobis distance (Mahalanobis 1936) of the position coordinates of each UAV i ($s_k^{i,\text{pos}}$) from the probability distribution (Gaussian) of the threat's location given by $\mathcal{N}(\xi_k^{t,\text{pos}}, \mathbf{P}_k^{t,\text{pos}})$ is evaluated, where $\xi_k^{t,\text{pos}}$ represents the first two elements of the threat state estimate ξ_k^t and $\mathbf{P}_k^{t,\text{pos}}$ represents a 2×2 sub-matrix of the threat state error covariance matrix \mathbf{P}_k^t , such that

$$\mathbf{P}_k^t = \begin{bmatrix} \mathbf{P}_k^{t,\text{pos}} & \mathbf{A}_k \\ \mathbf{A}_k^T & \mathbf{B}_k \end{bmatrix}.$$

The Mahalanobis distance of the i th UAV from the probability distribution of the threat's location is given by

$$D_M^i = \sqrt{[s_k^{i,\text{pos}} - \xi_k^{t,\text{pos}}]^T [\mathbf{P}_k^{t,\text{pos}}]^{-1} [s_k^{i,\text{pos}} - \xi_k^{t,\text{pos}}]}. \quad (73.20)$$

2. The closest UAV I from the threat is estimated as follows:

$$I = \arg \min_i D_M^i.$$

The *nominal* threat belief state ($\hat{b}_k^{\chi,t}$) is identified with the *nominal* tracks of the threat ($\hat{\xi}_k^t, \hat{\mathbf{P}}_k^t$) as follows:

$$\begin{aligned}\hat{b}_k^{\chi,t}(\chi) &= \mathcal{N}\left(\chi - \hat{\xi}_k^t, \hat{\mathbf{P}}_k^t\right), \\ \hat{\xi}_{k+1}^t &= \hat{\Phi}_k \hat{\xi}_k^t, \\ \hat{\mathbf{P}}_{k+1}^t &= \left[(\hat{\mathbf{P}}_{k+1|k}^t)^{-1} + \mathbf{H}_{k+1}^T \left[\mathbf{R}_{k+1} \left(\hat{\xi}_{k+1}^t, s_{k+1} \right) \right]^{-1} \mathbf{H}_{k+1} \right]^{-1},\end{aligned}$$

where $\hat{\mathbf{P}}_{k+1|k}^t = \hat{\Phi}_k \hat{\mathbf{P}}_k^t \hat{\Phi}_k^T + \mathbf{Q}_k^t$. The *constant velocity* (CV) model (73.8) is incorporated for target dynamics into the NBO method (as in the previous subsection). The next subsection provides an enhancement to the objective function to guide UAVs for tracking targets while evading threats.

73.5.3.3 Enhancement to the Objective Function

Let N_{targs} represent the number of targets and N_{sens} represent the number of UAVs. To guide the UAVs for tracking targets while avoiding threats, a penalty metric is included in the objective function. This penalty term increases whenever a UAV comes close to a threat. Therefore, the Euclidean distance between the locations of a UAV and the closest threat from the UAV can be used as a penalty metric. As mentioned in the previous section, the exact Euclidean distance between a UAV and a threat cannot be evaluated because of the uncertainty in the location of the threat. Therefore, as before, the Mahalanobis distance (Mahalanobis 1936), which is a statistical distance, between a UAV and its closest threat is used as the penalty metric. The new objective function can be written as follows:

$$J_H(b_0) = \sum_{k=0}^{H-1} \left(\sum_{i=1}^{N_{\text{targs}}} \text{Tr} \hat{\mathbf{P}}_{k+1}^i + \gamma \sum_{j=1}^{N_{\text{sens}}} G_{k+1}^j \right), \quad (73.21)$$

where γ is a scaling constant and

$$G_{k+1}^j = \begin{cases} D - d_{k+1}^j & \text{if } d_{k+1}^j < D, \\ 0 & \text{otherwise,} \end{cases} \quad (73.22)$$

where

$$d_{k+1}^j = \min_m D_M^m(s_{k+1}^{j,\text{pos}}),$$

where $s_{k+1}^{j,\text{pos}}$ represents the position coordinates of the j th UAV and $D_M^m(s_{k+1}^{j,\text{pos}})$ represents the Mahalanobis distance (73.20) of $s_{k+1}^{j,\text{pos}}$ from the probability distribution of the location of m th threat, i.e., $\mathcal{N}(\hat{\xi}_{k+1}^{t,\text{pos},m}, \hat{\mathbf{P}}_{k+1}^{t,\text{pos},m})$, where $\hat{\xi}_{k+1}^{t,\text{pos},m}$ represents the first two elements of $\hat{\xi}_{k+1}^{t,m}$ and $\hat{\mathbf{P}}_{k+1}^{t,\text{pos},m}$ represents the top-left-corner 2×2

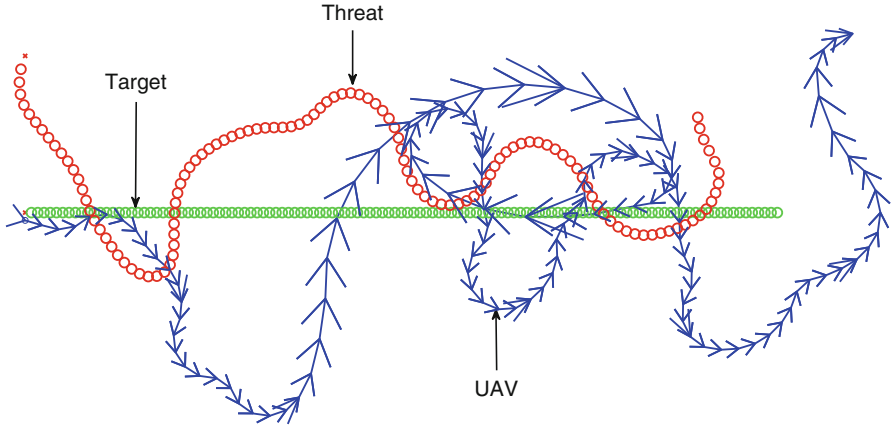


Fig. 73.11 UAV tracking a target while evading a threat: UAV (*blue*), threat (*red*), and target (*green*)

sub-matrix of $\hat{\mathbf{P}}_{k+1}^{t,m}$. The next subsection provides an empirical study to demonstrate the effectiveness of this enhancement.

73.5.3.4 Empirical Study

Figure 73.11 shows the simulation of a scenario with one UAV, one threat, and one target. In this simulation, the value of D in (73.22) is set to 100. Figure 73.12 shows the plot of the distance $D_{\text{threat-UAV}}$ between the UAV and the threat (in the scenario Fig. 73.11) as a function of time. It is evident from Fig. 73.12 that the UAV tracks the target while maintaining a safe distance, $D_{\text{safe}} (= D)$, from the threat. This demonstrates the effectiveness of the enhancement to the objective function (73.21).

A performance metric called the *threat index* is introduced for the scenario in Fig. 73.11, defined as the fraction of time the distance between the UAV and the threat is less than D . The scenario in Fig. 73.11 is simulated for 1,000 Monte Carlo runs for various values of γ used in (73.21). Figure 73.13a shows the plot of the cumulative frequency of threat indices for various values of γ . From Fig. 73.13a, it is evident that the performance with respect to *threat index* can be improved by increasing the value of γ . However, the performance with respect to average target-location error degrades with increasing γ , which is evident from Fig. 73.13b. In summary, the parameter γ can be used to tune the trade-off between the performance with respect to *threat index* and the performance with respect to average target-location error.

73.5.4 Tracking Evasive Targets

An evasive target is a ground vehicle that actively moves away from UAVs that are tracking it to avoid being tracked. In this section, all targets are evasive.

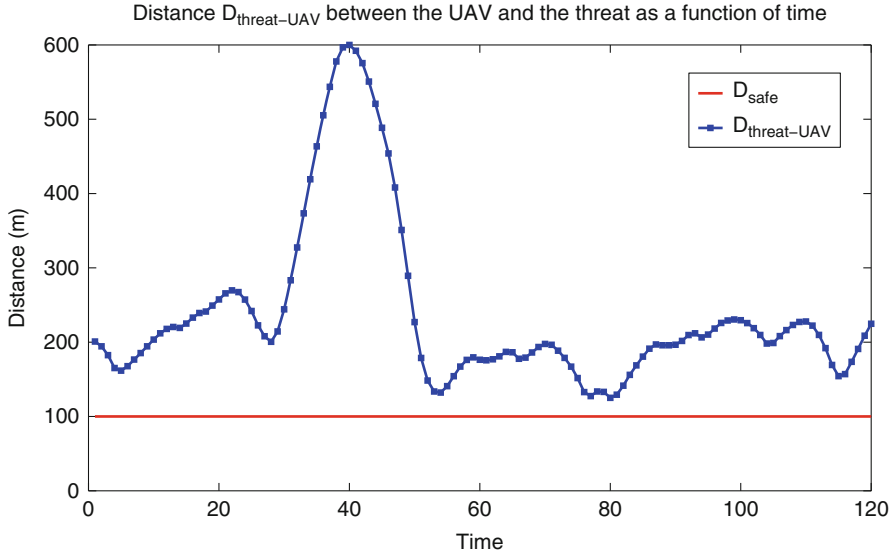


Fig. 73.12 Distance between the UAV and the threat as a function of time in the scenario Fig. 73.11

Each target knows the locations of all UAVs at every time step. The state of a target evolves according to the evasive target motion model as described in the following subsection.

73.5.4.1 Evasive Target Motion Model

The evasive target motion model described here is very similar to the threat motion model presented in the previous section. Let χ_k represent the state of a target at time k , which includes the position coordinates (x_k and y_k) and speed (v_k), i.e.,

$$\chi_k = [x_k, y_k, v_k]^T.$$

The target state evolves according to the following equation:

$$\chi_{k+1} = \Phi_k \chi_k + n_k, \quad n_k \sim \mathcal{N}(0, \mathbf{Q}_k^{\text{Evas}}),$$

where $\mathbf{Q}_k^{\text{Evas}}$ represents the process noise covariance matrix and

$$\Phi_k = \begin{bmatrix} 1 & 0 & \cos(h_k)T \\ 0 & 1 & \sin(h_k)T \\ 0 & 0 & 1 \end{bmatrix},$$

where h_k is the heading angle of the target, which is given by

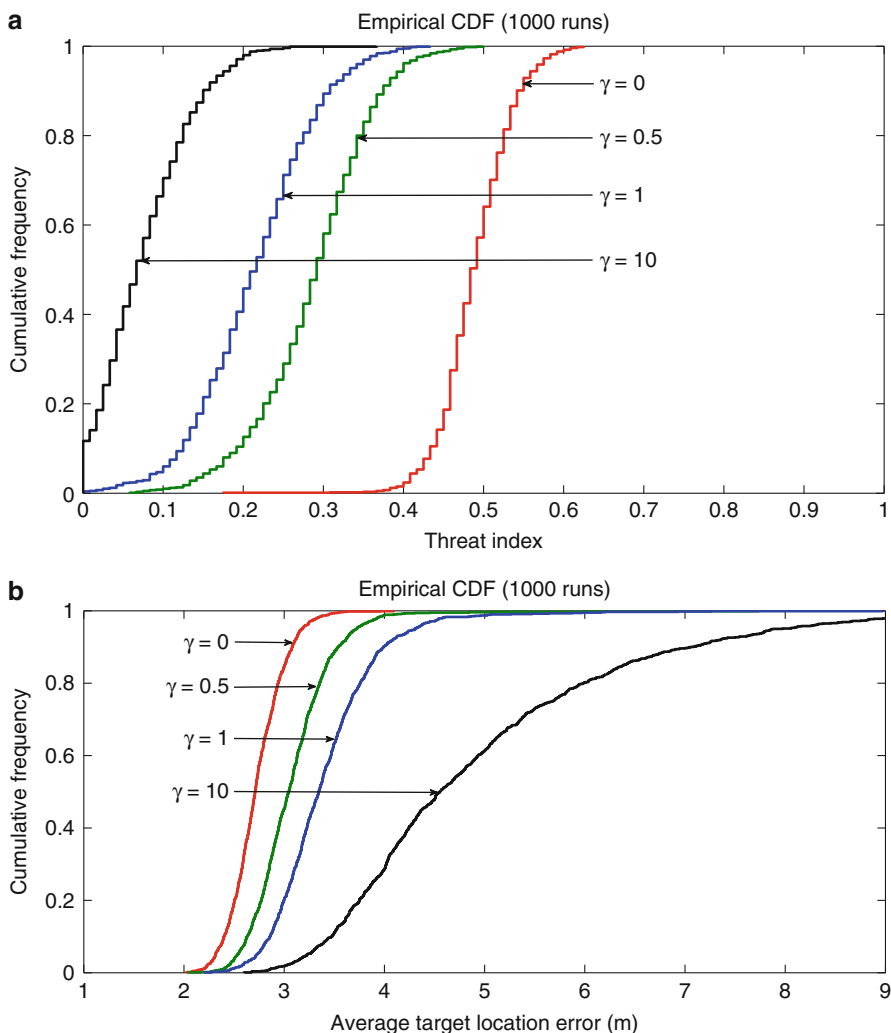


Fig. 73.13 Cumulative frequency of performance measures for various values of γ in the scenario Fig. 73.11. (a) Threat index and (b) Avg. target location error

$$h_k = \pi + \tan^{-1} \left(\frac{y_k^{\text{closest UAV}} - y_k}{x_k^{\text{closest UAV}} - x_k} \right), \quad (73.23)$$

where $(x_k^{\text{closest UAV}}, y_k^{\text{closest UAV}})$ are the position coordinates of the closest UAV from the target at time k and (x_k, y_k) are the position coordinates of the target at time k . Equation (73.23) differs from (73.19) in that the former induces an addition of π radians, representing motion away from the closest UAV.

73.5.4.2 Target Belief-State Evolution

As before, the observation of the target state is given by

$$z_k = \mathbf{H}_k \chi_k + w_k, \quad w_k \sim \mathcal{N}(0, \mathbf{R}_k).$$

The target belief state b_k^χ and the *nominal* target belief state \hat{b}_k^χ evolve according to the heuristic Kalman filter equations as in Sect. 73.5.3.2. However, the heading angle estimate (used in $\hat{\Phi}_k$) is evaluated according to the following equation:

$$\hat{h}_k = \pi + \tan^{-1} \left(\frac{\hat{y}_k^{\text{closest UAV}} - \hat{y}_k}{\hat{x}_k^{\text{closest UAV}} - \hat{x}_k} \right),$$

where $(\hat{x}_k^{\text{closest UAV}}, \hat{y}_k^{\text{closest UAV}})$ are the position coordinates of the estimated closest UAV from the target at time k (the closest UAV from the target is estimated in the same way the closest UAV from a threat was estimated in the previous section) and (\hat{x}_k, \hat{y}_k) are the estimated position coordinates of the target at time k . The nominal target belief-state update equations (described above) are incorporated into the NBO method. The objective function is given by

$$J_H(b_0) = \sum_{k=0}^{H-1} \left(\sum_{i=1}^{N_{\text{targs}}} \text{Tr} \hat{\mathbf{P}}_{k+1}^i \right),$$

where $\hat{\mathbf{P}}_{k+1}^i$ is the nominal error covariance matrix corresponding to the i th target at time $k+1$, which is obtained from the nominal target belief-state update equations (described above).

73.5.4.3 Empirical Study

To demonstrate the effectiveness of incorporating the evasive target motion model into the NBO method (as described in Sect. 73.5.4.2), its performance is compared with the use of a target motion model that does not account for the evasive property of the target (in particular *constant velocity* model). First, a scenario is simulated with one UAV and an evasive target while adopting the *constant velocity* model (henceforth referred to as the CV model) (Bar-Shalom et al. 2001; Blackman and Popoli 1999) for evasive target dynamics (which means the target is evasive, but the evasive property of the target is unaccounted for) (73.8). Second, the previous scenario is simulated while adopting the evasive target motion model (presented in Sect. 73.5.4.2) for target dynamics. Figure 73.14a and b shows the plots of one UAV tracking an evasive target with the CV model and with the evasive target motion model (Evas model), respectively. These scenarios are simulated for 1,000 Monte Carlo runs, and the cumulative frequency of average target-location errors is plotted, as shown in Fig. 73.15. From Fig. 73.15, it is evident that by adopting the Evas model over the CV model for the dynamics of an evasive target, the performance with respect to average target-location error can be improved significantly.

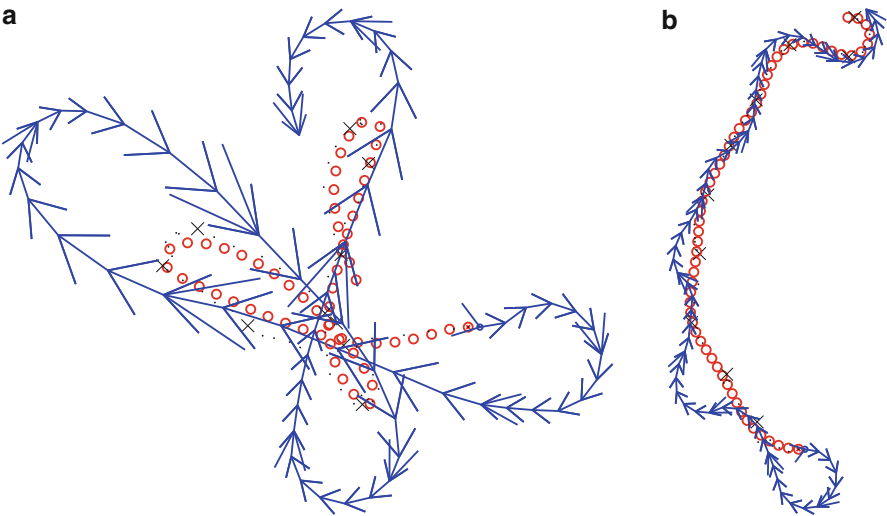


Fig. 73.14 UAV tracking an evasive target: UAV (*blue*) and target (*red*). (a) NBO with CV model and (b) NBO with Evas model

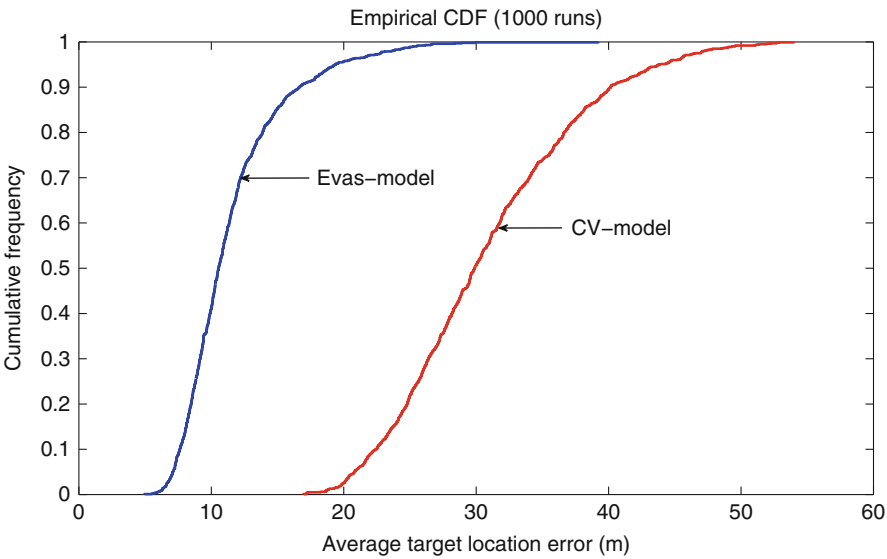


Fig. 73.15 Performance comparison of Evas model and CV model for the scenarios in Fig. 73.14

73.5.5 Track Swap Avoidance

A track swap is a switch in the association between the tracks and the targets. The identities of the targets are interpreted through the association between the tracks

and the targets. Therefore, a track swap switches the identities of targets, which is undesirable. The likelihood of a track swap depends on the tracker state, which in turn depends on the locations of UAVs relative to the targets. Therefore, track swaps can be mitigated by appropriately controlling the UAVs. This is achieved by incorporating a metric into the objective function that represents the risk of a track swap (this idea was first introduced in Miller et al. 2009).

73.5.5.1 Problem Description

Suppose that a UAV is tracking three targets as shown in Fig. 73.16. In this scenario, the bottom two targets come in close proximity to each other periodically, and the topmost target remains far from the bottom two targets. The planning algorithm discussed in Sect. 73.4 maximizes the coverage of the targets by guiding the UAV to weave between the topmost target and the bottom targets (as shown in Fig. 73.16), which is achieved by minimizing the overall trace objective. The likelihood of a track swap is high when the measurement sources are ambiguous. In the scenario of Fig. 73.16, when the UAV is far from the bottom targets, i.e., when the UAV is close to the topmost target, the likelihood of a track swap (corresponding to the bottom targets) is high because the chance that the sources of the measurements from the bottom two targets becoming ambiguous is high. The following subsection provides an enhancement (Miller et al. 2009) to mitigate track swaps.

73.5.5.2 Enhancement for Mitigating Track Swaps

The similarity between the target state distributions is a good predictor for a track swap because the likelihood of the measurement sources becoming ambiguous is high when the similarity between the target state distributions is high. The target state distributions depend on the tracker states and the locations of the UAVs over

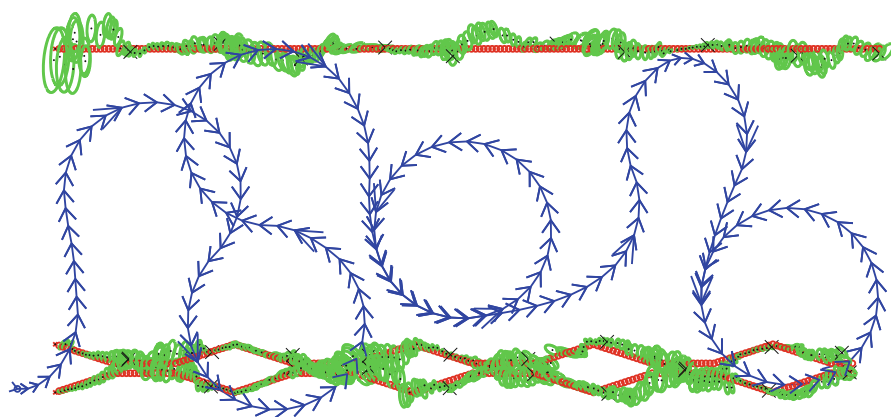


Fig. 73.16 UAV tracking three targets: UAV (*blue*), targets (*red*), and error concentration ellipses of target's location (*green*)

time. Therefore, the UAVs can be controlled appropriately such that the target state distributions are less similar. The similarity between the probability distributions can be measured as the inverse of a statistical distance between the distributions. To minimize the likelihood of a track swap, a term that is inversely proportional to a statistical distance (between the target state distributions) is incorporated into the objective function. The new objective function is written as

$$J_H = \sum_k \left(\sum_{i=1}^{N_{\text{targs}}} \text{Tr} \mathbf{P}_{k+1}^i + \gamma(1/D_{k+1}) \right), \quad (73.24)$$

where γ is a scaling factor, N_{targs} represents the number of targets, and

$$D_{k+1} = \min_{p \neq q; p, q \in S} D(\chi_{k+1}^p || \chi_{k+1}^q),$$

where $S = \{1, 2, \dots, N_{\text{targs}}\}$ and $D(\chi_{k+1}^p || \chi_{k+1}^q)$ is a statistical distance between the distributions of the p th target and the q th target.

There are several statistical distances defined in the literature: KL divergence (Kullback and Leibler 1951), Bhattacharyya distance (Bhattacharyya 1943), Hellinger distance (Cam and Yang 2000), and worst-case chi-square distance (see Miller et al. 2009 for a description of the worst-case chi-square distance). Figure 73.17 shows the simulation of a scenario with three targets and one UAV (similar to the scenario in Fig. 73.16) with the new objective function (73.24) (where the worst-case chi-square distance is used to measure the similarity between distributions). Figure 73.17 shows that, with enhancement to the objective function (73.24), the UAV stays close to the bottom two targets in contrast to the behavior in Fig. 73.16. This reduces the similarity between the probability distributions of the bottom targets, which in turn reduces the chance of a track swap.

KL divergence, Bhattacharyya distance, and Hellinger distance are average-case measures of how often the state values from the distributions fall in a small neighborhood of each other, whereas worst-case chi-square distance is a

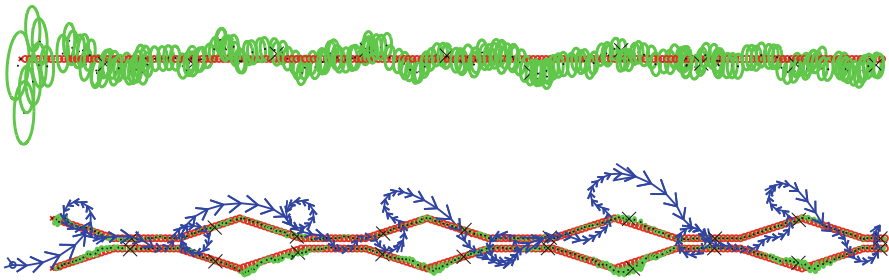


Fig. 73.17 UAV tracking three targets while mitigating track swaps

worst-case measure. Miller et al. (2009) explain that a worst-case measure is more suited to track swap prediction than an average-case measure. The next subsection provides an empirical study to show that the worst-case chi-square distance is an appropriate metric in predicting track swaps, which corroborates the claim in Miller et al. (2009).

73.5.5.3 Empirical Study

The scenario where a UAV tracks three targets (as described in Sect. 73.5.5.1) is simulated for 2,000 runs with and without the enhancement in the objective function. In every run, the fraction of track swaps—a metric to evaluate the tracking performance—is calculated according to the following method.

Fraction of Track Swaps: Let N_{TS} represent the number of track swaps, defined as follows. The tracks are checked for a track swap only at instances when the bottom targets are the farthest apart. These instances occur periodically, as can be seen in Figs. 73.16 and 73.17. At each of these instances, the associations between the tracks and the targets are evaluated. Whenever the association at a particular instance differs from the association in the previous instance, N_{TS} is incremented by one. At the end of the simulation, the fraction of track swaps is evaluated as follows:

$$\text{fraction of track swaps} = \frac{N_{TS}}{N},$$

where N represents the total number of instances the track associations are evaluated.

For the purpose of simulations, the following notation is used to represent the statistical distances: (1) KL divergence, D_{KL-div} ; (2) Hellinger distance $D_{Hell-dist}$;

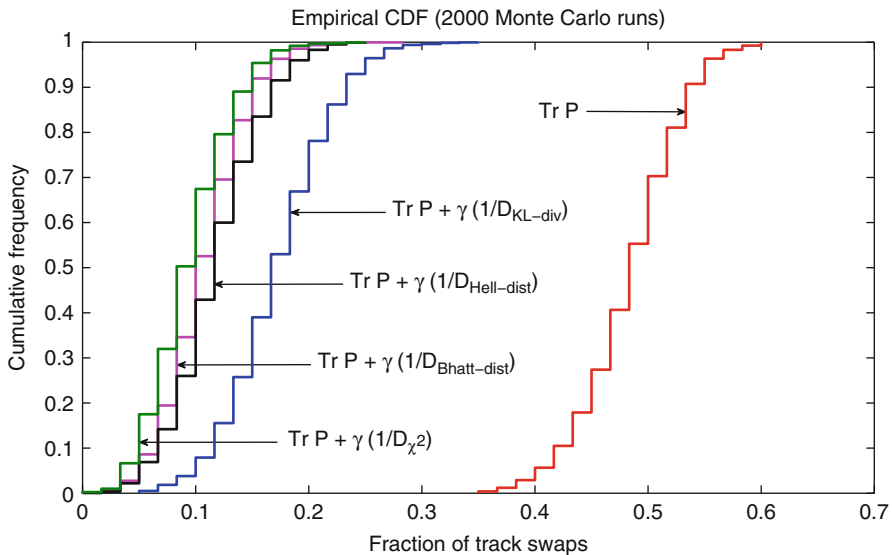


Fig. 73.18 Performance comparison for various statistical distances

Table 73.2 Fraction of track swaps (averaged over 2,000 Monte Carlo runs) for various statistical distances

| Objective function | Fraction of track swaps (average) |
|--|-----------------------------------|
| $\text{Tr } \mathbf{P}$ | 0.48 |
| $\text{Tr } \mathbf{P} + (\gamma/D_{\text{KL-div}})$ | 0.17 |
| $\text{Tr } \mathbf{P} + (\gamma/D_{\text{Hell-dist}})$ | 0.12 |
| $\text{Tr } \mathbf{P} + (\gamma/D_{\text{Bhatt-dist}})$ | 0.11 |
| $\text{Tr } \mathbf{P} + (\gamma/D_{\chi^2})$ | 0.08 |

(3) Bhattacharyya distance, $D_{\text{Bhatt-dist}}$; and (4) worst-case chi-square distance, D_{χ^2} . Figure 73.18 shows the plot of the cumulative frequency of the fraction of track swaps for various statistical distances. From Fig. 73.18 and Table 73.2, it is evident that the enhancement to the objective function improves the performance (in terms of the fraction of track swaps) significantly for all candidate statistical distances. It is also evident that D_{χ^2} is the best among other statistical distances in mitigating track swaps. This shows that the worst-case chi-square distance is an appropriate metric for predicting track swaps—which corroborates the claim in Miller et al. (2009).

73.6 Conclusion

In this study, a dynamic path-planning algorithm was designed to guide autonomous UAVs for multitarget tracking based on the theory of POMDPs. The key features of this study are summarized as follows:

- Variable-speed UAVs are considered for the purpose of this study. The UAVs are controlled by forward acceleration and bank angle. This study described how to incorporate these controls into the framework by appropriately formulating the POMDP action space and state-transition law. The results show that the variable-speed UAVs give better tracking performance over fixed-speed UAVs, which demonstrates that higher the agility in UAV motion better the tracking performance.
- A wind-compensation method is discussed in the context of POMDP framework. The speed and direction of wind are incorporated into the UAV kinematic model to nullify the effect of wind disturbance on UAVs. The results from the Monte Carlo simulations show that the performance remained good even with increasing wind speed when wind was accounted for, whereas the performance deteriorated significantly when wind was unaccounted for.
- To avoid collisions between UAVs and obstacles and among UAVs, a penalty term (scaled appropriately) is included in the objective function. The penalty function penalizes the actions that bring UAVs close to the obstacles or bring two UAVs close to each other (these two scenarios are addressed separately). In the first scenario, the simulation results show that by incorporating the penalty term into the objective function, a UAV tracks a target while safely evading the obstacles. In the second scenario, the inclusion of the penalty term in the objective function ensured that the UAVs track the targets while avoiding collisions among themselves. In this scenario, the Monte Carlo simulation results

show that, by varying a certain parameter (γ), one can trade off between the performances with respect to *collision threat index* (CTI) and average target-location error.

- This study presented a method to guide UAVs for tracking targets while evading threats. First, the threat motion model is defined while incorporating the pursuing property of threats. Second, the threat belief-state update equations are written based on the threat motion model. Third, a penalty term is included in the objective function, which penalizes the actions that bring the UAVs close to the threats. The results show that the enhancement to the objective function ensured that the UAVs track the targets while evading the threats. The results also show that the trade-off between the performances with respect to *threat index* and average target-location error can be tuned with a parameter γ .
- This study presented a method to guide UAVs to track evasive targets. The evasive property of targets is incorporated into the NBO method, which is accomplished by appropriately defining the evasive target motion model. First, the evasive target motion model (Evas model) is defined while incorporating the evasive property of targets. Second, the target belief-state update equations are written using the Evas model. Third, an empirical study is conducted to demonstrate the effectiveness of incorporating the evasive property of targets into the NBO method.
- To reduce the likelihood of a track swap, a term that is proportional to the inverse of the statistical distance (between the target state distributions) is included in the objective function. The simulation of the scenario with one UAV and three targets shows that the modified objective function successfully enabled the UAV to track the targets while reducing the similarity between the target state distributions, which effectively reduced the chance of a track swap. The performances of various statistical distances are compared, and the results support the argument that D_{χ^2} is an appropriate metric in predicting track swaps.

The above-mentioned features are incorporated into the framework by simply plugging in the appropriate models, showing the generality and flexibility of the framework.

Acknowledgments This material is based upon work supported in part by Northrop Grumman Corporation through the Rocky Mountain Aerospace Technology Incubator (RMATI) program and by AFOSR contract FA9550-09-1-0518.

References

- W. Adiprawita, A.S. Ahmad, J. Sembiring, B.R. Trilaksono, Reinforcement learning with heuristic to solve POMDP problem in mobile robot path planning, in *Proceedings of the International Conference on Electrical Engineering and Informatics*, Bandung, July 2011
- H. Bai, D. Hsu, M. Kochenderfer, W.S. Lee, Unmanned aircraft collision avoidance using continuous-state POMDPs, in *Proceedings of the Robotics: Science and Systems*, Los Angeles, June 2011
- Y. Bar-Shalom, T.E. Fortmann, *Tracking and Data Association* (Academic, London, 1988)

- Y. Bar-Shalom, X.R. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation* (Wiley, New York, 2001)
- R. Bellman, *Dynamic Programming* (Princeton University Press, Princeton, 1957)
- J. Berger, J. Happe, C. Gagne, M. Lau, Co-evolutionary information gathering for a cooperative unmanned aerial vehicle team, in *Proceedings of the 12th International Conference on Information Fusion*, Seattle, July 2009, pp. 347–354
- D.P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2 (Athena Scientific, Belmont, 2007)
- D.P. Bertsekas, D.A. Castanon, Rollout algorithms for stochastic scheduling problems. *J. Heuristics* **5**, 89–108 (1999)
- D.P. Bertsekas, J.N. Tsitsiklis, *Neuro-Dynamic Programming* (Athena Scientific, Belmont, 1996)
- A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.* **35**, 99–109 (1943)
- S. Blackman, R. Popoli, *Design and Analysis of Modern Tracking Systems* (Artech House, Boston, 1999)
- J. Borenstein, Y. Koren, Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. Syst. Man Cybern.* **19**, 1179–1187 (1989)
- J. Bruce, M. Veloso, Real-time randomized path planning for robot navigation, in *Proceedings of the International Conference on Intelligent Robots and Systems*, EPFL Lausanne, Oct 2002, pp. 2383–2388
- A. Bry, N. Roy, Rapidly-exploring random belief trees for motion planning under uncertainty, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, May 2011, pp. 723–730
- L.L. Cam, G.L. Yang, *Asymptotics in Statistics* (Springer, Berlin, 2000)
- S. Candido, S. Hutchinson, Minimum uncertainty robot path planning using a POMDP approach, in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Taipei, Oct 2010, pp. 1408–1413
- J. Capitán, M.T.J. Spaan, L. Merino, A. Ollero, Decentralized multi-robot cooperation with auctioned POMDPs, in *Proceedings of the Sixth Annual Workshop on Multiagent Sequential Decision-Making in Uncertain Domains*, Taipei, May 2011
- C.G. Cassandras, W. Li, A receding horizon approach for dynamic UAV mission management, in *Proceedings of SPIE 17th Annual International Symposium*, Orlando, Apr 2003, pp. 284–293
- N. Ceccarelli, J.J. Enright, E. Frazzoli, S.J. Rasmussen, C.J. Schumacher, Micro UAV path planning for reconnaissance in wind, in *Proceedings of the American Control Conference*, New York, July 2007, pp. 5310–5315
- E.K.P. Chong, R.L. Givan, H.S. Chang, A framework for simulation-based network control via hindsight optimization, in *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Dec 2000, pp. 1433–1438
- E.K.P. Chong, C. Kreucher, A.O. Hero, Partially observable Markov decision process approximations for adaptive sensing. *Discret. Event Dyn. Syst.* **19**, 377–422 (2009)
- R.V. Cowlagi, P. Tsiotras, Beyond quadrees: cell decompositions for path planning using wavelet transforms, in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, Dec 2007, pp. 1392–1397
- N. Dadkhah, B. Mettler, Survey of motion planning literature in the presence of uncertainty: considerations for UAV guidance. *J. Intell. Robot. Syst.* **65**, 233–246 (2012)
- T. Dong, X.H. Liao, R. Zhang, Z. Sun, Y.D. Song, Path tracking and obstacle avoidance of UAVs – fuzzy logic approach, in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Reno, May 2005, pp. 43–48
- Y. Du, D. Hsu, H. Kurniawati, W.S. Lee, S.C.W. Ong, S.W. Png, A POMDP approach to robot motion planning under uncertainty, in *Appeared in ICAPS 2010 POMDP Practitioners Workshop*, Toronto, May 2010

- B.R. Geiger, J.F. Horn, A.M. DeLullo, L.N. Long, Optimal path planning of UAVs using direct collocation with nonlinear programming, in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Keystone, Aug 2006
- C. Geyer, Active target search from UAVs in urban environments, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, May 2008, pp. 2366–2371
- C. Goerzen, Z. Kong, B. Mettler, A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *J. Intell. Robot. Syst.* **57**, 65–100 (2010)
- Z. He, R.V. Iyer, P.R. Chandler, Vision-based UAV flight control and obstacle avoidance, in *Proceedings of the American Control Conference*, Minneapolis, June 2006, pp. 2166–2170
- R. He, A. Bachrach, M. Achtelik, A. Geramifard, D. Gurdan, S. Prentice, J. Stumpf, N. Roy, On the design and use of a micro air vehicle to track and avoid adversaries. *Int. J. Robot. Res.* **29**, 529–546 (2010a).
- R. He, A. Bachrach, N. Roy, Efficient planning under uncertainty for a target-tracking micro-aerial vehicle, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, May 2010b
- A.L. Jennings, R. Ordonez, N. Ceccarelli, Dynamic programming applied to UAV way point path planning in wind, in *Proceedings of the IEEE Multi-conference on Systems and Control*, San Antonio, Sept 2008, pp. 215–220
- A. Kaneshige, T. Akamatsu, K. Terashima, Real time path planning based on the potential method with 3-D diffusion process for an overhead traveling crane, in *Proceedings of the 5th Asian Control Conference*, Melbourne, July 2004, pp. 1715–1722
- L.E. Kavraki, P. Svestka, J. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **12**, 566–580 (1996)
- L.E. Kavraki, P. Svestka, J. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **12**, 566–580 (1996)
- O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, Mar 1985, pp. 500–505
- Y. Kim, D. Gub, I. Postlethwaite, Real-time path planning with limited information for autonomous unmanned air vehicles. *J. Autom.* **44**, 696–712 (2008)
- Y. Koren, J. Borenstein, Potential field methods and their inherent limitations for mobile robot navigation, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, Apr 1991, pp. 1398–1404
- C. Kreucher, A.O. Hero, K. Kastella, D. Chang, Efficient methods of non-myopic sensor management for multitarget tracking, in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Dec 2004, pp. 722–727
- B.H. Krogh, C.E. Thorpe, Integrated path planning and dynamic steering control for autonomous vehicles, in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, Apr 1986, pp. 1664–1669
- S. Kullback, R.A. Leibler, On information and sufficiency. *Ann. Math. Stat.* **22**, 79–86 (1951)
- H. Kurniawati, Y. Du, D. Hsu, W.S. Lee, Motion planning under uncertainty for robotic tasks with long time horizons. *Int. J. Robot. Res.* **30**, 308–323 (2011)
- S.M. LaValle, Robot motion planning: a game-theoretic foundation. *Algorithmica* **26**, 430–465 (2000)
- S.M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, 2006)
- N.R.J. Lawrence, S. Sukkariéh, Path planning for autonomous soaring flight in dynamic wind fields, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, May 2011, pp. 2499–2505
- P. Leven, S. Hutchinson, A framework for real-time path planning in changing environments. *Int. J. Robot. Res.* **21**, 999–1030 (2002)
- W. Li, C.G. Cassandras, A cooperative receding horizon controller for multivehicle uncertain environments. *IEEE Trans. Autom. Control* **51**, 242–257 (2006)
- Y. Lu, X. Huo, O. Arslan, P. Tsiotras, Incremental multi-scale search algorithm for dynamic path planning with low worst-case complexity. *IEEE Trans. Syst. Man Cybern. B Cybern.* **41**, 1556–1570 (2011)

- X. Ma, D.A. Castanon, Receding horizon planning for dubins traveling salesman problems, in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, Dec 2006, pp. 5453–5458
- P.C. Mahalanobis, On the generalised distance in statistics, in *Proceedings of the National Institute of Sciences of India*, Calcutta, Apr 1936, pp. 49–55
- T.G. McGee, J.K. Hedrick, Path planning and control for multiple point surveillance by an unmanned aircraft in wind, in *Proceedings of the American Control Conference*, Minneapolis, June 2006, pp. 4261–4266
- T.G. McGee, S. Spry, J.K. Hedrick, Optimal path planning in a constant wind with a bounded turning rate, in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, San Francisco, Aug 2005
- M.L. McIntyre, W.E. Dixon, D.M. Dawson, B. Xian, Adaptive tracking control of on-line path planners: velocity fields and navigation functions, in *Proceedings of the American Control Conference*, Portland, June 2005, pp. 3168–3173
- S.A. Miller, Z.A. Harris, E.K.P. Chong, A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking. *EURASIP J. Adv. Signal Process.* **2009**, 724597 (2009)
- M. Naveed, A. Crampton, D. Kitchin, T.L. McCluskey, Real-time path planning using a simulation-based markov decision process, in *Proceedings of the Thirty-First SGAI International Conference on Artificial Intelligence*, Cambridge, Dec 2011
- I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis, A.N. Kostaras, Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Trans. Syst. Man Cybern. B Cybern.* **33**, 898–912 (2003)
- J. Ousingsawat, M.E. Campbell, On-line estimation and path planning for multiple vehicles in an uncertain environment. *Int. J. Robust Nonlinear Control* **14**, 741–766 (2004)
- J. Park, H. Oh, M. Tahk, Uav collision avoidance based on geometric approach, in *Proceedings of the SICE Annual Conference*, Tokyo, Aug 2008, pp. 2122–2126
- S. Park, H. Choi, N. Roy, J.P. How, Learning covariance dynamics for path planning of UAV sensors in a large-scale dynamic environment, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Chicago, Aug 2009
- J. Redding, J.N. Amin, J.D. Boskovic, A real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters, in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, Aug 2007, pp. 4261–4266
- W. Ren, J. Sun, R.W. Beard, T.W. McLain, Nonlinear tracking control for nonholonomic mobile robots with input constraints: an experimental study, in *Proceedings of the American Control Conference*, Portland, June 2005, pp. 4923–4928
- P.W. Sarunic, R.J. Evans, B. Moran, Control of unmanned aerial vehicles for passive detection and tracking of multiple emitters, in *Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications*, Ottawa, July 2009
- J.B. Saunders, O. Call, A. Curtis, A.W. Beard, T.W. McLain, Static and dynamic obstacle avoidance in miniature air vehicles, in *Proceedings of the Infotech at Aerospace Conference*, Arlington, Sep 2005, pp. 2005–6950
- A. Sud, E. Andersen, S. Curtis, M.C. Lin, D. Manocha, Real-time path planning in dynamic virtual environments using multiagent navigation graphs. *IEEE Trans. Vis. Comput. Graph.* **14**, 526–538 (2008)
- R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction* (MIT, Cambridge, 1998)
- L. Techy, C.A. Woolsey, K.A. Morgansen, Planar path planning for flight vehicles in wind with turn rate and acceleration bounds, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, May 2010, pp. 3240–3245
- R.B. Tilove, Local obstacle avoidance for mobile robots based on the method of artificial potentials, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, May 1990, pp. 566–571

- J. Tisdale, H. Durrant-Whyte, J.K. Hedrick. Path planning for cooperative sensing using unmanned vehicles, in *Proceedings of the ASME International Mechanical Engineering Conference and Exposition*, Seattle, Nov 2007, pp. 715–723
- G. Wu, E.K.P. Chong, R. Givan, Burst-level congestion control using hindsight optimization. *IEEE Trans. Autom. Control* **47**, 979–991 (2002)
- C. Yong, E.J. Barth, Real-time dynamic path planning for dubins' nonholonomic robot, in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, Dec 2006, pp. 2418–2423
- D. Zu, J.D. Han, M. Campbell, Artificial potential guided evolutionary path plan for multi-vehicle multi-target pursuit, in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Shenyang, Aug 2004, pp. 855–861

Section XIV

UAV Sense, Detect and Avoid Systems

Lennaert Speijker

Kimon P. Valavanis and George J. Vachtsevanos

The absence of a pilot onboard the UAV may endanger the life of the UAV itself and that of other flying objects, as well as endanger the life of human personnel on the ground when crashing, on top of other collateral damage. Safe UAV flying in the national airspace is a major concern of regulatory agencies, OEMs, and UAV users. UAVs flying in the national airspace must be capable of sensing and identifying the world around them, deciding upon potential collision risks, and taking appropriate actions to avoid such collisions.

However, UAS have emerged as a viable, operational technology for potential civil and commercial applications in the National Airspace System (NAS). Although this new type of technology presents great potential, it also introduces a need for a thorough inquiry into its safety impact on the NAS. The FAA, ICAO, and other regulatory agencies, worldwide, are stipulating requirements for safe UAV flight and their integration into the national airspace, which may be summarized in *one* statement: “UAVs should fly as if there were a human pilot onboard.”

Sense and avoid (SAA) remains one of the key enabling capabilities required for the safe airspace integration of UAS into the global airspace system. Very little has been done to satisfy the technical and operational requirements necessary for allowing the unfettered access to civil airspace, domestically and internationally, projected to be needed by the U.S. DoD and the civil/commercial market. Although standard organizations like RTCA and EUROCAE are focusing on the problem,

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

the current lack of specific UAS standards is clearly slowing the progress of the defense and commercial development of this exploding technology. In addition, as it relates to SAA, the impact of the lack of an SAA-focused “program of record” within one or all of the federal agencies and the associated funding that will be required and/or the lack of existing technical capability to move toward an autonomous solution believed by many to be necessary for complete, unfettered access to the national airspace appears to be slowing progress. Regardless, it looks like 2013 is the turning point year as roadmaps have been published for integration of unmanned aviation into the national airspace.

This section of the handbook introduces the reader to those sensing, processing, and control-enabling technologies that constitute the “sense-detect-and-avoid” technical foundations. Requirements, technologies, and future trends in sense, detect, and avoid are covered thoroughly.

► *Development of a Regulatory Safety Baseline for UAS Sense and Avoid* by Oztekin and Wever presents a systems-level approach to analyze the safety impact of introducing UAS into the NAS. Utilizing Safety Management Systems (SMS) principles and the existing regulatory structure, a methodology is outlined to determine a regulatory safety baseline for a specific area of interest regarding a new aviation technology, such as UAS Sense and Avoid (SAA). The presented methodology is then employed to determine a baseline set of hazards and causal factors for the UAS Sense and Avoid problem domain and associated regulatory risk controls.

► *Achieving Sense and Avoid for Unmanned Aircraft Systems: Assessing the Gaps for Science and Research* by Davis and Cook focuses on the efforts by the Office of the Under Secretary of Defense (OUSD) for Acquisition, Technology, and Logistics (AT&L) to make progress in solving the SAA problem by the establishment of the SAA Science and Research Panel (SARP) in February 2011. The SARP consists of a panel of experts in technologies necessary to provide UAS with the ability to sense and avoid other aircraft. The panel’s primary purpose is to promote partnerships between the U.S. DoD and the broader academic and scientific community on UAS NAS Integration science and research initiatives in order to identify open questions and challenges in research and science efforts that must be addressed to provide UAS with an effectual SAA capability. Further, this chapter describes the methods of identifying and closing the science and research gaps that have been identified to date.

► *Automatic Traffic Alert and Collision Avoidance System (TCAS) Onboard UAS* by Meyer, Gottken, Vernaleken, and Scharer reports on the SAA capabilities of the Barracuda UAS demonstrator, on which Cassidian installed a cooperative collision avoidance system based on Traffic Alert and Collision Avoidance System (TCAS). The demonstration is intended as a risk mitigation measure to support future UAS development efforts. The demonstration steps involve laboratory tests with real equipment in a closed-loop demonstration and the automatic execution of a collision avoidance maneuver in-flight. The in-flight demonstration takes place in segregated air space involving the Barracuda demonstrator platform and a real intruder. The chapter outlines the setup of the cooperative collision avoidance

system demonstrator, including the interface with the flight control system of the Barracuda and the control authority granted to the cooperative collision avoidance system demonstrator. It summarizes the Human-Machine Interface (HMI) layout and situation awareness provided to the UAS operator and illustrates how the potential loss of data link connection is covered by the system design. The development process is detailed, including requirements definition, validation, and verification steps as well as hardware-in-the-loop demonstration in support of the in-flight demonstration. It illustrates the demonstration scenarios utilized in the flight demonstration and the results achieved. Conclusions are drawn regarding the potential of such cooperative collision avoidance systems based on TCAS, the operational implications, and the certification aspects to be solved.

► *Test Flights to Demonstrate Effectiveness of a Ground-Based Detect and Avoid Integration Concept* by Udovic, de Jong, and Vielhauer presents and reports on the VUSIL program (*Validierung von UAS zur Integration in den Luftraum*) that has developed a concept for integrating UAS into uncontrolled nonsegregated airspace, which is also used by manned civil aviation. The surrounding traffic is indicated on a display in the pilot ground control station (GCS), based on radar data provided by the Air Navigation Service Provider (ANSP). In addition, the pilot is supported by means of a “detect and avoid” tool in detecting potential conflicts. This tool shows conflicts and possible avoidance maneuvers on a separate display. To validate the integration concept, two flight campaigns with two different UAS (helicopter and fixed-wing UAS) and several intruders were conducted. In the first flight campaign, the intruders for the real flying UAS were simulated. In the second flight campaign, the intruder was a manned helicopter. The concept has been validated by the results of both flight campaigns and by a safety analysis. The latter has addressed demonstrating the equivalent level of safety (ELOS) of “see and avoid” in manned aviation and “detect and avoid” as developed in the VUSIL program.

► *Scalable RADAR-Based Sense-and-Avoid System for Unmanned Aircraft* by Moses, Rutherford, and Valavanis presents one possible solution for addressing the midair collision problem in addition to increasing the levels of autonomy of UAS beyond waypoint navigation to include preemptive sensor-based collision avoidance. The proposed solution goes beyond the current state of the art by demonstrating the feasibility and providing an example of a scalable, self-contained, RADAR-based, collision avoidance system. The RADAR hardware and collision avoidance algorithms are presented in detail. The technology described herein can be made suitable for use on a miniature (maximum takeoff weight <10 kg) UAS platform. This is of paramount importance as the miniature UAS field has the lowest barriers to entry (acquisition and operating costs) and consequently represents the most rapidly increasing class of UAS.

► *Assessment of Detect and Avoid Solutions for Use of Unmanned Aircraft Systems in Nonsegregated Airspace* by Verstraeten, Stuij, and Birgelen summarizes and assesses the most promising candidate solutions for Detect and Avoid (DAA) for use of UAS in nonsegregated airspace against a set of DAA requirements. A DAA solution is the combination of the sensor suite, the avoidance algorithms, and the method of operation. The functions of a DAA solution are collision avoidance and,

in uncontrolled airspace, separation provision. The chapter focuses on the detection of conflicting traffic by the sensor suite. The sensor suite needs to be able to detect different classes of conflicting traffic in varying environments. Requirements for DAA solutions have to be set, and DAA systems must then be developed to meet these requirements. ATM requirements for use of UAS in nonsegregated airspace have been drafted by EUROCONTROL, including a subset of requirements that are directly or indirectly of influence on the sensor suite used to DAA other traffic. This subset is used as baseline for the establishment of DAA requirements and expanded by the development of five additional generic requirements based on the main tasks of a DAA solution: (1) detection of other traffic, (2) tracking of other traffic, (3) assessing if there is a conflict, (4) determining which evasive maneuver is to be executed, and (5) executing the selected maneuver. Five candidate solutions are assessed against the requirements: a noncooperative solution, a cooperative solution, and three solutions mixing cooperative and noncooperative sensors. Overall it is concluded that it is a great challenge to develop a collision avoidance solution for UAS with a satisfactory level of safety. It is even more difficult to develop a DAA system that is capable of both collision avoidance and separation provision.

Collectively, this section informs but also alerts the reader about challenges that need to be overcome in order for UAS to function safely in the global airspace system. It is expected that standards will soon be introduced that will accelerate the development of sense, detect, and avoid systems.

Ahmet Oztekin and Rombout Weaver

Contents

75.1 Introduction 1818

75.2 Defining a Safety Baseline 1819

75.3 Challenge with the UAS 1820

75.4 Regulatory-based Causal Factor Framework 1821

75.5 Approach to Construction of a Safety Baseline for SAA 1823

75.6 Causal Model for Midair Collision..... 1825

75.7 Approach to Identifying Regulatory Risk Controls for SAA 1828

75.8 Analysis of Regulatory Risk Controls 1831

75.9 Concluding Remarks..... 1837

References 1838

Abstract

Unmanned Aircraft Systems (UAS) emerge as a viable, operational technology for potential civil and commercial applications in the National Airspace System (NAS). Although this new type of technology presents great potential, it also introduces a need for a thorough inquiry into its safety impact on the NAS. This study presents a systems-level approach to analyze the safety impact of introducing a new technology, such as UAS, into the NAS. Utilizing Safety Management Systems (SMS) principles and the existing regulatory structure, it outlines a methodology to determine a regulatory safety baseline for a specific area of interest regarding a new aviation technology, such as UAS Sense and

A. Oztekin (✉)
Hi-Tec Systems, Inc., FAA William J. Hughes Technical Center, Atlantic City, NJ, USA
e-mail: ahmet.ctr.oztekin@faa.gov

R. Weaver
NLR Air Transport Safety Institute (NLR-ATSI), Amsterdam, The Netherlands
e-mail: Rombout.Weaver@nlr-atsi.nl

Avoid. The proposed methodology is then employed to determine a baseline set of hazards and causal factors for the UAS Sense and Avoid problem domain and associated regulatory risk controls.

75.1 Introduction

Unmanned Aircraft Systems (UAS) present great potential for civil and commercial applications in non-segregated airspace. Unrestricted UAS access into the National Airspace System (NAS) of the United States requires a thorough examination of its safety impact on the current operations in the NAS. In addition, a lack of regulatory guidance is considered an obstacle against achieving the full potential that UAS has to offer (FAA Flight Plan 2009–2013; Weibela and Hansman 2005). Recognizing the need for regulations and guidance material, aviation regulators initiated efforts to develop policies and establish requirements, procedures, and standards that will support UAS technology development and certification to enable safe operations of UAS. In the United States, Federal Aviation Administration (FAA) is working closely with the UAS community through RTCA Special Committee 203 (SC-203) to define the Minimum Aviation System Performance Standards (MASPS). Similarly, EUROCAE Working Group 73 (WG-73) is coordinating European efforts to deliver standards and guidance that will ensure the safety and reliability of unmanned aircraft missions operating in non-segregated airspace (EUROCAE 2009). These efforts are also being informally coordinated to facilitate harmonization (RTCA 2010).

The integration of UAS into the NAS presents various unique challenges, which will require novel and mostly platform-specific technological solutions. However, it can be argued that demonstration of airworthiness of these technologies will not present the only barrier to the introduction of UAS in the NAS. The difficulty that UAS is currently facing arises in obtaining authorization to enter and use civil airspace. This originates from the legitimate concern that unmanned aircraft may collide with other aircraft. Given the consensus that there will be no dedicated airspace for UAS operations, some authorities place the primary role of avoiding any collision between UAS and manned aircraft solely to the UAS. For all practical purposes, the UAS is, therefore, expected to have full responsibility to sense other aircraft and take effective evasive action. In this context, “see and avoid” or “sense and avoid” (SAA) emerges as one of the areas, which introduces new challenges and technologies compared to manned aviation and raises attention of regulators as well as the UAS manufacturers and future operators.

The body of current research projects on sense and avoid in the UAS domain mainly focuses on technology development and demonstration to provide a portfolio of workable technological solutions for the see and avoid concept. As compared to technology development, research on UAS safety risk analysis with an emphasis on SAA is still in its infancy. Most current UAS safety studies perform the risk analysis at a very detailed level based, on limited event or occurrence data. Examples of such research are preliminary functional hazard assessments (Hayhurst et al.

2007), event-based safety models of UAS (Weibel 2005), and simulation-based encounter models (Kochenderfer et al. 2008). However, a systems-level safety analysis focusing on the regulatory aspects of the SAA concept for UAS operations with an emphasis on future NAS access is lacking.

This study outlines a systems-level safety risk analysis framework for the SAA concept. Within the context of this study, a system-level perspective refers to looking at the air transport system as a whole from a high-level of abstraction as a system, or a system of subsystems. Thus, NAS may be considered as the system and UAS as a subsystem. In particular, the proposed framework presents a novel regulatory-based and integrated approach to understand hazards associated with midair collision risk and SAA and provides an analysis of current regulatory controls related to this topic. Utilizing Safety Management Systems (SMS) principles, the proposed framework establishes a systems-level safety analysis approach based on the FAA regulatory requirements to support the safe integration of UAS into the NAS with a particular focus on “sense and avoid.” The framework is intended to provide insight in risks and risk controls in current regulations when integrating new and complex technologies into the NAS while meeting the FAA’s SMS mandates. This study divides potential operations in the NAS into two main subgroups: flights conducted under Visual Flight Rules (VFR) and Instrument Flight Rules (IFR). The analysis and results presented here were developed for IFR operations.

In this study, the terms “hazard” and “causal factor” are used within the following context: A hazard is a condition, object, or activity with the potential of causing injuries to personnel, damage to equipment or structures, loss of material, or reduction of ability to perform a prescribed function. Hazards occur usually due to several causal factors. In general, hazards are considered at a higher level of abstraction, whereas causal factors are of a more detailed level.

The research presented here is based on the concepts and ideas that have been introduced in (Luxhøj et al. 2009, 2010; Oztekin and Luxhøj 2008, 2009). The next section provides background information on these concepts that are instrumental to the analysis, results, and discussion presented. Consequently, the proposed approach is illustrated by using a causal model for midair collision as the basis to develop a safety baseline for the SAA concept. Finally, an analysis on the SAA safety baseline is presented along with some concluding remarks.

75.2 Defining a Safety Baseline

In order to regulate a new technology that is to operate within an already-established and well-regulated infrastructure, such as NAS, without risking stifling its potential, one needs to understand existing safety criteria required to achieve the level of safety associated with the current operations. Within the context of the NAS, the existing safety criteria are the applicable aviation rules and regulations governing everyday manned flight and flight support and management operations of commercial or noncommercial nature. In this context, aviation rules and regulations act as controls

against potential risks and provide a baseline for safe operations in the NAS. More specifically, regulations are risk controls that constitute a safety baseline for all operations in the NAS. All aircraft operating in the NAS have to satisfy requirements set by Title 14 Code of Federal Aviation Regulations (14 CFR) and follow supporting guidance material. 14 CFR provides the risk controls for safe operations and establishes a baseline for all prospective operation in the NAS. For UAS technology and operations in nonsegregated airspace, current regulations (14CFR) apply. Thus, understanding the risk controls as defined by 14 CFR and outlining a baseline set of hazards and underlying causal factors that existing regulations control lie at the crux of the regulatory safety baseline concept (Oztekin and Lee 2011; Oztekin et al. 2011).

75.3 Challenge with the UAS

Limited availability of data on emergent nature of UAS operations introduces a challenge for the safety analysis and assessment of UAS operations. Conventional quantitative safety risk analysis techniques, essentially event-driven and largely built upon past experience and vast amount of historical data, may not provide adequate information for risk controls necessary for emerging technologies such as UAS. In the absence of operational data, it becomes very difficult to perform a systems-level safety analysis of UAS using conventional quantitative safety analysis methodologies. However, this situation can be considered as typical for any new technology with a limited accumulation of historic operational data. In this context, it is argued that a new approach may add valuable insight to understand the safety impact of emerging UAS operations on NAS. This new approach should not rely on historic data about the new technology, therefore would not be hindered by the lack of it. Furthermore, it should also assume a systems-level perspective while performing the safety analysis. Thus, a successful attempt to understand the safety impact of emerging UAS operations of civil/commercial nature can be achieved through a higher systems-level approach, which takes into account the problem domain as a whole. In this case, the problem domain in question is the NAS and it should be treated as a single complex system. Subsystems, such as Air Traffic Control (ATC), Airmen, Aircraft, Flight Operations, and Airspace constituting the NAS are interdependent and their interactions determine safety that permeates the whole system and defines minimum mandatory safety requirements for the NAS. These minimum set of requirements constitute a mandatory baseline for conducting safe operations in the NAS. A systematic approach for the identification of such a safety baseline will provide guidance to understand systems-level safety impact of a new technology, such as UAS, onto the NAS. Utilizing the safety baseline concept, Regulatory-based Causal Factor Framework (RCFF) (Luxhøj et al. 2009, 2010; Oztekin and Luxhøj 2008, 2009; Oztekin and Lee 2011; Oztekin et al. 2011, 2012), provides such a systematic approach. Although RCFF is a new and intuitive approach, one should make it clear that it does not replace but complements existing qualitative and quantitative methods by recognizing existing rules and regulations

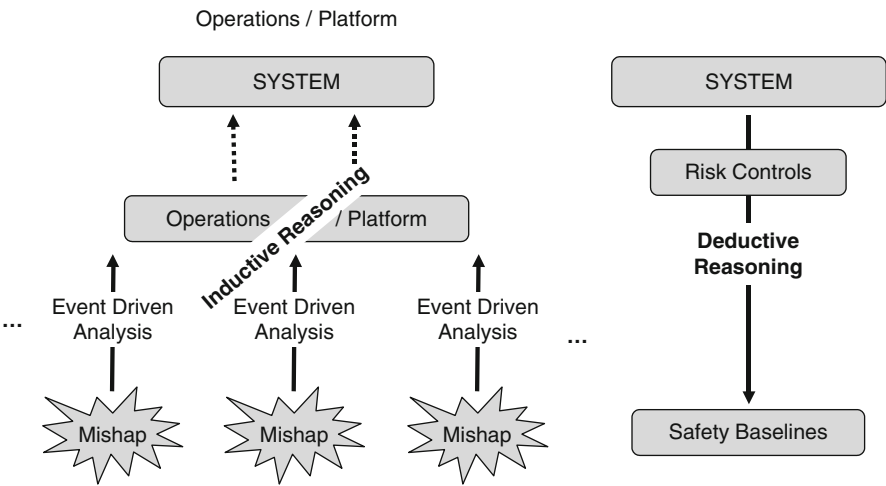


Fig. 75.1 Inductive reasoning versus deductive reasoning

covering the problem domain as key components of a system-level qualitative safety risk analysis framework.

Data-centric methodologies benefit from *inductive reasoning* when modeling the problem domain and the system in question. Although inductive reasoning has been successfully employed for data-rich systems for which extensive collections of case studies exist, due to similar reasoning outlined above, inductive frameworks may not be a good fit to understand and model new technology with limited accident/incident data. In this context, this study adopts *deductive reasoning* to understand the problem. Figure 75.1 illustrates deductive approach as compared to inductive reasoning. Deductive reasoning is a top-down approach, which puts emphasis on modeling the system based on its higher and more general components. When applied to the area of safety analysis, contrary to inductive approach where the analysis would be based on individual accident/incident cases and related data, a deductive approach will study the system as a whole and focus on its higher-level components and their interactions. Thus, a deductive approach will concentrate on understanding the safety minimums and, using an engineering term, determine the boundary conditions for conducting safe operations within the system.

75.4 Regulatory-based Causal Factor Framework

The study presented in this chapter utilizes concepts introduced by the RCFF. In very broad terms, RCFF is a qualitative, systems-level approach to safety assessment based on deductive reasoning to construct a safety baseline for operations in the

NAS in the United States. The basic concept relies on two fundamental premises: Aviation rules and regulations “Title 14 Code of Federal Regulations (14 CFR)” provide minimum mandatory requirements (i.e., risk controls) for safe operations in the NAS and various unrelated regulations interact to provide risk controls for hazards.

14 CFR can be considered as the culmination of efforts by the larger aviation community to provide an inherent minimum level of safety for every single operation to be conducted within the NAS. This notion of minimum safety is outlined as rules and requirements by the 14 CFR, and every aircraft in the NAS has to operate above the minimums of this mandatory safety baseline. In this sense, regulations act as minimum controls for potential risks associated with operating in the NAS.

However, individual regulations do not operate in a vacuum. When a specific aircraft or operation is concerned, a diverse collection of rules regulating different areas of the NAS interact to provide minimum requirements for safety as they apply to the specifics of the operation/aircraft in question. For example, issues related to certification of aircraft, aircraft engine, or propeller are regulated by 14 CFR Part 21, whereas airworthiness standards for the aircraft and its components are outlined in Parts 21–33. Subchapter D of 14 CFR focuses on the issues of certification and training of airmen and Subchapter E defines and partitions airspace, within which the proposed operation is set to take place. Thus, every single operation in the NAS, whether that particular type of operation has been conducted routinely for many years or it is the implementation of a new technology, is enveloped by a mandatory minimum safety baseline created collectively by various interacting rules regulating potential sources of various different hazards.

The notion of interactions between various parts of the 14 CFR to provide a minimum mandatory safety risk controls is a simple yet powerful idea, which brings forth a new approach to understand and study safety in aviation. This intuitive idea, in fact, borrows from the fundamental principle of the interdisciplinary field Systems Analysis. Formally, systems analysis is the dissection of a system into its component pieces for purposes of studying how those component pieces interact. In complex systems such as NAS, safety is the product of these interactions. However, a closer look at various current research efforts on UAS integration in the NAS reveals that such studies rarely explore potential interactions between their respective area of interest and various other components of the NAS, in a systematic fashion.

The regulatory-based approach of RCFF takes cues from FAA’s own Safety Management System (SMS) process. FAA Policy Document on SMS Guidance (FAA 2008) states that

...regulations will serve as risk control, if correctly applied in the context of the unique operational environments of service providers. Rule making process therefore should apply the concepts of safety risk management ...They [regulations] should identify hazards ... Compliance with the regulations would thus move beyond viewing them on as administrative requirements and into an environment where compliance entails effective control of clearly identified hazards. This would enhance the value of regulations as effective instruments of safety management. Regulations and subsequent oversight activities must be part of a strategy of risk control.

This understanding of regulation's role coincides with the fundamental concepts that RCFF is built upon; namely, use regulations as risk controls, identify hazards based on risk controls, identify causal factors underlying hazards, determine potential interactions between causal factors (thus between risk controls). Within the context outlined in the FAA SMS Guidance, RCFF can also be used as part of an exploratory risk-based rule-making process as a future research initiative, where the impact of the current regulations as risk controls are evaluated on the safety baseline and shortcomings are identified and corrected.

RCFF adopts a deductive, top-down approach to identify systems-level hazards and associated causal factors using regulations (i.e., 14 CFR) as risk controls. This approach is especially a good fit for providing a system-level qualitative safety risk analysis of emerging technologies, such as UAS, where limited availability of case data poses a challenge. RCFF also proposes a methodology to determine connections between potentially related causal factors, thereby creating an interlinked safety baseline. Ultimately, the RCFF safety baseline can be explored to understand the interactions between causal factors, as well as the dependencies between regulations (i.e., risk controls).

The outcome of the RCFF process is the safety baseline: hazards, causal factors, and regulations as risk controls. The context and scope of the safety baseline is determined by the set of regulations included in the RCFF analysis. Hazards and causal factors are identified based on risk controls outlined by these regulations. The scope of an RCFF analysis and the extent of the resulting safety baseline can be adjusted both depth-wise and breadth-wise in terms of detail and coverage.

Conceptually, the RCFF hierarchy closely follows the current regulatory structure. At the very top of this hierarchy, covering the entire NAS, Federal Aviation Regulations (14 CFR) provide minimum risk controls for safe operations. Thus, an RCFF top-down modeling process starts with regulations, or rather, it accepts regulations as input. However, risk controls can also be found beyond 14CFR: orders, technical manuals, guidance material, even prior safety studies are among the source materials that can be used as risk controls to initiate the process for an RCFF-based analysis.

The RCFF hierarchy including risk controls, hazards, causal factors, and linkages between them are stored in a database. A detailed discussion on the methodology used to construct an RCFF hierarchy based on the existing set of regulations and to populate the RCFF database is provided in Oztekin and Lee (2011) as part of a proof-of-concept study, where a potential utilization of the database is outlined. The high-level implementation of RCFF utilizes 14 CFR Parts as the basis to develop its hierarchy and the resulting system-level safety baseline (Oztekin and Lee 2011).

75.5 Approach to Construction of a Safety Baseline for SAA

Current regulations prescribe generic risk controls against midair collision and near midair collision risk. Under provisions that regulate operations near other aircraft, FAR Sect. 91.111 (b) states that “no person may operate an aircraft so close to

another aircraft as to create a collision hazard.” Additionally, FAR Sect. 91.113 (b) quoted below, explicitly uses language that includes the terms “see and avoid” and “well clear”: “When weather conditions permit, regardless of whether an operation is conducted under instrument flight rules or visual flight rules, vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft. When a rule of this section gives another aircraft the right-of-way, the pilot shall give way to that aircraft and may not pass over, under, or ahead of it unless well clear.” Obviously, the need for UAS to comply with the SAA concept in the NAS extends well beyond these two 14 CFR sections and a more detailed analysis of regulations with a particular emphasis on current risk controls for hazards related to (near) midair collision is needed.

See and Avoid can be used to assure separation from other aircraft and to avoid collisions in case separation failed. The SAA concept can be divided into two areas, namely, separation assurance and collision avoidance (Lacher et al. 2008). Separation assurance covers topics ranging from airspace structure and procedures to onboard alert systems such Traffic Collision Avoidance System (TCAS). On the other hand, collision avoidance entails the act of sensing and avoiding traffic conflicts or executing the collision avoidance maneuver once all preceding protective layers fail to provide the separation required. Most of the UAS SAA-related research focus on technology development that falls within the domain of collision avoidance. When it comes to understanding the safety impact of various interacting components of the NAS on the UAS SAA, there is more to it than sensor technology and algorithm development. Therefore a more integrated approach is needed to conduct a safety analysis of UAS SAA.

The objective of this study is to define a safety baseline for (near) midair collisions and the SAA concept. In this study, the RCFF concept has been applied to identify risk controls that the current regulations provide to mitigate the hazards and causal factors related with (near) midair collisions and the SAA concept. The result is a set of risk controls derived from existing regulations that will not only be applicable to manned operations in the NAS but will also provide a minimum, but possibly not sufficient set of risk controls to mitigate (near) midair collision risk for potential UAS operations. Operations conducted under instrument flight rules (IFR) and visual flight rules (VFR) have to be considered to develop a minimum mandatory baseline for SAA which applies to a wide range of traffic encounters under all operational environments. The current study is focused on IFR operations only. It is planned to expand the scope of the study by including VFR operations as part of future research.

In this context, the study presented in this chapter is composed of the following three phases:

1. Identify a baseline set of hazards and underlying causal factors for (near) midair collisions and the SAA concept. A causal model developed and validated specifically for this topic is used as the starting point to identify a baseline set of regulatory risk controls.
2. Identify risk controls for the causal factors. The risk controls are derived from existing regulations. The risk controls along with the hazards and causal factors constitute the safety baseline for preventing (near) midair collisions;

3. Perform an analysis of existing regulatory controls. The objective of the analysis is to quantify and understand the coverage that the identified regulatory risk controls provide to potentially mitigate the hazards and causal factors associated with the SAA safety baseline. Such an analysis also focuses on interactions between different domains of regulatory controls and helps to determine the coverage or gaps in regulatory material concerning regulating the new system, operation, etc.

Since the RCFF exists conceptually and not yet as a full-scale application, it is not possible to apply RCFF directly to the SAA concept, as the set of associated causal factors, hazards, and risk controls is not available yet. Instead, a detailed causal model developed specifically for the issue at hand provided a good starting point to identify a set of hazards and causal factors and associated regulatory controls that could form the safety baseline for the midair collision risk and SAA concept.

Considering that RCFF is a top-down framework, this “bottom-up” approach may seem inconsistent. However, the basic idea behind the RCFF is that regulations provide a set of risk controls for hazards (and causal factors); these regulations interact and their interactions can be identified through identifying dependencies between hazards (or between causal factors). The ultimate goal of RCFF is to determine risk controls and their interactions for the problem domain in question. Using existing knowledge about hazards and causal factors (i.e., a causal model) to identify the regulatory risk controls and their dependencies is still consistent with the RCFF concept.

75.6 Causal Model for Midair Collision

A causal model explains the functional and quantitative relationship between the various factors affecting risk in the Air Transport System (or NAS) or major parts of it. Generally speaking, such models allow the user to understand how, and how much, changes in a particular part of the ATS change the local as well as the overall safety risk of the ATS.

The hazards and causal factors in relation to (near) midair collision and SAA for IFR operations were identified by means of an existing causal model, which was developed as part of the Causal Model for Air Transport Safety (CATS) study (Ale et al. 2005; CATS 2008). The CATS study was a major effort sponsored by the Dutch government and developed by a consortium of parties including the NLR-Air Transport Safety Institute. The aim of CATS was to understand the causal factors underlying the risks of commercial air transport. It is an integrated quantitative causal model that can be used for safety risk analysis and assessment in civil air transport. The backbone of CATS consists of 33 generic accident scenarios. Each accident scenario was modeled as an Event Sequence Diagram (ESD), a flowchart which starts with an initiating event and progresses through pivotal event toward a set of possible outcomes (e.g., accident, incident, and continued flight). Each path through the flowchart is a scenario. Along each path, pivotal events are identified as

either occurring or not occurring. Fault Trees connect to the events in the ESD and represent the deeper, underlying causes of these events.

The causal model is composed of a qualitative as well as a quantitative part. The qualitative part is formed by the ESD and Fault Tree structure defining the accident scenario and causal pathways leading to different outcomes. In other words, event names and descriptions and the relationships between events, the model structure, constitute the qualitative model. The model was quantified using accident, incident, and occurrence data as well as expert judgments. It is crucial to maintain the context of the ESD and Fault Trees and avoid drastically revising language and model structure; otherwise the quantified model elements would no longer be valid.

This study utilizes (near) midair collision ((N)MAC) as the system-level hazard to define the SAA problem domain. (N)MAC as an event has been fully studied to understand underlying hazards and causal factors using available occurrence data from aviation safety databases. Analysis indicated that the SAA problem domain in the NAS can be decomposed into two main operational sub-domains, namely, operations conducted under instrument flight rules (IFR) and under visual flight rules (VFR). All potential scenarios need to be studied to fully cover the problem domain and develop a minimum mandatory baseline for SAA which applies to all encounters under all operational environments. The resulting safety baseline for SAA will not only be applicable to manned operations in the NAS but will also provide a minimum, but possibly not sufficient, set of risk controls to govern potential UAS-specific hazards.

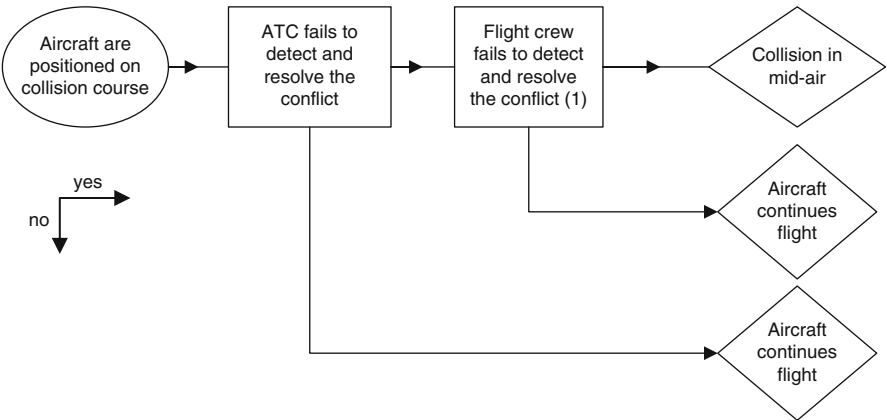
In this context, the study presented here is based on the MAC/NMAC encounters involving two aircraft-operated under IFR, thereby partially covering the operational domain of the NAS, as the current scope of this research.

The ESD for midair collision is one of the 33 generic accident scenarios modeled in CATS and describes generically hazards and causal factors in relation to (near) midair collision and SAA for IFR operations. The initiating event is “aircraft are positioned on collision course” and the end states are “collision in midair” or “aircraft continues flight.” The detailed specific or possible causes or contributing factors of the pivotal events in the ESD are added by Fault Trees underneath each pivotal event. The ESD models two layers of conflict detection and resolution: ATC and the flight crew (supported by, e.g., collision avoidance systems and SAA). The models were developed by a combination of retrospective and prospective analysis. The retrospective analysis consisted of a detailed and structured analysis of aviation accidents, which demonstrate typical accident patterns. The prospective analysis is based on engineering knowledge and aviation operational domain experts to identify potential hazards and hazardous combinations of causal factors that may have not (yet) resulted in an accident.

The ESD is representative for Part 121 operations with Part 25 aircraft and entails the encounter of two aircraft operating under IFR. It is composed of the initiating event and two pivotal events, and together with the associated Fault Trees, it has 29 intermediate events and 61 base events, see Fig. 75.2.

The ESD for midair collision is used as the basis to develop a minimum mandatory safety baseline for SAA. However since the scope of the midair collision

Accident type: mid-air collision.
Flight phases: initial climb, en-route and approach.
Initiating event: aircraft are positioned on collision course.



(1) This pivotal event includes the execution of ‘see-and-avoid’ principle and the response to a Traffic Collision Avoidance System alert.

Fig. 75.2 Event sequence diagram used to develop the components of the SAA safety baseline

model is limited to IFR, only the portion of the SAA safety baseline that covers IFR operations within the NAS is modeled by the study presented here.

The events constituting the Fault Tree supporting the ESD were originally defined using short descriptive texts which outline the intended scope of the event within the context of the accident scenario in question. The causal model elements were reviewed by a group of subject matter experts (SMEs) to identify causal factors and hazards that will constitute the safety baseline for SAA in the NAS. During the course of the review process, the original language used to describe the events was also revised by the SMEs so as to fit the terminology currently prevalent in the NAS. The review process resulted in 3 “system-level” hazards and 60 causal factors constituting the SAA safety baseline for IFR operations in the NAS.

The set of hazards and causal factors are generic and applicable to both manned and unmanned aviation operations. Some causal factors may have a minor different interpretation in case of unmanned aircraft, without affecting the cause-effect relationship. For example, some causal factors may refer to “pilot” which can be interpreted as “UAS operator” without changing its cause-effect relationship. However, the identified set of causal factors should be reviewed in more detail to identify potential missing causal factors related to UAS-specific operations and technology.

The original causal model for midair collision is composed of a qualitative as well as a quantitative part. The qualitative part is comprised of the ESD and Fault Tree structure defining the accident scenario and causal pathways leading to different outcomes. In other words, event names and descriptions and how they

are placed in the model structure constitute the qualitative model. The model was quantified using accident, incident, and occurrence data as well as expert judgments. Thus, the causal model for midair collision, as a whole, represents a certain probability distribution. However, if it becomes necessary to include additional UAS-specific hazards as part of a future study, associated Fault Trees may also need to be revised quantitatively resulting in a new probability distribution representing the revised causal model.

75.7 Approach to Identifying Regulatory Risk Controls for SAA

Once a baseline set of hazards and causal factors was determined from the causal model, the next step was the identification of existing regulations that provide potential controls to prevent and mitigate hazards and causal factors related to midair collision and SAA for IFR operations in the NAS.

The SAA safety baseline presents a simple hierarchical structure. The regulation that explicitly mentions SAA as a safety requirement for conducting operations in the NAS (i.e., FAR Sect. 91.113-b) is at the very top of this hierarchy. Hazards identified for SAA branch out from Sect. 91.113(b), and individual causal factors are listed for each hazard. In this context, risk controls for hazards are identified through individual causal factors. In other words, risk controls are identified for individual causal factors, thus their connections to hazards are indicated through causal factors. Regulations (i.e., 14 CFR) also present a hierarchical structure. Fourteen CFR is grouped into subchapters. Subchapters are partitioned into Parts and Parts into subparts. Subparts are divided into sections. Specificity of information that a regulation provides increases as one moves toward the next lower level in the regulatory hierarchy. Since the safety baseline is comprised of a collection of very detailed causal factors, pertaining risk controls should also present a level of detail that is comparable with the information content of the safety baseline. Thus, for each causal factor, specific 14 CFR sections were identified as potential risk controls. Notional representation of the SAA safety baseline hierarchy based on the structure of the NMAC ESD is illustrated in Fig. 75.3.

Due to resource constraints, this study limited the scope of the risk control identification to 13 FAR Parts representing three major subchapters of 14 CFR as sources for potential risk controls. The FAR Parts and corresponding subchapters included in this study are listed below:

Subchapter C – Aircraft:

Part 21 – Certification Procedures for Products and Parts

Part 23 – Airworthiness Standards: Normal, Utility, Acrobatic, and Commuter Category Airplanes

Part 25 – Airworthiness Standards: Transport Category Airplanes

Part 27 – Airworthiness Standards: Normal Category Rotorcraft

Part 33 – Airworthiness Standards: Aircraft Engines

Part 34 – Fuel Venting and Exhaust Emission Requirements for Turbine Engine Powered Airplanes

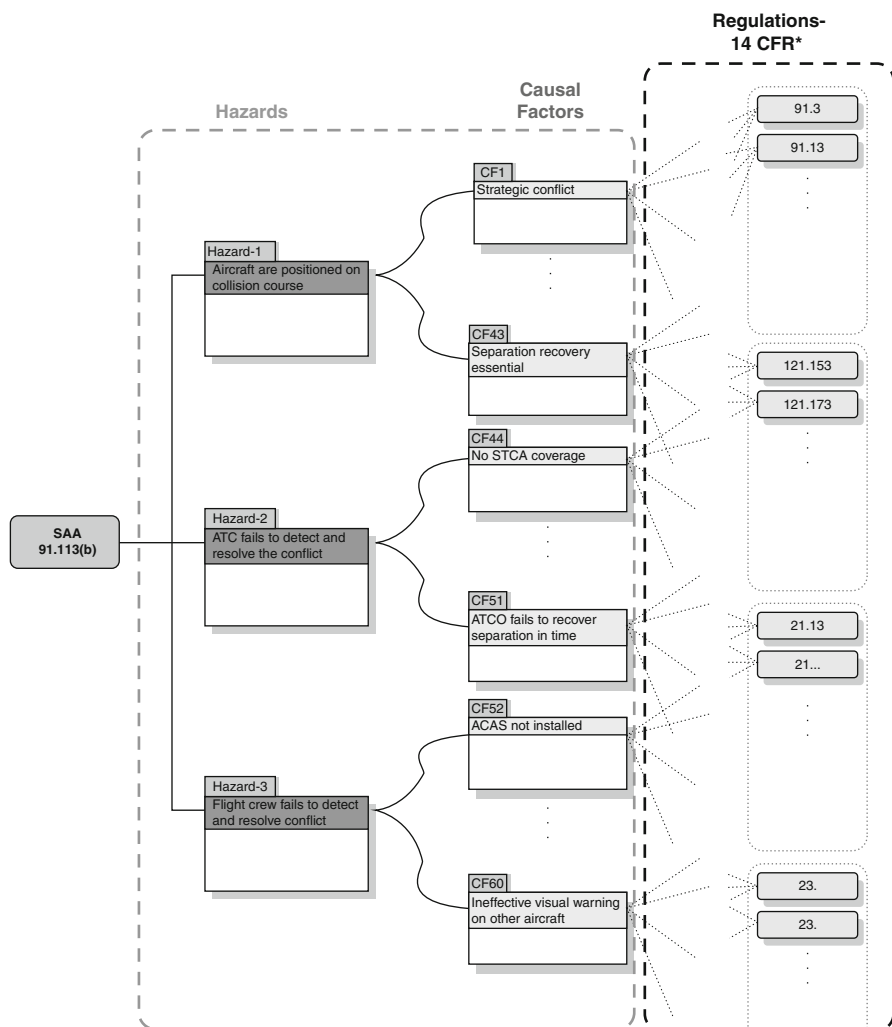


Fig. 75.3 Notional illustration of the SAA safety baseline

Part 43 – Maintenance, Preventive Maintenance, Rebuilding, And Alteration

Subchapter D – Airmen:

Part 61 – Certification: Pilots, Flight Instructors, and Ground Instructors

Part 65 – Certification: Airmen Other Than Flight Crewmembers

Subchapter F – Air Traffic and General Operations:

Part 91 – General Operating and Flight Rules

Subchapter G – Air Carriers and Operators for Compensation or Hire: Certification and Operations:

- Part 121 – Operating Requirements: Domestic, Flag, And Supplemental Operations
- Part 135 – Operating Requirements: Commuter and On Demand Operations and Rules Governing Persons OnBoard Such Aircraft
- Subchapter H – Schools and Other Certificated Agencies:*
- Part 145 – Repair Stations

Note that the Parts listed above are some of the most prominent rules regulating aviation safety. Thus they provide extensive coverage in terms of risk controls for the NAS.

The process of identifying section-level risk controls for the identified hazards and causal factors involved multiple knowledge elicitation sessions with Subject Matter Experts (SMEs), with extensive background and expertise on aviation-related rulemaking, regulatory oversight, as well as operations. The sessions were moderated in a structured manner with the participation of multiple SMEs and an aggregate approach was employed to determine potential risk controls for individual causal factors across the 13 FAR Parts mentioned before. The data sample provided below in Table 75.1 illustrates the type and content of the data that has been compiled as the result of the elicitation process to identify risk controls for the SAA safety baseline.

Table 75.1 Risk controls identified for CFs #4 and #5 of Hazard #1. Only risk controls from Parts 65, 91, 121, and 135 are shown

| Hazard #1 | | | | | | | |
|---|---|---|--------------------------------|------------------|--|---------------------------|--------------|
| Aircraft are positioned on collision course | | | | | | | |
| | | | Risk controls | | | | |
| # | Causal factor | Definition | Part 91 | Part 135 | Part 65 | Part 121 | Part 61 |
| 4 | Inadequate strategic surveillance picture | The radar picture is inadequate to allow the Planning Controller to identify the pre-tactical conflict, e.g., incomplete traffic picture, picture with overlapping labels, or too much traffic for the display system | N/A | 135.18 | N/A | 121.357, 121.356, 121.360 | N/A |
| 5 | Inadequate flight plan data | Flight plan data is inadequate to allow the Planning Controller to identify the pre-tactical conflict, e.g., incorrect flight plan, flight plan insufficient to identify conflicts, flight plan strips obtained too late, or aircraft not following flight plan | 91.173, 91.111, 91.113, 91.123 | 135.345, 135.347 | 65.31, 65.33, 65.35, 65.37, 65.39, 65.45, 65.49, 65.50 | 121.395 | 61.87, 61.93 |

75.8 Analysis of Regulatory Risk Controls

This section presents the analysis of risk controls that were identified according to the methodology outlined above. First, the identified risk controls are analyzed from a higher systems-level perspective and its interaction with the NAS as a whole. Next, individual causal factors are analyzed to achieve a more in-depth understanding of the SAA safety baseline and its interaction with the NAS.

The reader should bear in mind that the scope of this study includes only the 13 FAR Parts listed in the preceding section and this fact should be taken into consideration when reviewing the result presented here. Even though a set of regulations were identified in this study as applicable risk controls for a certain hazard or causal factor, there is still a need for further analysis to determine the extent to which these controls mitigate the risks associated with that hazard.

Potential system-level hazard sources underlying midair collision can be analyzed according to the Hazard Classification and Analysis System (HCAS) (Oztekin and Luxhøj 2008; Luxhøj et al. 2009, 2010). HCAS identifies four system-level hazard sources, namely, Aircraft, Operations, Airmen, and Environment, whose interactions impact any potential hazards within the context of the NAS. These system-level hazard sources are also in line with the 14 CFR subchapters.

The causal factors identified in the SAA safety baseline can be allocated to these four system-level hazard sources. Likewise the risk controls derived from the review of 13 FAR Parts can be categorized under the four main categories of interest: Aircraft, Airmen, Operations, and Certification. The distribution of risk controls over those categories is shown in Fig. 75.4 for the SAA safety baseline (IFR case only).

Note that Fig. 75.4 indicates a distribution over the total count of risk controls identified according to the methodology outlined in this chapter. Thus, 31 % of the risk controls identified originates from Part 91 corresponding to Air Traffic Control and General Operations Rules. Risk controls from Operational Certification–related Parts, namely, Parts 121 and 135, also provide 31 % of all the risk controls identified.

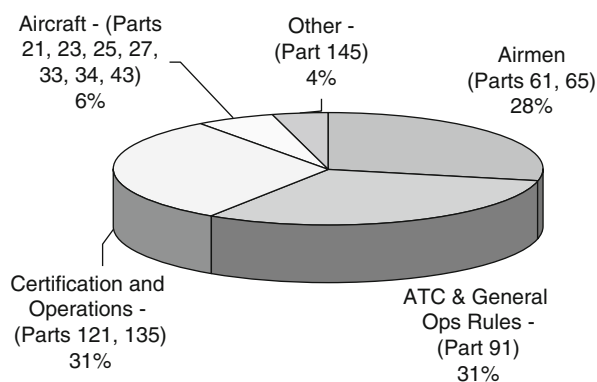


Fig. 75.4 Distribution of risk controls for the SAA safety baseline according to the four major categories of interest (IFR only)

While 28 % of the risk controls are Airmen related, only 6 % can be attributed to Aircraft certification–related regulations. Thus, one can conclude that, for the SAA safety baseline, existing rules regulating operational aspects of the NAS provide the majority of controls, whereas only a minority of the controls originate from current regulations governing aircraft certification. However, one should bear in mind that the analysis presented throughout this chapter is based only on the 13 Parts included in this study.

The SAA safety baseline includes three system-level hazards, namely, Loss of Separation, Failure by ATC, and Failure by Flight Crew. The hazard “Loss of Separation” represents the case that two aircraft are on a collision course/lost separation. The hazard “Failure by ATC” depicts a situation where, given that loss of separation occurs, ATC fails to detect and resolve the conflict. The hazard “Failure by Flight Crew” refers to a case where, given that Loss of Separation and Failure by ATC have occurred, the flight crew fails to detect and resolve the conflict. The distribution of risk controls identified for these three hazards over the categories Aircraft, Airmen, Operations, and Certification are shown in Figs. 75.5–75.7, respectively.

The system-level hazard “Loss of Separation” contains 43 causal factors for which 11 separate FAR Parts provide risk controls. Figure 75.5 indicates that 36 % of the risk controls for Loss of Separation are provided by sections of Part 91, which provides ATC and general operations rules for the NAS. Operation certification–related Parts (i.e., Parts 121 and 135) and Airmen-related Parts (61 and 65) each provide 26 % of the risk controls for Loss of Separation respectively. Considering how closely the hazard is associated with operational and ATC-related issues, this sort of a distribution is to be expected. Regulations such as Parts 21, 23, and 25 that govern aircraft and component certification provide a small portion of the risk controls (7 %) for the identified hazards/causal factors.

The system-level hazard “Failure by ATC” has eight causal factors. Three separate FAR Parts provide risk controls as shown in Fig. 75.6. Half of the risk controls are provided by Parts 61 and 65, whereas 44 % of the controls come from

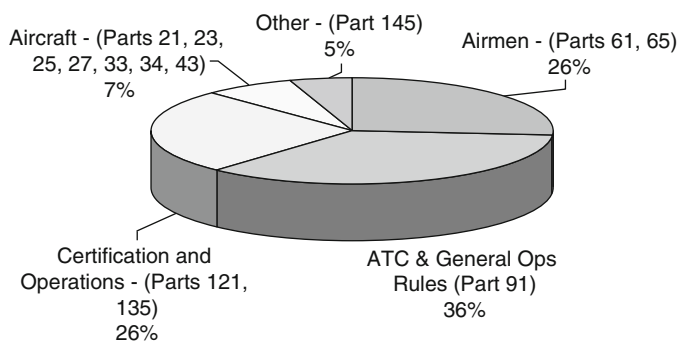


Fig. 75.5 Distribution of risk controls for the system-level hazard loss of separation

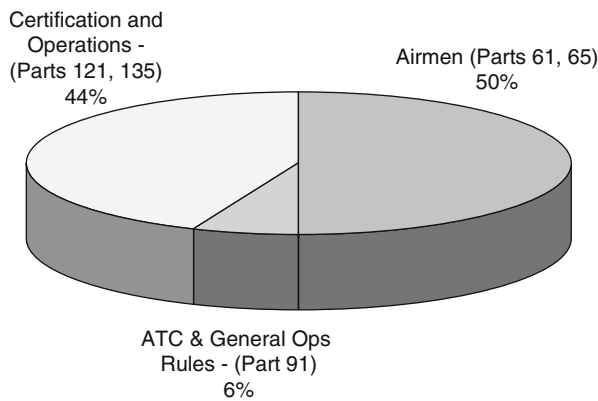


Fig. 75.6 Distribution of risk controls for the system-level hazard failure by ATC

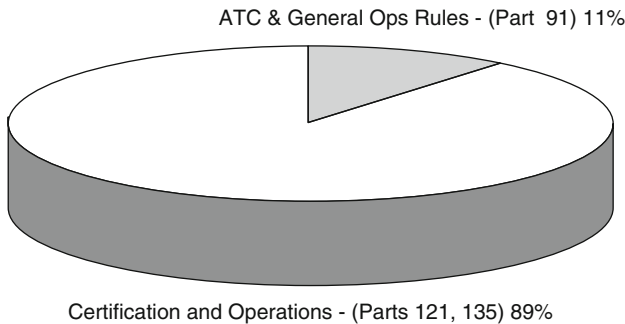


Fig. 75.7 Distribution of risk controls for the hazard failure by flight crew

operation certification–related regulations. Only 6 % of the controls originate from Part 91.

Figure 75.7 presents the distribution of the risk controls for the third and the last system-level hazard constituting the safety baseline: Failure by Flight Crew. Nine causal factors were identified under this hazard and three FAR Parts provide potential risk controls. Among the 13 FAR Parts included in the scope of this study, only Parts 91, 121, and 135 provide risk controls for the causal factors grouped under this hazard. Operation certification–related regulations, namely, Parts 121 and 135, present the overwhelming majority of the risk controls, whereas Part 91 provides only 11 % of the total controls for this hazard and underlying causal factors.

A more detailed analysis of the risk controls can also be performed at the level of causal factors constituting the SAA safety baseline for the IFR operations. Such an analysis is presented below with a particular emphasis on individual causal factors.

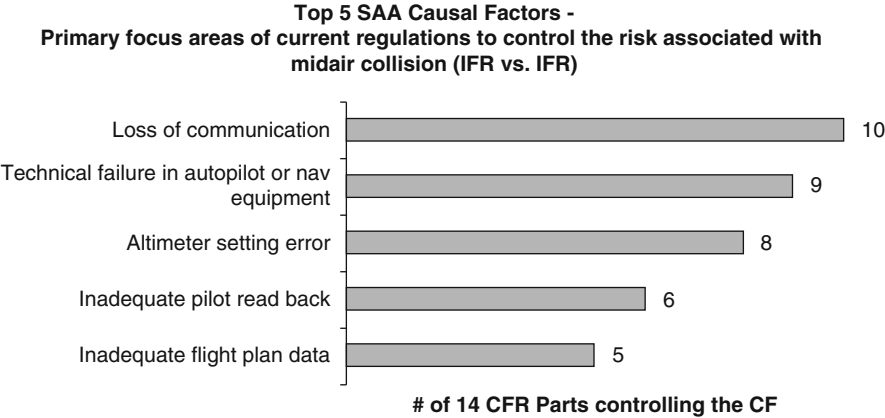


Fig. 75.8 Top causal factors in the SAA safety baseline with the highest number of 14 CFR sections acting as risk controls

Certain causal factors in the safety baseline receive relatively better coverage by 14 CFR sections acting as potential risk controls. The top five causal factors for which existing regulations provide the highest number of risk controls are presented in Fig. 75.8.

The causal factor “Loss of Communication” is defined as “communication between ATC and pilot is lost during a conflict in uncontrolled airspace due to technical failure or human error.” Forty-six regulatory sections are identified as potential risk controls for this factor. A closer look at these risk controls reveals that 10 different FAR Parts out of the 13 investigated by this study provide controls for mitigating or preventing potential risk associated with loss of communication. However, note that even though a set of regulations was identified as applicable risk controls for this causal factor, there is still a need for further analysis to determine that these controls fully mitigate the risks associated with the factor.

The lower end of the coverage spectrum, on the other hand, provides a glimpse of causal factors for which review regulations presents little to no coverage on the identified hazards/causal factors. Table 75.2 shows the causal factors that are potentially covered by only one or no section-level risk control.

Figure 75.7 and Table 75.2 help by illustrating the value of the proposed approach, which provides an analysis regarding to what extent the hazards and causal factors identified for the SAA are covered by existing regulations. Thus, the proposed approach facilitates the identification of gaps in the current regulations for a specific problem domain, in this case for SAA. These gaps indicate areas of potential risk within the SAA baseline, for which current regulations do not provide proper mitigation.

From the study results, the coverage and gaps in the current regulatory structure as potential risk controls for the identified hazards and causal factors related to midair collisions and SAA are presented in Figs. 75.9 and 75.10.

Table 75.2 SAA safety baseline causal factors with the least coverage in terms of risk controls that existing regulations provide

| Causal factor | Description | # of risk controls |
|--|---|--------------------|
| Other aircraft effectively invisible | The other aircraft cannot be seen from the cockpit. (other aircraft may not be visible due to glare, weather (rain), obstruction of view by wings, window frame, poor contrast, etc.) | 0 |
| Inadequate tactical surveillance picture | The radar picture is inadequate to allow the Tactical Controller to maintain separation in a plannable conflict, e.g., incomplete traffic picture or picture with overlapping labels | 0 |
| ATCO failure to recognize conflict | Tactical Controller obtains adequate flight information but fails to recognize the conflict | 1 |
| Conflict due to military traffic | Unauthorized penetration of civilian controlled airspace by military traffic | 1 |
| Weather induced level bust | Vertical deviation resulting from weather conditions | 1 |
| Level bust results in conflict | Given a level bust occurs, the aircraft has separation infringement with another aircraft | 1 |
| ACAS avoidance invalidated by other aircraft | ACAS avoidance action is canceled out by incorrect action from the other aircraft | 1 |
| Flight crew fail to observe visible aircraft in time | Pilots fail to observe visible aircraft in time to make avoidance action | 1 |
| Pilot fails to take avoidance action in time | Pilots fail to make appropriate avoidance action, having observed the other aircraft with sufficient time to take the necessary action | 1 |
| Visual avoidance invalidated by other aircraft | Pilot's response is canceled out by opposing maneuver from the other aircraft | 1 |
| Ineffective visual warning on other aircraft | Pilots on the conflicting aircraft fail to resolve the conflict using see and avoid techniques, given similar failure on the subject aircraft | 1 |

Figure 75.9 presents 14 CFR sections that act as potential risk controls for causal factor #16 “loss of communication between ATC and pilot.” Within the hierarchy of the SAA safety baseline, CF16 is associated with Hazard #1 “loss of separation.” A closer inspection of Fig. 75.10 shows that CF16 is controlled by four distinct grouping (i.e., subchapter) of 14 CFR Parts. In other words, to address hazards related to loss of communication between ATC, the applicability of the individual sections covering these four separate domains of interest need to be investigated to demonstrate that a baseline level of safety is achieved regardless of whether the operation is manned or unmanned. Not all the causal factors of the SAA baseline are

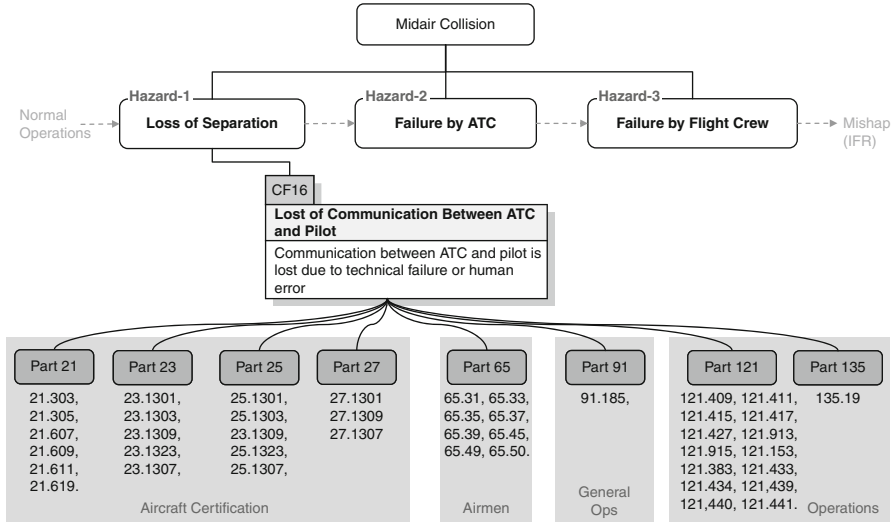


Fig. 75.9 Section level regulatory risk controls identified for CF 16 in the SAA safety baseline

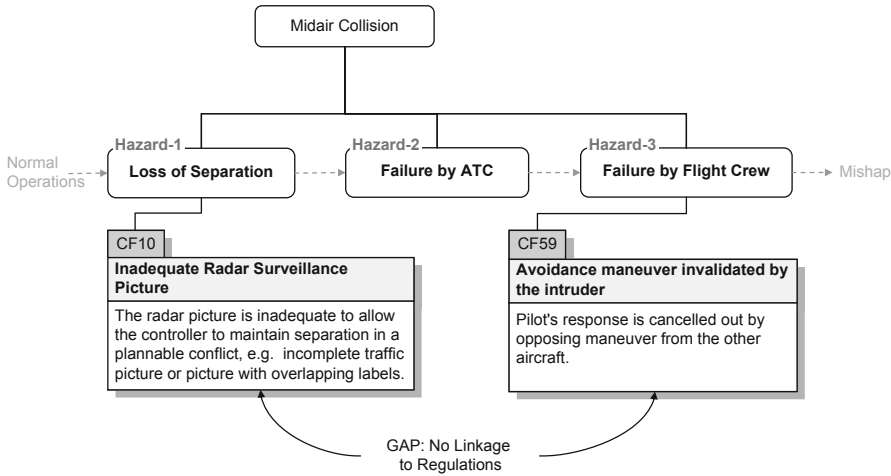


Fig. 75.10 Potential gap in the existing regulations. No risk controls were identified for CF 10 and CF 59 in the SAA safety baseline

covered by an existing regulation acting as risk control, indicating a potential gap in the existing regulatory structure. For example, Fig. 75.10 provides two causal factors in the IFR SAA baseline, for which no regulatory risk controls were identified from within the 13 FAR Parts included in this study.

75.9 Concluding Remarks

Ongoing efforts to develop rules and requirements for UAS SAA underlines the need to understand to what extent existing regulations cover SAA related hazards. In this context, this study presents a methodology that can be used to define a minimum set of risk controls based on current rules and regulations to control or mitigate hazards and causal factors related to a certain operation or technology. The value of the presented approach lies in the structured analysis to identify the existing regulatory coverage for the risks present in a certain domain. In particular, it provides an analysis regarding the extent to which hazards and causal factors identified for that domain are covered by existing regulations. Thus, the proposed approach facilitates the identification of gaps in the current regulations for a specific risk or domain.

The regulatory-based methodology (Regulatory-based Causal Factor Framework, RCFF) was applied to the domain of midair collision and See and Avoid (SAA). The study provided the identification of a set of hazards and underlying causal factors for (near) midair collision and SAA concept based on a causal model. The midair collision scenario was modeled as an Event Sequence Diagram with underlying Fault Trees that further detail the underlying causal factors within the context of IFR-only commercial air transport operations. The causal model provided an initial set of hazards and causal factors for the near midair collision and the SAA concept, which was consequently employed to identify a minimum set of risk controls for the SAA baseline using current rules and regulations.

The resulting SAA safety baseline is comprised of three hazards, 60 underlying causal factors, and a large number of applicable regulatory risk controls. While studying aviation regulations as potential risk controls for the identified hazards and causal factors, an initial set of FAR Parts representative of all major areas of interest in 14 CFR was selected and included in the scope of this study. Risk controls defined in the reviewed set of regulations were identified at the sections level. Finally, a systems-level analysis of risk controls was presented along with a more detailed look at the distributions with respect to the individual hazards and causal factors within the SAA safety baseline.

Although this study concentrates on the SAA problem domain, the proposed approach, coupled with the RCFF concept, is intended to be used for system-level safety analysis and assessment of other core areas of interest such as command, control, and communication for UAS integration into the NAS.

The outcome and the potential value of this study can be surmised as follows:

- It presents a structured approach to determine existing regulatory risk controls in the NAS for hazards related to a specific problem domain, e.g., a new technology. It is argued that the identified risk controls along with the set of hazards constitute a baseline for conducting safe operations within the context of that specific problem domain. This safety baseline establishes potential safety minimums that apply to all current and emergent operations, such as UAS.

Consequently, the safety baseline concept can be used to outline an initial set of safety provisions that apply to the UAS within the context a particular area of interest such as SAA in the NAS.

- It identifies a preliminary set of current risk controls for SAA in the NAS based on a selected group of aviation rules and regulations. The resulting analysis of the risk controls indicates to what extent the current regulations act as risk controls for hazards associated with the SAA in the NAS. This analysis can also be used to provide an understanding of potential gaps in the existing regulatory structure by identifying the hazards and underlying causal factors for which current regulations provide potentially no or limited mitigation. Since it is argued that the same set of risk controls apply to all operations in the NAS, they may also provide a roadmap for outlining safety provisions for the UAS within the context for which the risk controls were identified.

References

- B.J.M. Ale, *Development of Causal Model for Air Transport Safety*, ed. by K. Kolowrocki. Advances in Safety and Reliability ESREL 1 (Taylor and Francis, London, 2005), pp. 37–45
- CATS, The directorate general of civil aviation and maritime affairs, the Netherlands, Causal model for air transport safety, Final Report, May 2008
- Federal Aviation Administration, Safety management system guidance, Washington, DC, Order 8000.369, 30 September, 2008
- Federal Aviation Administration, Flight plan 2009–2013. Washington, DC, http://www.faa.gov/about/plans_reports/media/flight_plan_2009-2013.pdf. Accessed 8 March, 2011
- K.L. Hayhurst, J.M. Maddalon, P.S. Miner, G.N. Szatkowski, M.L. Ulrey, M.P. DeWalt, C.R. Spitzer, Preliminary considerations for classifying hazards of Unmanned Aircraft Systems. Washington, DC, NASA TM-2007-214539, February, 2007
- M.J. Kochenderfer, J.K. Kuchar, L.P. Espindle, J.L. Gertz, Preliminary uncorrelated encounter model of the National Airspace System, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts, Project Report CASSATT-1, ESC-TR-2007-056, 26 June, 2008
- A.R. Lacher, D.R. Maroney, A.D. Zeitlin, Unmanned aircraft collision avoidance: technology assessment and evaluation methods, MITRE Technical Paper, McLean, VA, July 2008. http://66.170.233.9/work/tech_papers/tech_papers.08/07_0095/. Accessed 8 March, 2011
- J. Luxhøj et al., Safety risk analysis of Unmanned Aircraft Systems integration into the National Airspace System: phase 1, Washington, DC, Final Report, DOT/FAA/AR-09/12, September, 2009
- J. Luxhøj et al., Safety risk analysis of Unmanned Aircraft Systems integration into the National Airspace System: phase 2, Washington, DC, Final Report, DOT/FAA/AR-10/12, September, 2010
- A. Oztekin, J.T. Luxhøj, Hazard, safety risk, and uncertainty modeling of the integration of Unmanned Aircraft Systems into the national airspace, in *26th Congress of International Council of the Aeronautical Sciences*, Anchorage, 14–19 September 2008
- A. Oztekin, J.T. Luxhøj, in *A Regulatory-Based Approach to Safety Analysis of Unmanned Aircraft Systems HCI International 2009 Conference Proceedings*. Lecture Notes in Computer Science (LNCS) series (Springer, San Diego, 2009)
- A. Oztekin, C. Flass, X. Lee, S. Keller, Development of a regulatory-based safety analysis framework for Unmanned Aircraft Systems, in *AIAA Unleashing Unmanned Systems Infotech@Aerospace Conference*, St. Louis, AIAA-2011-1418, 29–31 March 2011

- A. Oztekin, C. Flass, X. Lee, Development of a framework to determine a mandatory safety baseline for Unmanned Aircraft Systems. In *J. Intell. Robot. Syst.* **65**(1–4), 3–26 (2012). 10.1007/s10846-011-9578-0
- RTCA, Terms of Reference, Special Committee (SC) 203, Minimum performance standards for Unmanned Aircraft Systems, Washington, DC, Paper No. 065-10/PMC-790, 26 April, 2010
- The European Organization for Civil Aviation Equipment (EUROCAE), Work Group 73 (WG-73): Unmanned Aircraft Systems, Information Compendium, Lucerne, Switzerland, UAS-021.4, 24 April, 2009
- R.E. Weibel, R.J. Hansman, Safety considerations for operation of unmanned aerial vehicles in the National Airspace System, MIT International Center for Air Transportation, Cambridge, Report No. ICAT-2005-1, March 2005

Achieving Sense and Avoid for Unmanned Aircraft Systems: Assessing the Gaps for Science and Research

76

K. Douglas Davis and Stephen P. Cook

Contents

| | | |
|-------|---|------|
| 76.1 | Introduction | 1842 |
| 76.2 | Sense and Avoid Background | 1843 |
| 76.3 | Lessons Learned from TCAS | 1845 |
| 76.4 | Progress in SAA Development for Unmanned Aircraft | 1846 |
| 76.5 | The Sense and Avoid Science and Research Panel (SARP) | 1847 |
| 76.6 | Workshops | 1850 |
| 76.7 | Requests for Information | 1853 |
| 76.8 | Open Meetings | 1853 |
| 76.9 | Closing the Gaps | 1854 |
| 76.10 | Conclusions | 1854 |
| | References | 1855 |

Abstract

Sense and avoid (SAA) remains one of the key enabling capabilities required for the safe airspace integration of unmanned aircraft systems (UAS) into the global airspace system. Very little has been done to satisfy the technical and operational requirements necessary for allowing the unfettered access to civil airspace, domestically and internationally, projected to be needed by the Department of Defense and the civil/commercial market. Although standard organizations like RTCA and EUROCAE are focusing on the problem, the current lack of specific UAS standards is clearly slowing the progress of the defense and commercial development of this exploding technology. In addition, as it relates to SAA,

K.D. Davis (✉)

Physical Science Laboratory, New Mexico State University, Las Cruces, NM, USA

e-mail: ddavis@psl.nmsu.edu

S.P. Cook

MITRE Corporation, Bedford, MA, USA

e-mail: scook@mitre.org

the impact of the lack of a SAA focused “program of record” within one or all of the federal agencies and the associated funding that will be required and/or the lack of existing technical capability to move toward an autonomous solution believed by many to be necessary for complete, unfettered access to the NAS appears to be slowing progress.

The chapter entitled “Achieving Sense and Avoid for Unmanned Aircraft Systems: Assessing the Gaps for Science and Research” focuses on the efforts by the Office of the Under Secretary of Defense (OUSD) for Acquisition, Technology and Logistics (AT&L) to make progress in solving the SAA problem by the establishment of the SAA Science and Research Panel (SARP) in February 2011. The SARP consists of a panel of experts in technologies necessary to provide UAS with the ability to sense and avoid other aircraft. The panel’s primary purpose is to promote partnerships between the DoD and the broader academic and scientific community on UAS NAS Integration science and research initiatives in order to identify open questions and challenges in research and science efforts that must be addressed to provide UAS with an effectual SAA capability.

This chapter describes the methods of identifying and closing the science and research gaps that have been identified to date.

76.1 Introduction

“Airspace is the biggest hurdle that unmanned has today,” recently declared Mr. Dyke Weatherington, the Deputy Director for Unmanned Warfare Portfolio Systems Acquisition in the U.S. Office of the Under Secretary of Defense (OUSD) for Acquisition, Technology and Logistics (AT&L) (Magnuson 2012). One of the key enabling capabilities needed to overcome this hurdle of integrating unmanned aircraft systems (UAS) into the global airspace system is providing them with the ability to sense and avoid (SAA) other aircraft. Much remains to be done to satisfy the technical and operational requirements necessary for allowing UAS unfettered access to civil airspace, domestically and internationally, projected to be needed by the public aircraft (such as the U.S. Department of Defense) and the civil/commercial market. Although multiple international standards organizations are focusing on the problem, the current lack of specific UAS standard is holding back the development of UAS. In addition, as it relates to SAA, the fact that a best-of-breed SAA solution has yet to emerge for either civil or military use and/or the lack of existing technical capability to move toward an autonomous solution believed by many to be necessary for complete, unfettered access to the NAS appears to be slowing progress.

UAS flight hours have dramatically increased over the last decade. For example, U.S. Department of Defense (DoD) UAS flight hours (see Fig. 76.1) have increased almost exponentially over that time frame (United States Department of Defense). Although the primary user of UAS since that time has been the DoD, many of the other federal agencies, state agencies, and law enforcement organizations have seen

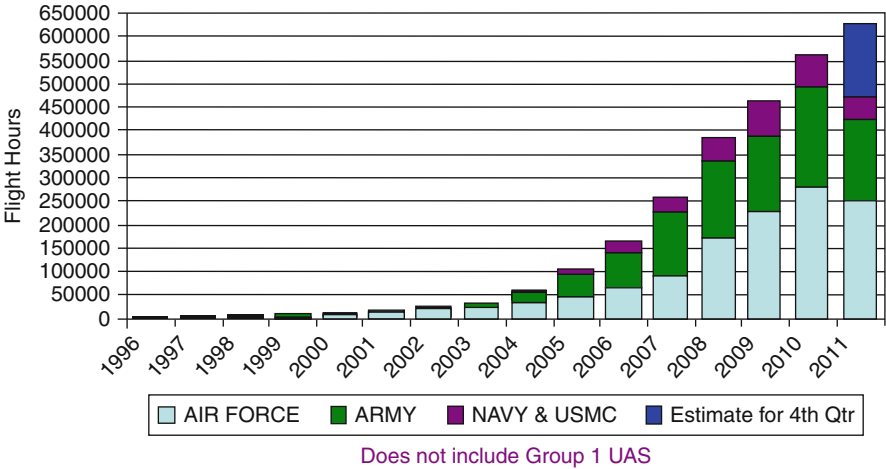


Fig. 76.1 DoD UAS flight hours per year (From [United States Department of Defense](#))

the intrinsic potential value that UAS can bring to their respective organizations. A June 2011 U.S. Congress Budget Office report stated that “Over the next 10 years, the Department of Defense (DoD) plans to purchase about 730 new medium-sized and large unmanned aircraft systems based on designs currently in operation, while also improving the unmanned aircraft already in service. By the Congressional Budget Office’s (CBO’s) estimates, completing the investments in systems for which there are detailed plans will require about \$36.9 billion through 2020” (United States Congressional Budget Office 2011). In addition, the commercial potential for UAS is beginning to gain traction, pushing the market value of these unique aircraft systems into the multibillion dollar range over the next decade. A 2012 Congressional Research Service report states that the last 10–15 years has featured UAS advances in “navigation, communications, materials, and other technologies” (Gertler 2012). However, while the development of the airframe and sensor capability has improved dramatically over the last decade, the same cannot be said for the development of SAA. This chapter will discuss a process of assessing gaps in science and research in SAA technologies in order to assist in overcoming the technical hurdles that confront achievement of SAA.

76.2 Sense and Avoid Background

Before going farther into this chapter, some discussion is needed to adequately define the challenge of assessing the science and research gaps for SAA. In today’s environment manned aviation has been the driving force for the evolution of aviation worldwide. All of the existing airworthiness, manufacturing, and operational guidance and regulations have been created with the desire to better protect pilots,

passengers onboard the aircraft, and people on the ground. These regulations have been constantly maturing since the beginning of aviation. The goal of this evolution has been to continue making the system safer.

Along come unmanned aircraft (UA) and much of the learning process over the past 100 years seem to be in need of recreation. Why? In addition to the unique airworthiness, spectrum, and security considerations with having the cockpit outside the aircraft, UAS have challenges in complying with several of the 14 Code of Federal Regulations, Part 91, General Operating Rules. Specifically, compliance with the following critical rules comes to mind from the Federal Aviation Regulations (Title 14 of the U.S. Code of Federal Regulations):

Sec. 91.111 Operating near other aircraft.

No person may operate an aircraft so close to another aircraft as to create a collision hazard. ...

Sec. 91.113 Right-of-way rules: Except water operations.

General. When weather conditions permit, regardless of whether an operation is conducted under instrument flight rules or visual flight rules, vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft. When a rule of this section gives another aircraft the right-of-way, the pilot shall give way to that aircraft and may not pass over, under, or ahead of it unless well clear ... (United States Code of Federal Regulations).

There are many other airspace rules with which UAS will need to show compliance for unfettered airspace access, like weather avoidance, cloud clearance, ATC clearance compliance, and terrain avoidance. Furthermore, these same issues and problems must be solved to fly outside of U.S. domestic airspace existing in all non-U.S. regulatory guidance issued by the International Civil Aviation Organization (ICAO) and the European Aviation Safety Agency.

It has been argued that the Federal Aviation Regulations (FARs) could be rewritten with UAS in mind, but this option is not straightforward in the foreseeable future due, in part, to concerns raised in the manned aircraft community (The Sunday Telegraph 2007). In fact, the Aircraft Owners and Pilots Association (AOPA) has stated, “Unmanned aircraft and their operation should be certified to the same level of safety as piloted aircraft. Their operation in the National Airspace System should not have a negative impact on general aviation operations and should not require specially designated airspace for their operation” (Aircraft Owners and Pilots Association 2008). Furthermore, the Airline Pilots Association (ALPA) has been watching UAS development very closely and has stated that “The Air Line Pilots Association’s (ALPA’s) position is that no UAS should be allowed unrestricted access to public airspace unless it meets all the high standards currently required for every other airspace user” (Air Line Pilots Association 2011). Thus any discussion of forecasting future FAR changes will not be discussed in this chapter.

How then can UAS comply with existing airspace rules such as 14 CFR Part 91.113 and others, as the regulation specifically states “see” and avoid other aircraft? This is derived from a manned aircraft capability where the pilot has access to the windscreen and can “see” directly outside the aircraft and avoid any potential collisions. The FARs also require an aircraft to have the ability to give “right-of-way” to another aircraft. In fact, ICAO Circular 328 on UAS states,

“A fundamental principle of the rules of the air is that a pilot can see other aircraft and thereby avoid collisions, maintain sufficient distance from other aircraft so as not to create a collision hazard, and follow the right-of-way rules to keep out of the way of other aircraft. Integration of (UAS) may not require a change to the Standards, however, as (UAS) technology advances, alternate means of identifying collision hazards will have to be developed . . .” (ICAO 2011).

Therefore, compliance with current airspace rules point toward a manned aircraft capability. But because UAS do not have a pilot onboard and the technical capability needed to fill the void has yet to be developed, they cannot presently comply with the regulations as written. It is specifically because of this difference that the dialogue took on the title of “*Sense & Avoid*” instead of “*See and Avoid*.” In order to comply, UAS will be required to have the sensing capability to detect traffic using ground-based or aircraft-based sensing equipment, and avoid that traffic using collision avoidance algorithms, in a manner similar to the traffic collision and alerting system (TCAS) (the ICAO term is airborne collision avoidance system or ACAS) used widely in the commercial aviation sector and mandated equipage for air carriers and air taxis that carry in excess of 19 passengers. Lessons learned for addressing research gaps in UAS SAA should be drawn from the development of TCAS/ACAS.

76.3 Lessons Learned from TCAS

TCAS development was a lengthy and expensive process. A recent MITRE review of UAS collision avoidance stated that “By some estimates, TCAS took almost 20 years and expenditures of millions of dollars per year, just for the research and development. This estimate does not include the deployment cost of TCAS” (Lacher et al. 2008). The TCAS software logic for collision avoidance is owned by the federal government and is provided to industry for use in the development of the avionics to be installed onboard aircraft to ensure a specific level of safety and compatibility across platforms. Existing guidance states that any future UAS collision avoidance system be designed so as to integrate with existing collision avoidance systems like TCAS/ACAS. RTCA DO-304 “Guidance Material and Considerations for Unmanned Aircraft Systems” states that the SAA system, “. . . must be interoperable with existing NAS, ATC, and aircraft equipped with existing collision avoidance systems such as ACAS or TCAS” (RTCA DO-304 2007). It is important to point out that TCAS was designed for a human-in-the-loop implementation (with the pilot physically located in the aircraft) and was not designed for an autonomous maneuver. The TCAS system only alarms the pilot to a potential collision and, in some versions, may provide a vertical maneuver recommendation to the pilot to resolve the forecasted collision. TCAS is only compatible with cooperating traffic, in other words, traffic that have an operating transponder onboard. For these and other reasons, direct employment of TCAS technologies on UAS is challenging as explained in ICAO Airborne Collision Avoidance System Manual: “ACAS II was not designed with the intent of being

installed on tactical military (e.g., fighter aircraft) or unmanned aircraft. As such, there are technical and operational issues that must be addressed and resolved prior to installing ACAS II on these types of aircraft” (ICAO 2006). The operational issues uncovered by comparisons to TCAS have helped to inform many of the research gaps for which closure is necessary to achieve UAS SAA.

76.4 Progress in SAA Development for Unmanned Aircraft

The journey of UAS SAA development has been epitomized by starts and stops, lack of funding, and, most importantly, the lack of defined operational/technical requirements. RTCA Special Committee 203 on Unmanned Aircraft Systems (SC-203) and its European counterpart, EUROCAE WG-73, have been working diligently to develop these requirements. However, the complexity of the challenge and the demand for near-term airspace access for UAS have in some cases led developers to pursue solution sets that on the one hand may contribute to problem resolution but on the other may not be “certifiable” by the regulatory authority. To move the process forward, the FAA supplemented the RTCA SC-203 efforts with SAA workshops. These SAA workshops have done important work in decomposing the SAA problem into defined subfunctions. One of the critical outcomes was the definition and functional decomposition of SAA. From the FAA Workshop Final Report (2009):

Definition of Sense and Avoid

Sense and Avoid (SAA) is the capability of a UAS to remain well clear from and avoid collisions with other airborne traffic. SAA provides the intended functions of self-separation and collision avoidance as a means of compliance with the regulatory requirements to “see and avoid” compatible with expected behavior of aircraft operating in the airspace system. As depicted in (Fig. 76.2), a SAA capability performs the following sub-functions:

1. Detect – Determine presence of aircraft or other potential hazards.
2. Track – Estimated position and velocity (state) of a single intruder based on one or more surveillance reports.
3. Evaluate – Assess collision risk based on intruder and UA states.
4. Prioritize – Determine which intruder tracks have met a collision risk threshold.
5. Declare – Decide that action is needed.
6. Determine Action – Decide on what action is required.
7. Command – Communicate determined action
8. Execute – Respond to the commanded action.

While these activities have been ongoing, the OUSD (AT&L) Office of Unmanned Warfare has also been applying and aligning their resources toward technologies that may offer hope in resolving this difficult problem set. Every 2 years since 2003, OUSD (AT&L) has published a roadmap for DoD unmanned systems with an emphasis on gaining airspace access. Additionally, in 2011, OUSD (AT&L) released an updated version of the Unmanned Aircraft System Airspace Integration Plan which states “Without a clear regulatory path to (UAS) integration, DoD intends to leverage new technologies, procedures, and policies to maximize regulatory compliance” (DoD UAS 2011).

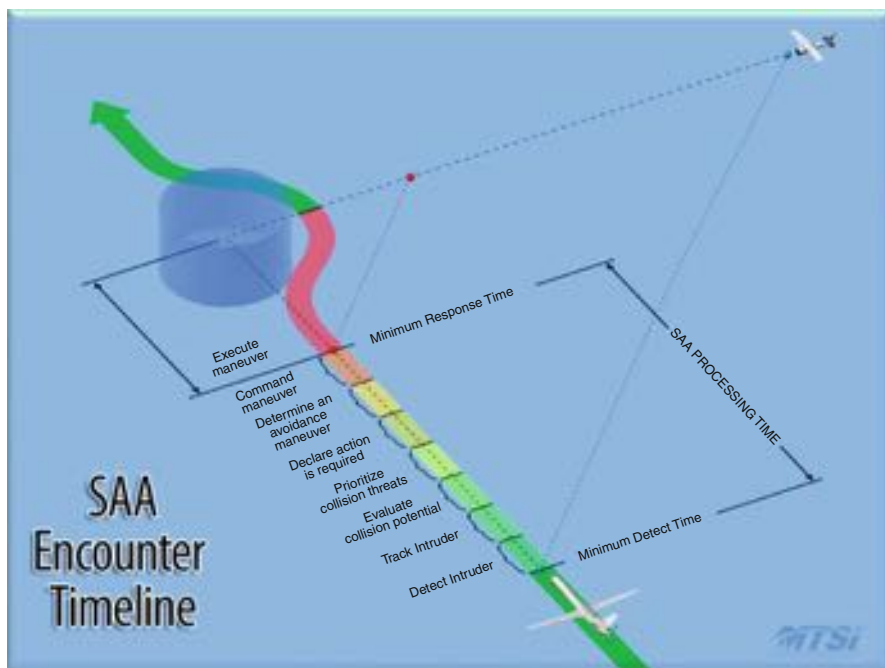


Fig. 76.2 SAA capability subfunctions (From FAA Workshop Final Report (2009))

As part of that leveraging intent, DoD has noted that science and research solutions to SAA are not being fully coordinated. The plan states that “Partnering with internal DoD and external stakeholders will be critical to success” (DoD UAS 2011). Specifically, the plan found that SAA research and development stakeholders and activities are not fully aligned. With shrinking budgets and the lack of progress being made toward tangible solutions OUSD (AT&L), in collaboration with the military services, recognized that a common forum was needed to bring the SAA science and research community together to understand common challenges and the various independent research activities ongoing that might address these open research questions.

76.5 The Sense and Avoid Science and Research Panel (SARP)

In response to this need, the OUSD (AT&L) established the SAA Science and Research Panel (SARP) in February 2011. The SARP consists of a panel of experts in technologies necessary to provide UAS with the ability to sense and avoid other aircraft. It advises the UAS Airspace Integration IPT chaired within OUSD (AT&L) and is currently cochaired by individuals outside of the DoD to provide a more independent perspective of the state of the research and science efforts.

Fig. 76.3 Means that the SARP employs to reach its goals



The panel's primary purpose is to promote partnerships between the DoD and the broader academic and scientific community on UAS NAS Integration science and research initiatives in order to identify open questions and challenges in research and science efforts that must be addressed to provide UAS with an effectual SAA capability. The DoD benefits from these partnerships through a broader range and depth of scientific expertise applied to challenges aligned with DoD priorities. The science and research community benefits from DoD advocacy of potential solutions and through evaluation, test, and implementation of viable approaches. These partnerships are critical to SAA science and research efforts to positively inform and influence OUSD (AT&L) strategic SAA decisions.

The SARP has four primary means at its disposal to promote partnerships and apply expertise to specific SAA challenges and capability gaps, shown in Fig. 76.3. The first of these tools are the panel meetings themselves. The panel consists of experts from DoD, government research facilities, federally funded research and development centers, academia, and systems engineering and technical assistance (SETA) contractors who are conducting SAA science and research. The panel meets three times per year to identify unanswered research questions and technical challenges (or "gaps") necessary to solve and implement SAA and determine how these gaps relate to OUSD (AT&L) SAA priorities. In order to classify the research gaps, the SARP uses a capability-based taxonomy developed from the eight core SAA functions displayed in Fig. 76.2. This taxonomy, originated in support of a OUSD (AT&L) SAA blueprint, describes ten core SAA capabilities – directly related to the Fig. 76.2 SAA functions – and seven crosscutting SAA capabilities which support several SAA functions, shown in Fig. 76.4.

The ten SAA core capabilities are defined in Table 76.1. An example of a research gap related to a core capability is "How does the collision avoidance algorithm address uncertainty with respect to noncooperative traffic and maneuvering targets?" This research gap is mapped to the SAA core capability of threat assessment. The seven SAA crosscutting capabilities are defined in Table 76.2. A research gap identified by the SARP that relates to a SAA crosscutting capability is "What are the advantages and disadvantages of informative, directive, and/or suggestive display

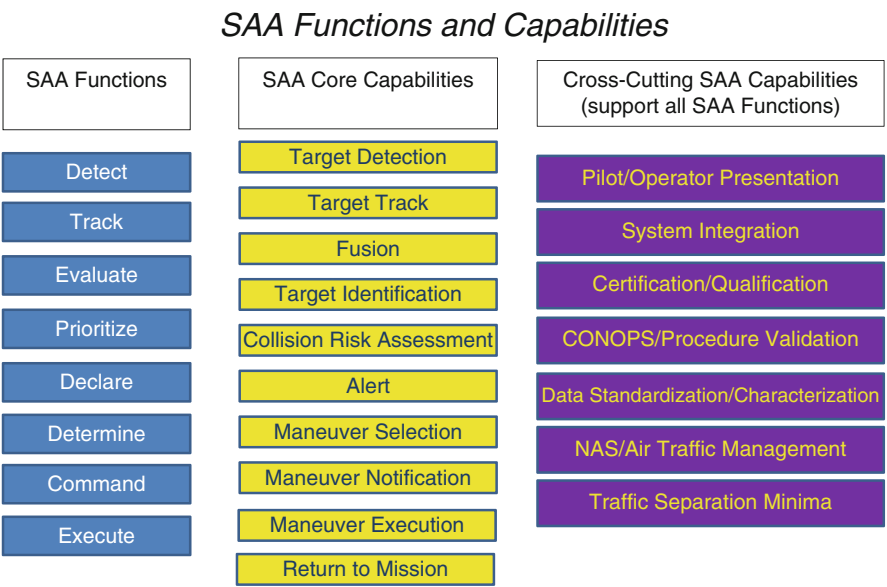


Fig. 76.4 SAA capabilities used to classify SAA research gaps

Table 76.1 Definitions of SAA core capabilities

| SAA capability | Definition |
|-----------------------|---|
| Target detection | Ability to determine presence and location of any airborne object that presents a collision threat to the UA covered by this SAA system |
| Target track | Ability to refine a detected airborne object’s position (2-D or 3-D) and velocity based on one or more target detection(s) from a single sensor |
| Fusion | Ability to refine a detected airborne object’s position (2-D or 3-D) and velocity based on one or more target detection(s) or track(s) from multiple sensors |
| Object identification | Ability to discriminate type and characteristics of a detected airborne object |
| Threat assessment | Ability to determine risk of collision based on relative position (2-D or 3-D) and velocity of the detected airborne object and the UA covered by this SAA system |
| Alert | Ability to communicate that a risk threshold has been met |
| Maneuver selection | Ability to determine the appropriate action needed to resolve identified collision risk(s) to the appropriate separation threshold |
| Maneuver notification | Ability to communicate the selected maneuver in order to execute within the timing window |
| Maneuver execution | Ability to execute the selected maneuver within the timing window |
| Return to mission | Ability of UAS to return to intended or amended course after an avoidance or separation maneuver is complete |

Table 76.2 Definitions of SAA crosscutting capabilities

| SAA capability | Definition |
|---|--|
| Pilot/operator presentation | Ability to provide the pilot/operator with the required information needed to perform/monitor the core SAA capabilities |
| System integration | Ability to integrate the core capabilities into existing and planned UAS |
| Certification/qualification | Ability to show through process and evidence that the SAA system meets an acceptable safety level |
| Concept of operations (CONOPS)/procedure validation | Ability to conform to approved CONOPS and operational procedures |
| Data standardization/characterization | Ability to enable collaboration and meaningful comparisons through databases, data standards, and data integrity |
| NAS/air traffic management | Ability to assess the impact on the NAS that UAS system performance and the performance of supporting systems, such as SAA, may have on air traffic management (ATM) |
| Traffic separation minima | Ability to determine the separation standards and the environmental or operational conditions which inform SAA design |

types for SAA and what lessons have been learned from previous applications?” This gap is mapped against the “pilot/operator presentation” as a crosscutting SAA capability.

To date, the SARP has identified 54 research gaps with at least one gap belonging to each of the 17 SAA capabilities. Figure 76.5 shows the number of gaps identified in each SAA capability. In addition to identifying research gaps, the SARP also receives presentations on ongoing SAA research activities and attempts to use its partnerships to identify ongoing science and research activities that may be ongoing that can close these gaps. These activities may include those conducted under DoD sponsorship or those being conducted outside DoD sponsorship by other government research laboratories (e.g., NASA), academic institutions, or internal corporate research initiatives.

76.6 Workshops

A second way in which the SARP accomplishes its goals is through chartering workshops on specialized topics. Workshop participants are invited from government, academia, and industry based on specific skills and expertise in an area that the workshop is investigating. Thus far the SARP has held four workshops on the following topics:

- SAA certification
- Use of bird RADARs for GBSAA Target Classification
- RADAR cross-section guidelines for SAA
- Guidelines for quantifying the effects of qualitative contributions to safety

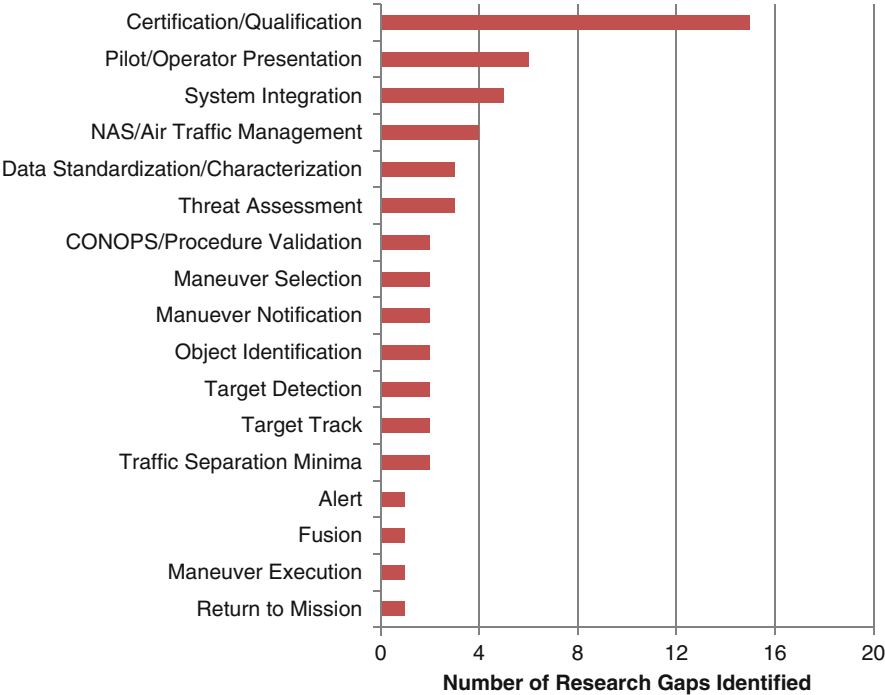


Fig. 76.5 Number of research gaps identified in each SAA capability

Per Fig. 76.5, the highest number of research gaps belongs to SAA certification and qualification. The SAA certification workshop was held with the objective of determining the research and science efforts needed to answer key questions related to public UAS SAA certification. The workshop participants examined 12 research gaps identified by the SARP specific to SAA certification/qualification as shown in Table 76.3. For each of these areas, workshop participants identified ongoing research activities that may assist in answering these questions and provided a recommended first step.

The second workshop chartered by the SARP was to investigate how bird RADARs could be used to address some of the gaps in ground-based SAA target classification. The workshop found that the inability to differentiate between birds and other clutter in the radar data is a serious challenge for SAA systems and that while bird radars are cost-effective, they seldom have the reliability needed for safety-critical systems due to their update rates, airspace volume coverage, and frequencies. However, bird RADARs have cost advantages and some have “stop and stare” capabilities. If birds could be reliably identified by SAA systems, algorithms could be developed to use this information in a fusion tracker to provide this information to the UAS crew and ATC to improve airspace safety and lower the false alarm rate for SAA systems.

Table 76.3 Research gaps examined at SAA certification workshop

| Gap area | Description |
|---------------------------------|--|
| Airspace characterization | How should airspace characterization be included as part of SAA certification? |
| RADAR data | What types and amounts of RADAR data must be collected to assure the certifying authority that all noncooperative traffic is sensed? |
| Safety case | What safety strategy and artifacts must be included in a SAA safety case? |
| Human factors | What human factor considerations must be applied to the ground control element for certification? |
| Commercial off-the-shelf (COTS) | How can COTS equipment be used in a SAA system-of-systems that minimizes recertification and redesign? |
| Architectures | How should SAA CONOPS and concepts of employment be matched to an optimal (e.g., cost, safety) SAA system architectures? |
| Modeling and simulation (M&S) | How should the M&S tools used in the SAA certification process be evaluated and validated as accurate and representative of the SAA system? |
| Airworthiness | How do the service 3-tier airworthiness certification levels input into a target level of safety calculation for the SAA system? |
| Segregated certification | What are the criteria to determine which parts of the overall SAA system can be certified independently (e.g., ground segment, RADAR, sensors) in a systems of systems approach? |
| Multi-UAS integration | What are the certification challenges associated with multiple UAS operating in close proximity to each other (e.g., determining how to change SAA thresholds for formation flight)? |
| FAR compliance | What challenges exist for all aircraft (manned and unmanned) in complying with the FARs in their present form (e.g., right-of-way rules in 14 CFR 91.113)? |
| Software assurance | How do civil software assurance considerations in RTCA DO-178B (1992) software assurance apply to public UAS airspace integration? |

The third workshop was chartered by the SARP to determine guidelines for RADAR cross section (RCS) as part of a RADAR-based SAA solution. Participants from government, industry and academia reached a general consensus at the workshop on the most stressing targets for RADAR-based SAA systems. Additionally, participants identified RCS data needs, shared RCS data, identified areas of potential cooperation, and identified useful parameters to consider in multidimensional design trade space for RADARs used as part of a SAA system.

The fourth workshop chartered by the SARP had the objective of understanding how the qualitative safety requirement “extremely improbable” could be interpreted and measured using quantitative means (e.g., a target level of safety approach). Interim guidance provided by the FAA states, “Applicants proposing ‘see and avoid’ strategies in lieu of visual observers, need to support proposed mitigations with system safety studies which indicate the operations can be conducted safely.

Acceptable system safety studies must include a hazard analysis, risk assessment, and other appropriate documentation that support an ‘extremely improbable’ determination” (FAA 08-01 2008). In exploring this topic, the workshop participants reviewed different quantitative safety levels used in the past in civil and military aviation systems to justify an “extremely improbable” or equivalent determination and investigated means of including qualitative contributions to safety (e.g., software assurance, human factor considerations) in a quantitative safety target. The outcome of the workshop was threefold:

- A relationship between the “extremely improbable,” MIL-STD-882D (2000) safety methodology and the target level of safety methodology was proposed.
- A method to identify contribution of qualitative hazards/mitigations to a quantitative level of safety was proposed.
- A list of disciplines that should undergo qualitative hazard analysis or comparable (instead of direct quantification) was established.

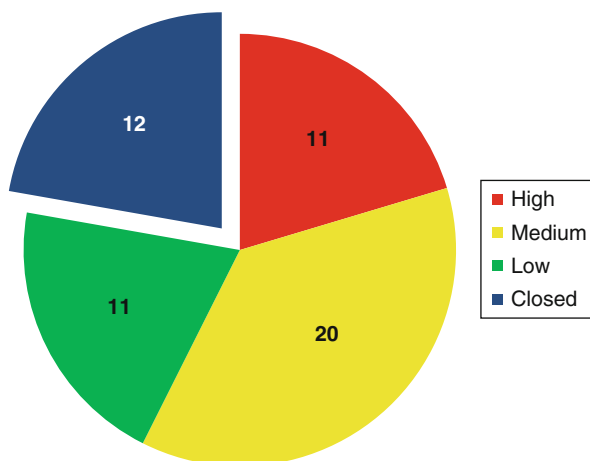
76.7 Requests for Information

Another way in which the SARP accomplishes its objectives is through formal requests for information. Requests for information (RFI) appear on the Federal Business Opportunities website and solicits feedback in the form of white papers from interested parties. In August 2011, the OUSD (AT&L) submitted a SARP RFI (OUSD ATL) that listed open research questions in SAA displays and algorithms so that the SARP can obtain information on ongoing activities and obtain a sense of the state of the art in these areas. SAA algorithms and the display of SAA information are two areas that the SARP has identified as critical to the development of SAA systems. The specific questions in the RFI were developed by the SARP based on identified research gaps related to SAA algorithms and displays. The SARP received responses and feedback from over 30 parties in response to various aspects of the RFI.

76.8 Open Meetings

The fourth tool that the SARP uses to meet its goals is to sponsor open meetings. Open meetings are designed to be open to participation for interested parties beyond those that participate in the panel meetings and/or workshops. A 2-day open meeting was held in October 2011 as a forum to discuss selected white papers received in response to the RFI. Seventeen different parties from industry, government, and academia presented 20 separate presentations on ways to address research gaps related to SAA displays and algorithms. From the open meeting, the SARP was able to better understand the state of the art in these critical areas and also understand key areas where technical and/or policy guidance is needed in the community. The SARP consolidated these lessons learned from the open meeting and provided recommendations back to OUSD (AT&L).

Fig. 76.6 Number of research gaps by priority (including *closed* gaps)



76.9 Closing the Gaps

Through the four methods in Fig. 76.3, the SARP has brought to bear through partnerships the community's expertise and knowledge of ongoing research activities. As a result, over 20 % of identified research gaps are regarded as closed. As the SARP continues to address the open research gaps, a high-medium-low categorization scheme is employed to prioritize the areas where the SARP will focus its efforts and also advise OUSD (AT&L) as to the relative importance of the research gaps. Figure 76.6 shows the number of research gaps in each priority category, in addition to the number that have been closed.

76.10 Conclusions

The SARP exists to bring the science and research community together to address open research questions in SAA. The SARP has four primary means of accomplishing its objectives – panel meetings, workshops, requests for information, and open meetings. The SARP has identified 54 open research questions and has adopted a capabilities-based taxonomy to classify these research gaps using ten SAA core capabilities and seven crosscutting capabilities. Over 20 % of these research gaps have been closed as a result of SARP activities. As progress in SAA moves forward, the SARP will continue to identify and bring focus to research areas critical to the success of SAA. By promoting alignment and partnerships in the science and research community, the SARP strives to advance timely solutions to key SAA challenges for the DoD and broader UAS community.

Acknowledgments The authors are grateful to Mr. Dallas Brooks and Ms. Viva Austin for their leadership and vision in establishing the SARP. The authors recognize the dedication and technical contributions of the OUSD(AT&L) Airspace Integration IPT, the Systems Integration sub-IPT, Mr. Chris Jella, and the SARP in developing the SAA Blueprint from which the SAA Capability Definitions were derived.

References

- Aircraft Owners and Pilots Association, Regulatory brief: unmanned aircraft systems (2008), <http://www.aopa.org/whatsnew/regulatory/unmanned.html>
- Air Line Pilots Association White Paper on Unmanned Aircraft Systems, Unmanned aircraft systems challenges for safely operating in the national airspace system (2011), <http://www.alpa.org/portals/alpa/pressroom/inthecockpit/UASWhitePaper.pdf>
- DoD UAS, Airspace integration plan v. 2.0.29 (2011)
- FAA Workshop Final Report, Sense and avoid (SAA) for unmanned aircraft systems (UAS) (2009)
- FAA 08-01, FAA interim operational approval guidance for unmanned aircraft system operations in the U.S. national airspace system (2008)
- J. Gertler, U.S. unmanned aerial systems. Congressional research service report for congress R42136, 2012, p. 6
- ICAO, *Airborne Collision Avoidance System (ACAS) Manual*. Doc 9863, Paragraph 1.5.3, 1st edn. (ICAO, Montréal/Manchester, 2006)
- ICAO, *Unmanned Aircraft Systems*. ICAO Circular 328 (ICAO, Montréal, 2011)
- A. Lacher, D. Maroney, A. Zeitlin, Unmanned aircraft collision avoidance – technology assessment and evaluation methods (2008), http://www.mitre.org/work/tech_papers/tech_papers_08/07-0095/07-0095.pdf
- S. Magnuson, Safety concerns still blocking unmanned aerial vehicles from national airspace. National Defense Magazine (2012)
- MIL-STD-882D, *Department of Defense Standard Practice: System Safety* (Department of defense, Washington, DC, 2000)
- Solicitation Number OUSD(ATL)STS-RFI, Sense and avoid (SAA) science and research panel. Special notice (2011)
- RTCA DO-178B, *Software Considerations in Airborne Systems and Equipment Certification* (RTCA, Washington, DC, 1992)
- RTCA DO-304, *Guidance Material and Considerations for Unmanned Aircraft Systems* (RTCA, Washington, DC, 2007)
- The Congress of the United States Congressional Budget Office, *Policy Options for Unmanned Aircraft Systems* (Congress of the U.S., Congressional Budget Office, Washington, DC, 2011), p. vii
- The Sunday Telegraph, Pilots fear crash danger as drones prepare to take off, 2007, p. 12
- United States Code of Federal Regulations, Title 14, Part 91
- United States Department of Defense, Unmanned systems integrated roadmap FY 2011–2036

Automatic Traffic Alert and Collision Avoidance System (TCAS) Onboard UAS

77

Jörg Meyer, Matthias Göttken, Christoph Vernaleken, and
Simon Schärer

Contents

| | | |
|------|--|------|
| 77.1 | Introduction | 1858 |
| 77.2 | Objective and Status of the Project | 1859 |
| 77.3 | Definition of the Cooperative Collision Avoidance System | 1859 |
| 77.4 | Human Machine Interface | 1861 |
| 77.5 | Validation and Verification | 1863 |
| 77.6 | Flight Test | 1865 |
| 77.7 | Consequences of TCAS Onboard UAS | 1868 |
| 77.8 | Conclusions and Way Ahead | 1870 |
| | References | 1871 |

Abstract

In July 2010, Cassidian installs a cooperative collision avoidance system based on Traffic Alert and Collision Avoidance System (TCAS) onboard the Barracuda Unmanned Aerial System (UAS) demonstrator.

The “Sense&Avoid” capability is a key challenge, even when only considering operation in air space classes A–C. For that reason, the demonstration is intended as a risk mitigation measure to support future UAS development efforts.

The demonstration steps involve a laboratory test with real equipment in a closed-loop demonstration and the automatic execution of a collision avoidance manoeuvre in flight. The in-flight demonstration takes place in segregated air space involving the Barracuda demonstrator platform and a real intruder.

The section outlines the setup of the cooperative collision avoidance system demonstrator, including the interface with the flight control system of

J. Meyer (✉) • M. Göttken • C. Vernaleken • S. Schärer
Cassidian, Manching, Germany
e-mail: Joerg.Meyer@cassidian.com; matthias.goettken@cassidian.com;
Christoph.Vernaleken@Cassidian.com; simon.schaerer@cassidian.com

the Barracuda and the control authority granted to the cooperative collision avoidance system demonstrator.

It summarizes the Human-Machine-Interface (HMI) layout and situation awareness provided to the UAS operator and illustrates how the potential loss of data link connection is covered by the system design.

The paper details the development process applied, including requirements definition, validation, and verification steps as well as hardware-in-the-loop demonstration in support of the in-flight demonstration.

It illustrates the demonstration scenarios utilized in the flight demonstration and the results achieved.

Finally, the paper draws conclusions regarding the potential of such cooperative collision avoidance systems based on TCAS, the operational implications and the certification aspects to be solved.

77.1 Introduction

Sense&Avoid is considered as THE enabling capability required to allow routine operation of UAS in non-segregated airspace.

Several studies (Final Air4All Report for ‘UAV insertion into General Air Traffic GAT Follow On Contract’ 2009; WASLA/HALE Phase III 2008 among others) have shown that the Sense&Avoid capability is both technically challenging and lacking in mature system concepts which are able to comply with strict safety requirements.

The UAS Roadmap Study (Final Air4All Report for ‘UAV insertion into General Air Traffic (GAT) Follow On Contract’ 2009), performed by the Air4All consortium, analysed the status of the Sense&Avoid capability and concluded on a stepwise integration of UAS into the airspace. This stepwise integration is based on a preliminary step to allow operation in airspace classes A–C only, whereas a second step would cover operation in all airspace classes.

The preliminary step excludes separation and collision avoidance against non-transponding intruders, which is considered technologically challenging. No separation capability is foreseen, as separation (IFR vs. IFR and IFR vs. VFR (IFR: Instrument Flight Rules; VFR: Visual Flight Rules)) in airspace classes A–C is performed by Air Traffic Control (ATC). The measure of excluding those capabilities which are technically challenging comes at the cost of operational limitations to airspace classes A–C, with special procedures being required to transit in and out of this operational environment.

The proposed second step would extend the technical capabilities of the Sense&Avoid system to also cover separation and collision avoidance against non-transponding objects. This would allow the reduction of the operational limitations required for the preliminary step and further increase the operational benefits possible by UAS operation.

The benefit of this stepwise approach is that technically challenging capabilities are excluded for the time being, thereby allowing routine operation of UAS in non-segregated airspace in the short term. That way, more time is gained to enable further development of the technology needed to bridge the gap to the second step of the Sense&Avoid capability.

A Sense&Avoid capability covering the proposed preliminary step has to consider the operational environment in the airspace classes A–C and be compatible with mandatory equipment (Mode-S transponder and TCAS (MTOW (MTOW: Maximum Take-Off Weight) above 5.7 to or more than 19 passengers)).

77.2 Objective and Status of the Project

In July 2010, a Cassidian internal research and technology project was set up to demonstrate a technical solution enabling operation of the UAS in airspace classes A–C according to the stepped approach defined by Air4All (Final Air4All Report for ‘UAV insertion into General Air Traffic (GAT) Follow On Contract’ 2009).

With separation being the responsibility of ATC and non-transponding aircraft not being allowed to fly into those airspace classes, only a collision avoidance capability is required. Such a cooperative collision avoidance capability has to comply with the available operational framework and the equipment fit of the intruders surrounding the UAS. A TCAS II based cooperative collision avoidance system, as used in manned aviation for decades, is considered a suitable technical solution to this problem.

Consequently, Cassidian analysed a potential solution for a cooperative collision avoidance system based on TCAS II to enable routine operation of the UAS in airspace classes A–C with special procedures required to transit to and from these airspace classes.

The demonstration of this cooperative collision avoidance capability involved several steps:

- Validation and verification using Hardware-In-The-Loop simulation with closed-loop simulation involving a virtual intruder equipped with an ATC transponder
- Several flight tests being performed onboard Cassidian Barracuda UAS demonstrator with a real intruder ([Demonstration video](#)) (Fig. 77.1).

77.3 Definition of the Cooperative Collision Avoidance System

Airbus has developed and received certification for a system consisting of a common TCAS installation and a dedicated autopilot mode for the A380, allowing automated execution of TCAS Resolution advisories by the A380 AutoFlight System. The system certified by Airbus implements, among other things, a so called “opt out”

Fig. 77.1 Barracuda UAS demonstrator



logic, which automatically initiates the execution of the TCAS Resolution Advisory (RA) but gives the A380 pilots the ability to abort the automatic execution and to command a different avoidance action using the standard means they have to control the aircraft.

The cooperative collision avoidance system developed for the Barracuda UAS demonstrator further develops this concept by extending the control chain by the datalink connection.

The system has been set up to have full authority over the execution of the TCAS II defined avoidance manoeuvres, independent of data link availability. In case of data link availability, the UAS operator has the ability to override the automatic execution and to command a self defined avoidance action based on semi-automatic modes provided by the Barracuda FCS.

The automatic execution of the TCAS defined avoidance manoeuvre was performed immediately, in contrast to operation onboard a manned aircraft, where the TCAS RA needs to be indicated to and understood by the pilot, thereby causing a delay in execution before an avoidance action is initiated.

The cooperative collision avoidance system consists of the following elements:

- Commercial-Off-The-Shelf (COTS) Mode-S transponder including omnidirectional antennae.
- COTS TCAS II system including a directional antenna mounted on top the UAS demonstrator and an additional omnidirectional antenna.
- Collision avoidance software implemented as part of the Barracuda Flight Control System (FCS), involving evaluation of the TCAS II generated resolution advisories, system moding as well as commanding the avoidance manoeuvre to a dedicated autopilot mode responsible for execution of the TCAS RAs.
- Human-Machine-Interface to provide situation awareness and an interface for control commands defined by the UAS operator. HMI information was transmitted in real-time using the Barracuda datalink.

The system components are illustrated in the following picture (Fig. 77.2)

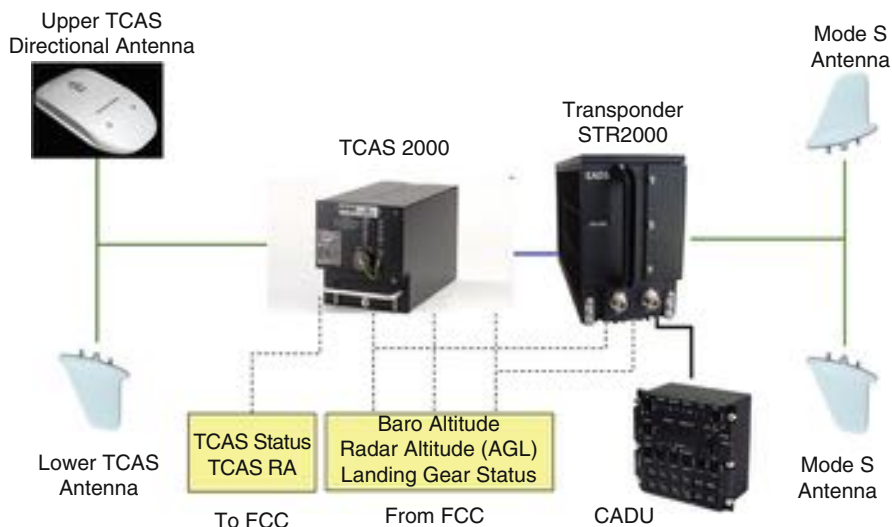


Fig. 77.2 System architecture of cooperative Sense&Avoid system

77.4 Human Machine Interface

The intention of the TCAS traffic display is to aid pilots in the visual acquisition of traffic, including the differentiation of intruder threat levels, to provide situational awareness and to ensure confidence in the presented resolution advisories. TCAS standardization documents (Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II [2008](#)) contain detailed guidelines on the traffic display, including a standardized symbology set for the depiction of other traffic.

Since all traffic tracked by TCAS is ranked by the collision avoidance logic, intruders are sent to the display in a prioritized order, thus ensuring that those intruders most relevant to collision avoidance are displayed.

Traffic is displayed in relation to a symbol reflecting ownship position, on a display featuring range rings. The representation of other traffic is typically limited to a vertical volume of $\pm 2,700$ ft, and the symbology used depends on the alert state and the proximity of the corresponding traffic. The picture below illustrates the traffic display used for the cooperative collision avoidance system implemented onboard the Barracuda UAS demonstrator (Fig. [77.3](#)).

As is evident from the figure, the traffic symbology used is identical to that used on manned aircraft. Accordingly, traffic symbols are supplemented by a two-digit altitude data tag in the same color as the intruder symbol, indicating the altitude of the intruder aircraft relative to ownship in hundreds of feet. For traffic above the own aircraft, the tag is placed above the traffic symbol and preceded



Fig. 77.3 TCAS presentation on map display

by a “+,” whereas other aircraft below ownship feature the tag below the symbol with a “-” as prefix. Whenever an intruder is climbing or descending with a vertical rate in excess of 500 ft/min, an arrow pointing upward or downward, respectively, is presented immediately adjacent to the intruder symbol. In case bearing information is unavailable for traffic generating either a TA or RA alert, alpha-numeric information is presented on the traffic display for up to two intruders, usually centered below the ownship symbol. Information is presented in the format threat level – range (in NM) – altitude (in 100 ft, consistent with absolute/relative altitude setting) and, if applicable, an intruder vertical speed arrow. As an example, “TA 5.2 -06↑” represents an intruder causing a TA at 5.2 nm with a relative altitude of -600 ft, which is climbing with 500 ft/min or more.

Since all decisions pertaining to the flight path of the aircraft are taken using the Command & Control (C&C) HMI, there was a clear need to have an indication of the TCAS alert condition on this format. Therefore, both intruder information and the standard TCAS callouts were replicated on the C&C HMI as shown in the following picture.

For TCAS callouts, the background of the text message corresponds to the respective alert level, i.e., amber for a TCAS Traffic Advisory (TA) and red for an RA. In the latter case, the standard callout was supplemented by an alphanumeric indication of the commanded vertical speed, if applicable. For the “CLEAR OF CONFLICT” announcing the end of the traffic conflict, the background was green to announce the reversion to Normal Operations (Fig. 77.4).

Operator feedback concerning the implementation of the TCAS HMI was requested from all Barracuda operators. The operators involved stated that the



Fig. 77.4 TA/RA/“Clear of conflict” indication (extract from Command and Control HMI)

overall impression was excellent, and that the display provided them with a level of situational awareness with respect to the surrounding traffic that was otherwise unachievable. In general, the implementation of the TCAS traffic display was according to their expectations, and consistent with the solutions which the operators know from manned aircraft. The presentation of traffic information in relation to a moving map of the surrounding terrain, a feature still uncommon on most manned civil aircraft, was particularly appreciated in this context. Likewise, the textual presentation of standard TCAS callouts, a novel feature compared to current TCAS implementations, was considered as a useful feature.

77.5 Validation and Verification

The system under test was developed in compliance with a development process implementing the well-known V-model. This process is illustrated in the following picture and involves requirement definition and breakdown from system top level to equipment level. In order to ensure requirements correctness and complete-

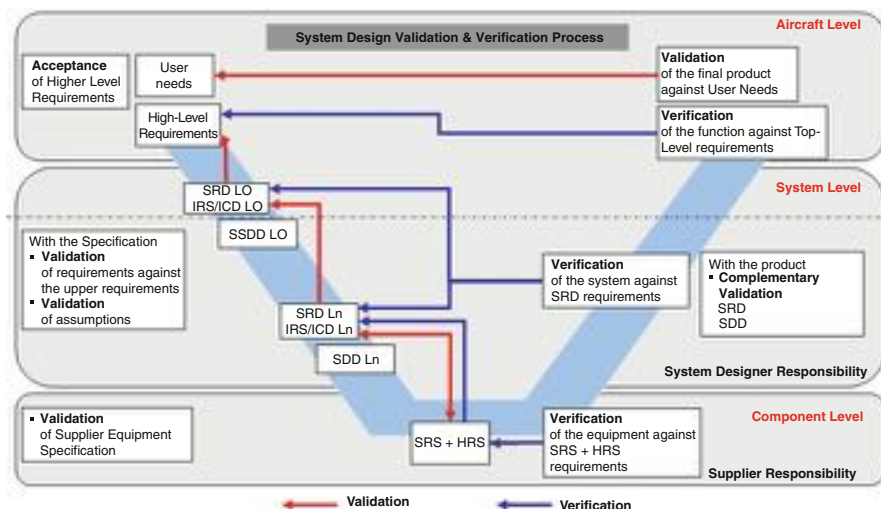


Fig. 77.5 Validation and verification process

ness, a validation matrix was set up to formalize the system validation activities (Fig. 77.5).

Requirement engineering and tracing was done using a dedicated requirements management tool, supporting validation and verification activities in the project.

A formal verification of the system was performed to ensure that the system design correctly implemented the requirements. Verification was performed using different means of verification, involving Hardware-In-The-Loop simulations including real hardware.

This Hardware-In-The-Loop Simulation (HILS) setup consisted of the system under test embedded in a closed loop simulation involving the Flight Control System (FCS) of the Barracuda UAS demonstrator, communication system involving LOS datalink simulation and Ground Control Station (GCS) for UAS operator situation awareness and monitoring system operation. This setup had been chosen to ensure end-to-end coverage for the test performed.

The system was stimulated by a dedicated TCAS test system capable of generating Mode-S equipped virtual intruders flying on a collision course to the Barracuda. This setup allowed the definition of all the relevant parameters of the collision scenario and the performance of those tests in a safe, controllable and repeatable manner.

The HILS setup is illustrated in the following picture (Fig. 77.6):

Verification of the system on closed-loop HILS was performed to ensure correct operation of the system and is considered an essential step before performing real flight tests.

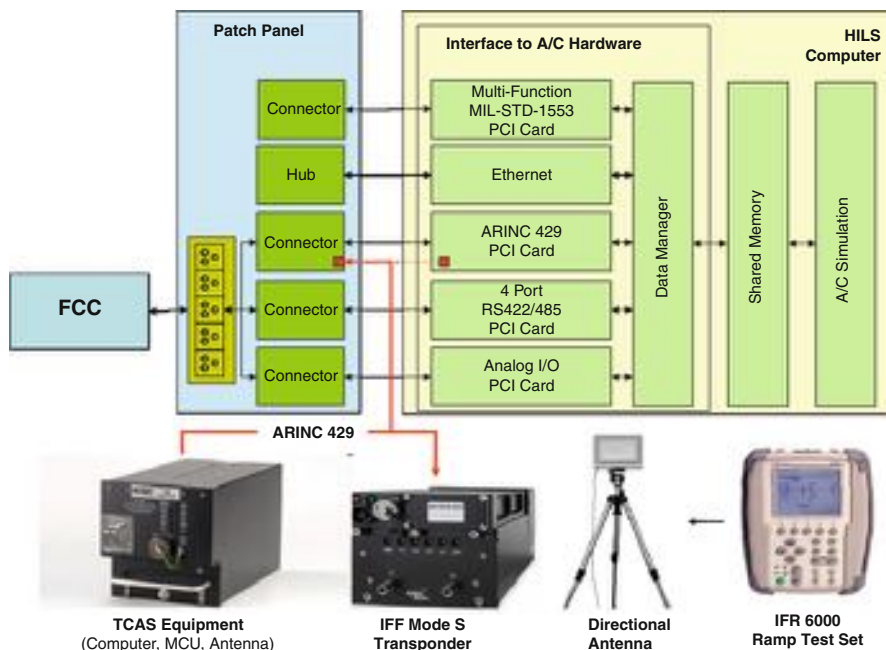


Fig. 77.6 Hardware-in-the-loop setup

The verification activity consisted of a great number of different collision scenarios with different approach angles, closing speeds and vertical separation.

77.6 Flight Test

Flight demonstration of the cooperative collision avoidance system was performed involving the Barracuda and a real intruder. The flight test demonstration took place at Goose Bay airport, Labrador, Canada in July and August 2010.

Collision scenarios were performed according to a dedicated flight test plan and involved those collision scenarios which had previously been verified using closed loop HILS.

The intruder used during flight tests was a King Air 100. It was equipped with a Mode-S transponder. However, a TCAS capability was not installed, but this did not impose any constraint on the cooperative collision avoidance system developed.

The test points were performed by visually flying the intruder in close proximity to the UAS demonstrator, which strictly followed the pre-programmed flight plan until TCAS commanded otherwise. For safety reasons, the test points required good visibility conditions.

Collision avoidance test points involved a stepwise increase in collision scenario criticality. Initial test points involved a verification of the TCAS surveillance function, correct behavior of the TCAS collision avoidance function and the UAS operator situation awareness. For these tests, the automatic execution of the TCAS RA onboard Barracuda UAS was disabled. In the related test points, the Intruder A/C approached the Barracuda UAS from behind at intruder bearing angles between approximately $\pm 135^\circ$ and relative altitudes of ± 200 ft. In all test points, detection of the intruder A/C, generation of the expected TCAS TA and RAs and indication to the UAS operator were successful.

After these test points had been performed, the automatic execution of the TCAS RAs was enabled. The test points involved a collision threat imposed by the intruder aircraft approaching the Barracuda UAS demonstrator from behind until the TCAS alert thresholds were violated, triggering the expected TCAS reactions. This time, under supervision of the UAS operator in the Ground Control Station (GCS) and the intruder pilot, the Barracuda Flight Control System executed the TCAS RA with the required climb/descend rate until the TCAS issued the "Clear of Conflict" signal. The flight test scenarios involved the intruding A/C approaching the Barracuda UAS from behind starting at a distance of approximately 900 m and at relative altitudes of +200 and -200 ft. It should be emphasized that these test scenarios did not yet involve crossing trajectories.

Detection of the intruding A/C, generation of TA and RA by the Barracuda TCAS and execution of the TCAS RA with the defined climb/descend rate were performed as expected. After the TCAS issued the "Clear of Conflict" signal, Barracuda returned to its pre-programmed flight trajectory.

Ultimately, the test points also involved two more challenging collision scenarios and higher approach speeds. In these test scenarios, the intruder A/C continued on its trajectory, crossing underneath the Barracuda UAS, thereby causing the generation of TCAS TA and climb RA, which was promptly executed by Barracuda. The following picture illustrates one of the tested collision scenarios. The intruder A/C is approaching Barracuda from the rear side (collision angle approx. 135°) at a relative altitude of -200 ft (Fig. 77.7).

The other test scenario was performed with a collision angle of approx. 90° and at a relative altitude of -200 ft. The following picture illustrates this test scenario (Fig. 77.8):

Vertical separation *before* TCAS activation for all test points was around 200 ft, with horizontal separation at point of collision being almost zero. By strict execution of the TCAS defined resolution advisory by the FCS of the Barracuda, the vertical separation was extended to approx. 350 ft in line with expectations for operation of TCAS.

When the intruder A/C was in position the performance of a scenario took about 1.5 min (until Barracuda returned to the pre-programmed flight plan after completion of the avoid manoeuvre).

Fig. 77.7 Intruder side approach scenario (collision angle 135°)

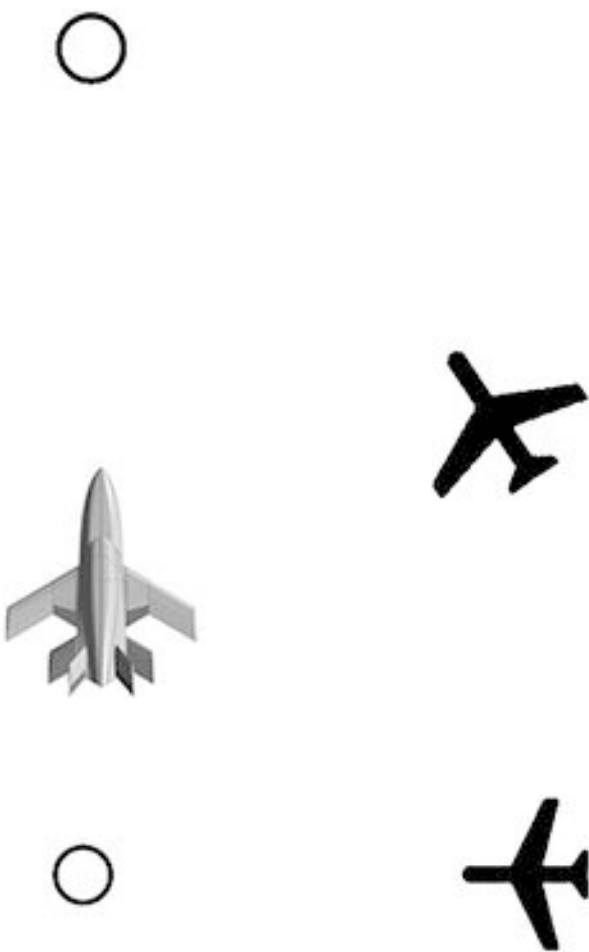


Fig. 77.8 Intruder side approach scenario (90° collision angle)



77.7 Consequences of TCAS Onboard UAS

The proposed solution of automatic execution of the TCAS generated avoidance manoeuvres provides benefits when compared to operation in manned aviation.

In the proposed solution, the automatic execution of the avoidance manoeuvre is performed by a dedicated, specialized autopilot mode. This autopilot mode is far more precise in following the TCAS defined resolution advisory than a human pilot could ever be.

Due to this fact, a more precise execution of the TCAS defined avoidance manoeuvre is possible, thereby reducing vertical and/or lateral deviation from the expected trajectory.

The achievable TCAS risk ratio is one factor (Other important factors include typical “1309-type” safety requirements applicable to the whole UAS and its operation. The relevance of the “logic risk ratio” figures can be understood as the reliability that can be attached to RAs, they are not supposed to be treated as the sole means to express safety of the system. Reference ICAO Annex 10 (2007) states: “[The ACAS risk ratio] express the effect that following an RA will have on the immediate risk of collision when, at the time it is issued, the pilot has no information other than the RA on which to base a decision whether to follow the RA or ignore it. As a rough guide, the collision risk created by ACAS arises from following the RA so the logic risk ratio overstates this “induced risk ratio”; on the other hand, it also overstates the capability of ACAS to prevent collisions because of the many other failure modes in the total system.”) to value the impact on safety achieved by such a system (ICAO Annex 10 2007):

$$\text{ACAS Risk Ratio} = \frac{\sum_{\text{all encounters}} \text{probability of a collision with ACAS installed}}{\sum_{\text{all encounters}} \text{probability of a collision without ACAS installed}}$$

Reference ICAO Annex 10 (2007) quantifies target levels of Airborne Collision Avoidance System (ACAS) risk ratio applicable to manned aviation:

[T]he collision avoidance logic shall be such that the expected number of collisions is reduced to the following proportions of the number expected in the absence of ACAS:

- a) when the intruder is not ACAS equipped: 0.18
- b) when the intruder is equipped but does not respond: 0.32
- c) when the intruder is equipped and responds: 0.04

Especially in crowded airspaces, it had been shown that the precise execution of the TCAS RA directly correlates with the risk of inducing new collision threats – the more accurate a TCAS RA is followed, the less induced new collision threats, thereby improving the TCAS risk ratio (Final Air4All Report for ‘UAV insertion into General Air Traffic GAT Follow On Contract’ 2009).

Furthermore, the TCAS risk ratio also depends on the time duration between issue of a Resolution Advisory by the TCAS system and initiation of the execution of the TCAS RA. ICAO Annex 10 (ICAO Annex 10 2007) specifies a standard pilot model, in which a typical reaction time of a human pilot is defined. According to

this standard pilot model, a typical response time between issue of TCAS RA and initiation of execution is 5 s.

It has been shown in (Final Air4All Report for ‘UAV insertion into General Air Traffic GAT Follow On Contract’ 2009), that this time duration directly impacts the TCAS risk ratio. With the proposed solution involving automatic execution of a TCAS RA, this time duration can be minimized. A direct impact on the TCAS risk ratio is illustrated in the following picture (Fig. 77.9).

The benefits possible by implementing an automatic execution of a TCAS defined avoidance manoeuvre have also been considered for manned aviation. A similar, but more complex solution has been proposed for the Airbus A380 and received certification from EASA in 2009 (EASA 2009).

The EUROCONTROL CAUSE study (Unmanned Aircraft Systems 2010) did assess the suitability of TCAS for UAS as well. Reference Unmanned Aircraft Systems (2010) puts emphasize on the UAS requirements necessary in order not to degrade TCAS operation:

- Limitations in ACAS hardware (particularly antennae and their siting) may become apparent when ACAS is deployed on a UAS airframe, and limitations in the ACAS software (particularly tracking algorithms) may become apparent if the UAS aerodynamic performance exceeds that expected from a commercial civil fixed-wing aircraft. These limitations will manifest themselves as degraded surveillance performance and lower reliability and quality of the tracking of targets.

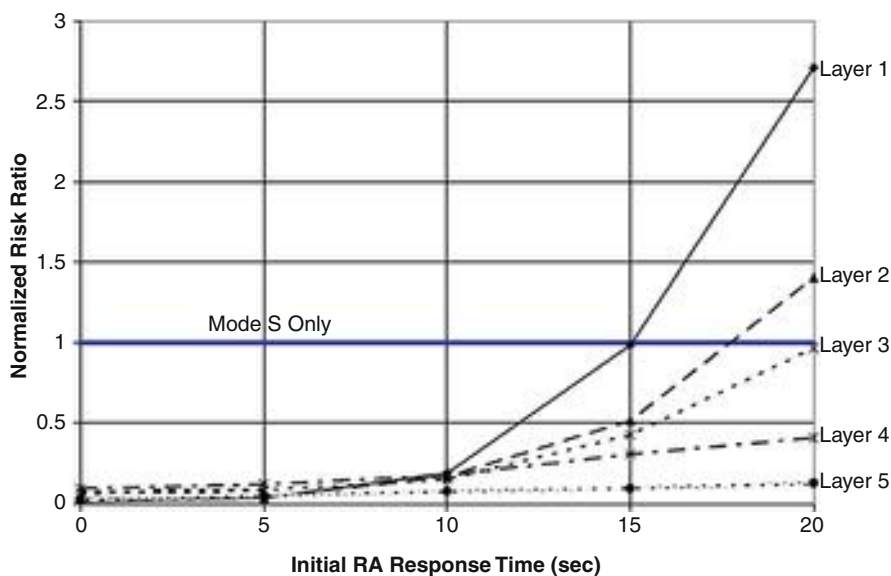


Fig. 77.9 Dependency of TCAS risk ratio from initial RA response time, derived for Global Hawk (Final Air4All Report for ‘UAV insertion into General Air Traffic GAT Follow On Contract’ 2009)

- These RAs require a response within a specified time, at an acceleration of a specified strength, to achieve a specified vertical rate. If for any reason the UAS cannot achieve this standard response the efficacy of the RA can be compromised.

The conclusion of (Unmanned Aircraft Systems 2010) on the certification of TCAS for UAS is as follows:

Certification of any UAS equipage with ACAS II should be conducted on a case-by-case basis for each airframe/equipment configuration.

It would need to be established that ACAS II surveillance was adequate for the purposes of collision avoidance, and that the UAS was able to comply with ACAS II RAs promptly and accurately.

77.8 Conclusions and Way Ahead

The proposed solution has the potential to solve the need for a cooperative collision avoidance capability needed to achieve preliminary integration of UAS into airspace classes A–C.

Aircraft operating in those airspace classes are required to carry an operational transponder to support ATC in providing separation service as well as the intruder to perform collision avoidance. Still, even with the transponder being mandatory equipment for operation in airspace classes A–C, there is a certain risk that not all intruders actually comply with this regulation, as pilots intentionally or unintentionally fly into airspace A–C without permission and proper equipage. The risk associated to this needs to be analysed. A suitable means could be to use ATC data to compare primary RADAR detection (sensitive to aircraft with a minimum RADAR cross section) with secondary surveillance RADAR detection (sensitive to aircraft equipped with a transponder).

As the proposed solution is limited to operation in airspace classes A–C, special procedures are required for transit in and out of those airspace classes for take-off and landing. These special procedures aim to ensure that no non-transponder equipped aircraft is operated in the vicinity of the UAS. Possible procedures could be to take-off and land through segregated airspace, the usage of primary RADAR coverage or to use chase aircraft for this transit phase. Implications on the operational flexibility of UAS need to be assessed. In order to comply with the EUROCONTROL requirement to ensure adequate ACAS II surveillance for the purposes of collision avoidance, the suitability of the UAS platform itself needs to be considered (i.e., UAS climb/descend performance, ACAS antennae placement). This is no different to certification for manned aircraft, where compliance with the same requirements must be demonstrated.

The requirement for prompt and accurate compliance with ACAS II RAs is an issue which needs to be covered by the design of a cooperative collision avoidance system based on TCAS together with the capabilities of the UAS FCS.

The demonstrated solution is considered an intermediate step towards a product level system enabling operation in airspace classes A–C that will be applicable to any UAS which complies with the defined requirements for TCAS installation as defined in Unmanned Aircraft Systems (2010).

References

- Demonstration video, http://www.cassidian.com/en_US/web/guest/barracuda1
- EASA certifies new “Autopilot/Flight Director” TCAS mode for A380, Airbus Press Release. (2009), <http://www.airbus.com/presscentre/pressreleases/press-release-detail/detail/easa-certifies-new-autopilotflight-director-tcas-mode-for-a380/>, 20 Aug 2009
- Final Air4All Report for ‘UAV insertion into General Air Traffic (GAT) Follow On Contract’, BAES-AS&FC-BDEV-PROP-RP-08258 Issue 1, 1 May 2009
- ICAO International Standards and Recommended Practices, Annex 10, Volume IV, Surveillance and Collision Avoidance Systems, 4th Edition, July 2007
- Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II), RTCA Do185, 2008-06-19
- Unmanned Aircraft Systems – ATM Collision Avoidance Requirements, Edition 1.3, EUROCONTROL, 2010-05-17
- WASLA/HALE Phase III, Studie: Nachweis von Technik und Verfahren für Teilnahme eines UAV am allgemeinen Luftverkehr (HALE III), BWB Vertrag E/LR3H/5A056/4F190, Abschlußbericht, Version: 1.0, September 2008

Test Flights to Demonstrate Effectiveness of a Ground-Based Detect and Avoid Integration Concept

78

Andreas Udovic, Hans de Jong, and Jürgen Vielhauer

Contents

| | | |
|--------|--|------|
| 78.1 | Introduction | 1874 |
| 78.1.1 | Background | 1874 |
| 78.1.2 | Structure of This Chapter | 1875 |
| 78.1.3 | The Problem | 1875 |
| 78.1.4 | The VUSIL Idea | 1876 |
| 78.2 | Aims of the Research | 1878 |
| 78.3 | Stepwise Approach | 1879 |
| 78.4 | First Realization: UAS Test Flights in Simulated Traffic | 1879 |
| 78.5 | Second Realization: UAS Test Flights with Real Intruder | 1882 |
| 78.6 | Ability Requirements for UA Pilots | 1886 |
| 78.7 | Safety Analysis | 1886 |
| 78.7.1 | Approach to Safety Analysis | 1887 |
| 78.7.2 | Effectiveness of “See and Avoid” | 1888 |
| 78.7.3 | Effectiveness of “Detect and Avoid” | 1890 |
| 78.7.4 | Conclusion of Safety Analysis | 1892 |
| 78.8 | Conclusions | 1892 |
| | References | 1893 |

A. Udovic (✉) • J. Vielhauer

Research and Development, DFS Deutsche Flugsicherung GmbH (German Air Navigation Services), Langen, Germany

e-mail: Andreas.Udovic@dfs.de; Juergen.Vielhauer@dfs.de

H. de Jong

Aeronautical Solutions, DFS Deutsche Flugsicherung GmbH (German Air Navigation Services), Langen, Germany

e-mail: Hans.de.Jong@dfs.de

Abstract

The VUSIL program (“Validierung von UAS zur Integration in den Luftraum” in German) has developed a concept for the integration of unmanned aircraft systems (UAS) into uncontrolled nonsegregated airspace, which is also used by manned civil aviation.

For the integration of UAS it is necessary for the unmanned aircraft (UA) pilot to be aware of the surrounding traffic in order to recognize and resolve possible conflicts. In the VUSIL program, the surrounding traffic was indicated on a display in the pilot ground control station (GCS), based on radar data provided by the Air Navigation Service Provider (ANSP).

In addition, the pilot was supported by means of a “detect and avoid” tool in detecting potential conflicts. This tool shows conflicts and possible avoidance maneuvers on a separate display. To validate the integration concept, two flight campaigns with two different UAS (helicopter and fixed-wing UAS), and several intruders were conducted.

In the first flight campaign, the intruders for the real flying UAS were simulated. In the second flight campaign, the intruder was a manned helicopter. The concept has been validated by the results of both flight campaigns and by a safety analysis.

The latter has addressed demonstrating the equivalent level of safety (ELOS) of “see and avoid” in manned aviation and “detect and avoid” as developed in the VUSIL program.

78.1 Introduction

This chapter is an elaboration of the publication of the VUSIL program in (Udovic et al. [2011](#)).

78.1.1 Background

UAS are new kinds of aircraft, which have great potential for conducting various kinds of tasks. In comparison to manned aircraft, the pilot of a UAS is not onboard the aircraft. He is controlling the unmanned aircraft (UA) from the ground. UAS have been mainly developed for the military environment to conduct dull, dirty, and dangerous (3D) missions. Therefore most of the civil UAS operations considered are spin-offs from military aviation, as for instance surveillance operations. In particular pipeline surveillance, border control and maritime or environmental observations are the most described civil applications. Dangerous applications like flying into volcanic ash or radioactive contaminations for measurements are also possible missions for civil UAS. The advantage of UAS conducting 3D missions is at the same time the challenge for the integration of UAS into civil aviation. In manned aviation, the pilot sitting in the cockpit is finally responsible for the safe conduction of the flight by scanning the vicinity. All civil aviation regulations are based on this premise.

78.1.2 Structure of This Chapter

At the beginning of this chapter, the problem related to the integration of UAS outside segregated airspace is explained. Based on this problem description, the VUSIL concept to solve this problem is described, followed by the results of the VUSIL program.

The VUSIL program contains two closely linked projects. In the VUSIL I project, the VUSIL idea was validated using a real flying UAS in combination with simulated surrounding traffic (Sect. 78.4).

In the VUSIL II project, the concept was validated with a real flying UAS in combination with real intruders. Approach and results of this project are provided in Sect. 78.5 of this chapter. Ability requirements for UA pilots are briefly discussed in Sect. 78.6. In Sect. 78.7 the equivalent level of safety is analysed by comparing effectiveness of “See and Avoid” and “Detect and Avoid”. Section 78.8 concludes this chapter.

78.1.3 The Problem

Based on the pilot not being onboard the aircraft, various problems for integration into nonsegregated airspace have to be solved. The problems can be divided into three areas:

1. Technical problems: For controlling an aircraft from the ground, a data link connection between the ground control station and the aircraft must be established. This data link connection must be stable, safe, and secure. At any moment of time, the pilot must have the possibility to interact with the aircraft. To handle the aircraft safely, he needs to have all relevant information in the ground control station. In particular, the status of the aircraft, involving for example the position, flight attitude, and altitude of the aircraft, and the status of the engine must be displayed. Therefore, a two-way data link is required. To disburden the pilot, several tasks, such as stabilizing the aircraft during flight, can be integrated into the avionics of the aircraft. Alternatively, a fully autonomous UAS (in which all of the pilot’s tasks have been taken over by onboard systems) could be designed.
2. Operational problems: One of the major operational problems is that the pilot of the UAS is not onboard the aircraft. Therefore, he cannot look outside to avoid conflicts with other traffic or obstacles according to the “see and avoid” principle. It is also not possible for the pilot to see signals from other aircrafts such as anti collision lights (ACL) or signals from the ground. At the moment, no technical equipment is available to support the pilot on the ground with this information. If such equipment were available, its certification for use in aviation is another hurdle. This leads to the third problem.
3. Regulatory problems: Rules and regulations for the safe integration of UAS into civil airspace do not exist at the moment. Certification rules for UAS requirements for a safe and secure data link or rules for the necessary information exchange between ground control station and UA are not defined yet.

Neither are requirements for a “detect and avoid” system comparable to the “see and avoid” principle in manned aircraft available. The International Civil Aviation Organization (ICAO) is addressing these problems (see ICAO 2011). On a national level, only some high level requirements are available in certain states (see UK CAA 2010). In Germany, flying UAS outside the visual line of sight (VLOS) of the pilot is currently not allowed. Only in segregated areas such flights can be conducted (see Beschluss des Bundesrates 816/09 2009).

Both VUSIL projects focused on the operational problem of “see and avoid.” In addition, some regulatory problems related to legal aspects and safety analyses were addressed.

78.1.4 The VUSIL Idea

To solve the “see and avoid” problem for the UA pilot, radar equipment of DFS Deutsche Flugsicherung GmbH (DFS) has been used (see Udovic and Vielhauer 2010). The radar data are available and can be forwarded to the ground control station to display the position of the UA and the surrounding traffic. A “detect and avoid” tool can be triggered based on these data as well. As in manned aviation, the pilot is in permanent contact with ATC via radiotelephony. Figure 78.1 shows the principle of the solution.

The UAS is equipped with a common transponder (mode A/C or mode S). The Air Navigation Service Provider (ANSP) can detect the UAS and the surrounding traffic with standard radar equipment. The corresponding data can be sent via a landline to the ground control station of the UA. In the ground control station, traffic

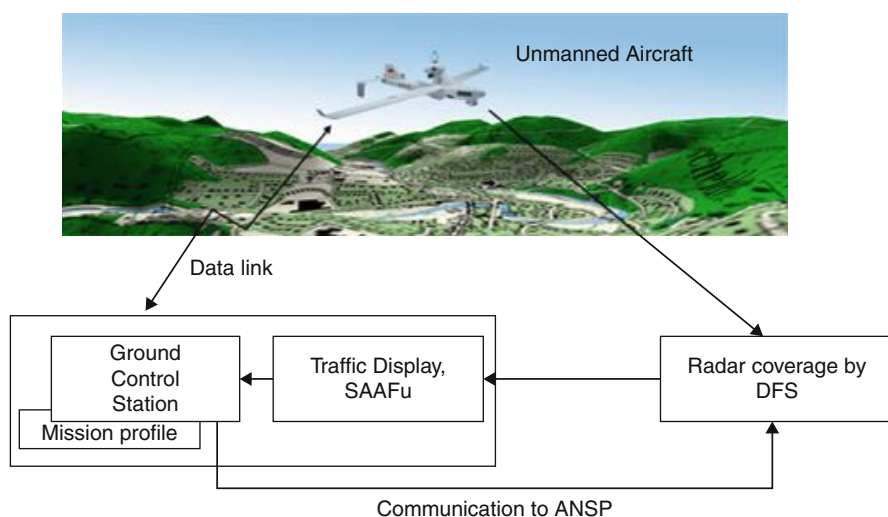


Fig. 78.1 Principle of the “detect and avoid” solution developed by VUSIL



Fig. 78.2 Traffic display “Phoenix” (see Heidger 2011)

is displayed on a radar screen comparable to those used by air traffic controllers. A “detect and avoid” tool (SAAFu) can as well be fed by radar data. Based on this information the pilot can avoid the surrounding traffic, and the UAS can be safely integrated into non-segregated airspace. The applied principle is that UAS are separated from other traffic by at least 0.5 NM laterally or 500 ft vertically. These values were based on considerations involving inter alia radar accuracy (see Eurocontrol 2007).

Figure 78.2 shows the traffic display based on the Phoenix air traffic management system of DFS. Figure 78.3 shows the “detect and avoid” display “SAAFu” developed by the German company ESG GmbH.

For successful application of the VUSIL concept, some preconditions have to be fulfilled:

- All traffic must be equipped with transponder, as for example, in a Transponder Mandatory Zone (TMZ).
- The whole concerned airspace is covered by secondary radar surveillance (in the concept developed, primary radar is not applicable due to missing altitude information and insufficient accuracy).
- Separation minima of 0.5 NM horizontally/500 ft vertically;
- If possible, collision avoidance based on existing visual flight rules (VFR).
- A traffic display is available in the control station.
- A “detect and avoid” assistance system is available.

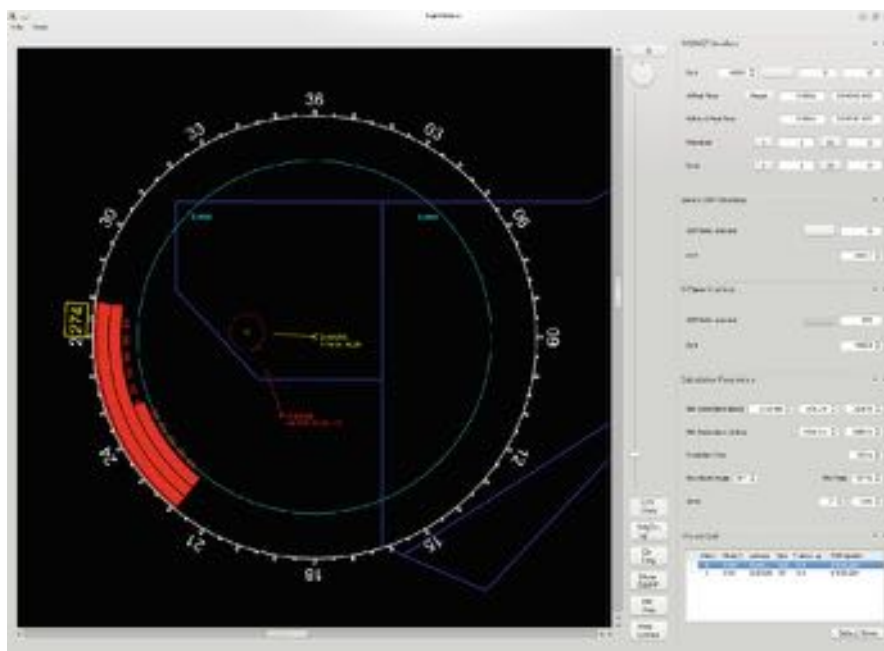


Fig. 78.3 “Detect and avoid” display (“SAAFu”)

The preconditions show that the concept is only valid for some special areas, such as TMZs. However, it can be used to gain experience with UAS to develop solutions for safe integration into a wider subset of nonsegregated airspace.

78.2 Aims of the Research

As an onboard “detect and avoid” system is not available, the idea of using a “ground-based system” was the starting point of the program. The aim of the program was to validate this idea by conducting flight trials and performing a safety analysis to investigate feasibility of safe integration of UAS into nonsegregated airspace.

In detail, the following four aims were defined for the program:

1. Development and optimization of the integration concept (conditions, equipment, and procedures)
2. Evaluation of the equivalent level of safety (ELOS) between “see and avoid” and “detect and avoid”
3. Possible certification rules of the ground-based system for UAS flights outside segregated airspace
4. Analysis of the ability requirements for UA pilots

78.3 Stepwise Approach

In a first step, a ground-based “detect and avoid” concept for the safe integration of UAS had to be developed. To that end, the DFS infrastructure had to be analyzed, and conditions for the safe integration had to be determined.

In a second step, the necessary infrastructure (radar data, simulated surrounding traffic, traffic display, “detect and avoid” tool) for the validation of the concept had to be defined. For conducting the flight trials, the infrastructure had to be built up in the test area.

To validate the concept, flight trials were conducted using certain procedures to maintain separation between the UAS and surrounding aircraft. In the first phase, the surrounding traffic was simulated (VUSIL I). This phase was sponsored by the German Ministry of Transport.

In the second phase, financed by the German Federal Police in cooperation with the Ministry of Internal Affairs, a real intruder participated in the flight trials (VUSIL II).

Based on the results of the flight trials and discussions with UA pilots, pilots of manned aircraft, air traffic controllers, and aviation experts, a safety analysis was conducted to identify and assess safety relevant differences between the UAS flight trials and normal manned flights in similar circumstances.

78.4 First Realization: UAS Test Flights in Simulated Traffic

Based on VUSIL’s integration concept, first flight trials were conducted. The UAS “LUNA” produced by the German supplier of air vehicles and aviation equipment EMT was flying during these trials. To investigate the avoid capability of the concept, simulated traffic around the LUNA was displayed on the traffic display as well as by means of a “sense and avoid” tool. Based on this information the task of the pilot of the real flying UAS was to avoid the displayed simulated traffic.

To conduct the flight trials, the necessary infrastructure was defined. As test environment, a restricted area (ED-R 138) near the military airfield Manching (in southern Germany) was selected. The radar data (of the real flying UAS) from DFS was sent via Universal Mobile Telecommunication System (UMTS) to the UAS ground control station. As foreseen in the concept, the LUNA was equipped with a transponder. The infrastructure was built up in Manching. For the data collection, questionnaires, interviews, and a structured debriefing were prepared. Furthermore, the radar data and the simulated traffic were recorded and analyzed after the flight trials were finished. For a detailed description of the infrastructure and methodology, see Udovic and Vielhauer [2010](#).

In June and July 2008, 13 flight trials were performed in ED-R 138, as planned. Radar data and data from the interviews and discussions were collected. As an example, Fig. [78.4](#) shows a flight track of the LUNA (flight number 5).

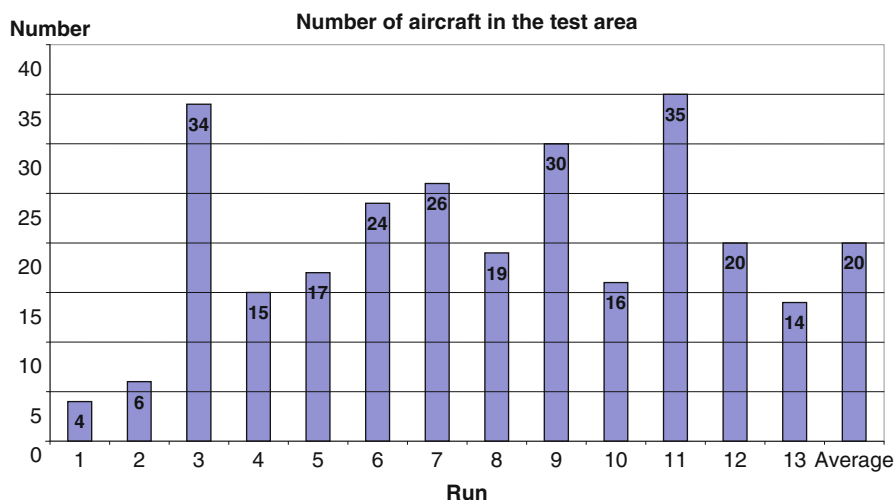


Fig. 78.5 Number of simulated aircraft during the flight trials

In most of these cases, the traffic density was very high (see Fig. 78.5). During flight number 3, there were 34 simulated aircraft, and during flight number 9, there were 30 simulated aircraft in the test area. In flight number 6, the amount of traffic was high as well (24 aircraft). During flight 13, technical problems were the reason for the separation infringement. For that reason, the flight was aborted.

Discussions with the UA pilots and the participants in the flight trials yielded the following results:

- The traffic display with the surrounding traffic was rated as very helpful for situational awareness. It was however not easy for the pilots to interpret the information on the display. To fulfill this task, more extensive training is needed.
- The “detect and avoid” tool was also rated as helpful. Especially for avoidance maneuvering this assistance was considered very supportive. Possible solutions to avoid the conflict are displayed to the UA pilot. The software could however be enhanced by integrating vertical avoidance maneuvers. More training could also enhance the effectiveness of the tool.
- The combination of both support tools was rated as very helpful. A better integration into the UAS ground equipment could further enhance their effectiveness.
- The necessary instruction and training of the UAS pilot based on the investigated integration concept was discussed. The participants agreed on a combination of a pilot licensing and elements of air traffic controller training.
- All in all, the tested integration concept was rated positively. The participants agreed that the concept could be a way to integrate UAS into nonsegregated airspace. However, training of the pilots must be taken into account, and the technical integration must be enhanced.

The results of the safety analysis workshop supported the results of the interviews and discussions. In addition, the following results were obtained:

- Compared to common manned aircraft, detection of a UAS by pilots of the surrounding traffic was rated negatively by the participants. UAS may be very small, and therefore their visibility is poor. This could generate problems if the pilots of the surrounding traffic are responsible for conflict avoidance; see the rules of the air – ICAO 2005. Special lights or a special color could increase the visibility of the UAS.
- The performance and steering of the UAS were also rated negatively. The speed of the LUNA is low, and especially during strong wind conditions, the LUNA is inert. This could create problems in avoiding collisions.
- The detection of the surrounding traffic was judged positively by the participants. The “detect and avoid” tool and the traffic display were very helpful to detect and avoid the surrounding traffic. Especially the range of the radar display was rated positively. The UA pilot can detect the traffic in advance and is able to react before the situation becomes critical.
- The instruction and training of UA pilot was not clear for the participants. Hence, a detailed analysis of the required abilities of the UA pilot was proposed. Based on these requirements, a concept for the instruction and training should be developed.

The results of the first flight trials indicate that the proposed integration concept is a viable way to integrate UAS into the air traffic outside restricted airspace. The following aspects have been indicated to further validate the integration concept:

- Test the concept in a more realistic traffic environment – with real instead of simulated intruders.
- Identify and rank the required abilities of UA pilots and develop training requirements.
- Compare the effectiveness of “detect and avoid” as developed in the integration concept with the analogue process “see and avoid” in manned aviation.

These topics are investigated in the succeeding phase.

78.5 Second Realization: UAS Test Flights with Real Intruder

The flight trials were organized by the NATO department “Centre of Excellence for Operations in Confined and Shallow Waters” and set up in segregated airspace (ED-R 10, Todendorf-Putlos), in which a UAS (“LUNA” or “NEO” by the Swiss company SWISS UAV) is flying together with a manned helicopter operated by the German Federal Police. The manned helicopter (indicated by “BPOL”) was the intruder for the UAS. Based on the VUSIL concept, the UAS should avoid conflicts with the helicopter crossing its flight path. The aim of this validation step was to prove the concept and to gather enough data for the safety analysis of the “detect and avoid” concept.

Figure 78.6 shows the technical realization of the flight trials in VUSIL II. radar data were taken from two sources: The main radar source was a local radar

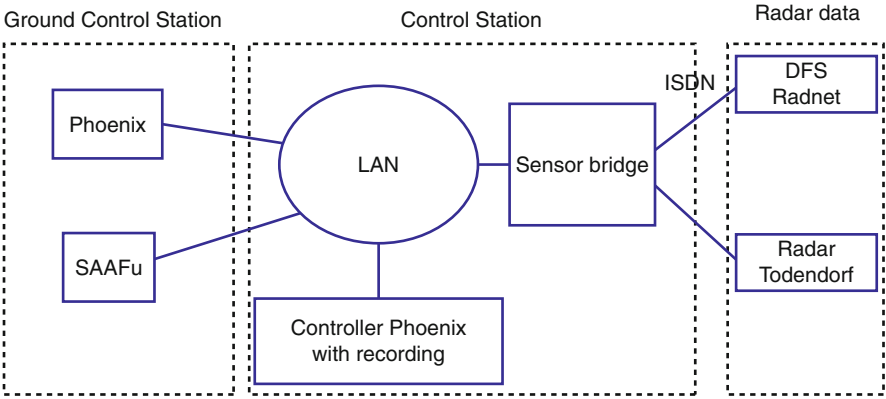


Fig. 78.6 Technical realization of the VUSIL concept

Table 78.2 Flight trials in April 2010

| No. | Flight duration | BPOL | UAS LUNA | UAS NEO | Civil helicopter |
|-----|-----------------|------|----------|---------|------------------|
| 1 | 01:52 | X | X | | |
| 2 | 01:45 | X | X | | |
| 3 | 02:22 | X | X | | |
| 4 | 01:08 | X | X | | |
| 5 | 00:26 | | X | X | |
| 6 | 00:19 | | X | X | |
| 7 | 01:34 | X | X | X | |
| 8 | 00:35 | X | | X | X |
| 9 | 00:40 | | | | |
| 10 | 01:37 | X | X | | |
| 11 | 01:40 | X | X | X | |
| 12 | 00:35 | X | | X | |
| 13 | 00:30 | X | | X | |

antenna in Todendorf, in the neighborhood of where the flight trials were conducted; the second source was the DFS Radnet. The DFS Radnet is a broadband local area network (LAN), which provides all radar data from the different DFS radar antennas. For the VUSIL II flight trials, radar data from the Todendorf area were selected. The radar data from both sources were sent to the control station of the flight trials. A sensor bridge merged the radar data and sent them to a LAN. The Phoenix radar display and a SAAFu in the ground control station of the UAS were connected to this LAN. In the ground control station a Phoenix display to supervise and record the flight trials was connected to the LAN.

The first flight trials were conducted in October 2009. Due to windy weather, only a few flights could be accomplished. Therefore, in April 2010, the flight campaign was continued. Between 22 and 29 April 2010, 13 flights were completed (see Table 78.2). The table also indicates the participating UAS and intruder.

Table 78.3 Distribution of the approaches of UAS and intruder

| No. | Direction of the approach | No. with LUNA | No. with NEO |
|-------|---------------------------|---------------|--------------|
| 1 | Approach from ca. 90° | 17 | 2 |
| 2 | Approach from ca. 270° | 11 | 2 |
| 3 | Approach from ca. 315° | 7 | 1 |
| 4 | Approach from ca. 45° | 5 | 1 |
| 5 | Approach from ca. 225° | 9 | |
| 6 | Approach from ca. 135° | 10 | |
| 7 | Approach from ca. 180° | 5 | |
| 8 | Approach from ca. 360° | 7 | |
| Total | | 71 | 6 |

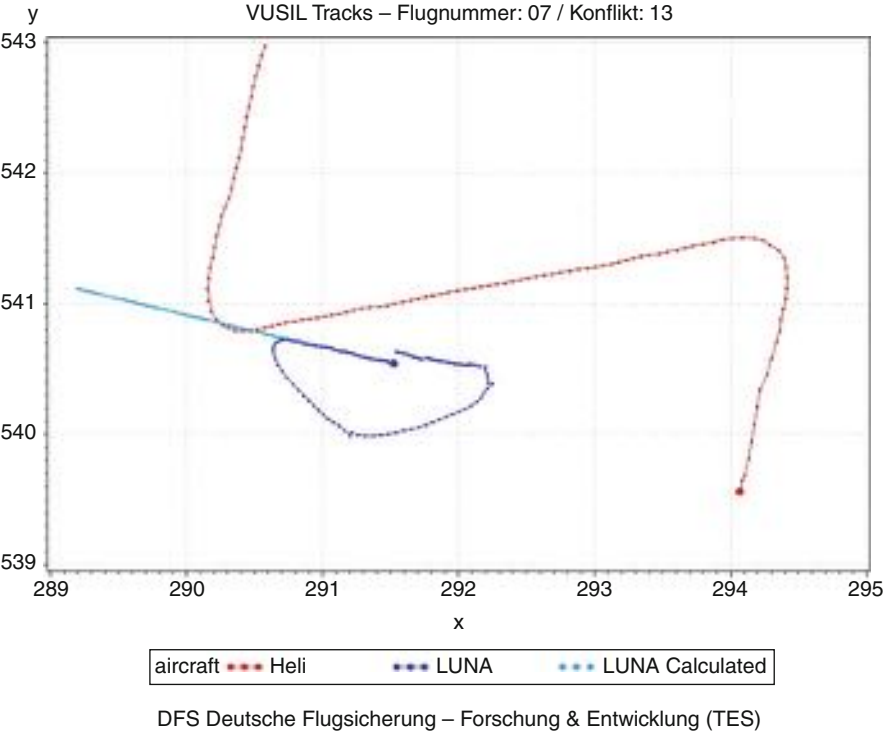


Fig. 78.7 Example of an approach between UAS and intruder

Altogether 77 approaches of a UAS and the intruder were recorded and analyzed (71 of LUNA and intruder, and 6 of NEO and intruder). The approaches of the intruder and the UAS were conducted from different directions, see Table 78.3.

Figures 78.7 and 78.8 provide plots visualizing an exemplary data analysis. For all 77 approaches, the flight paths of the intruder (red) and the UAS (dark blue) were displayed example in Fig. 78.7. If the pilot of the UA decided to fly a

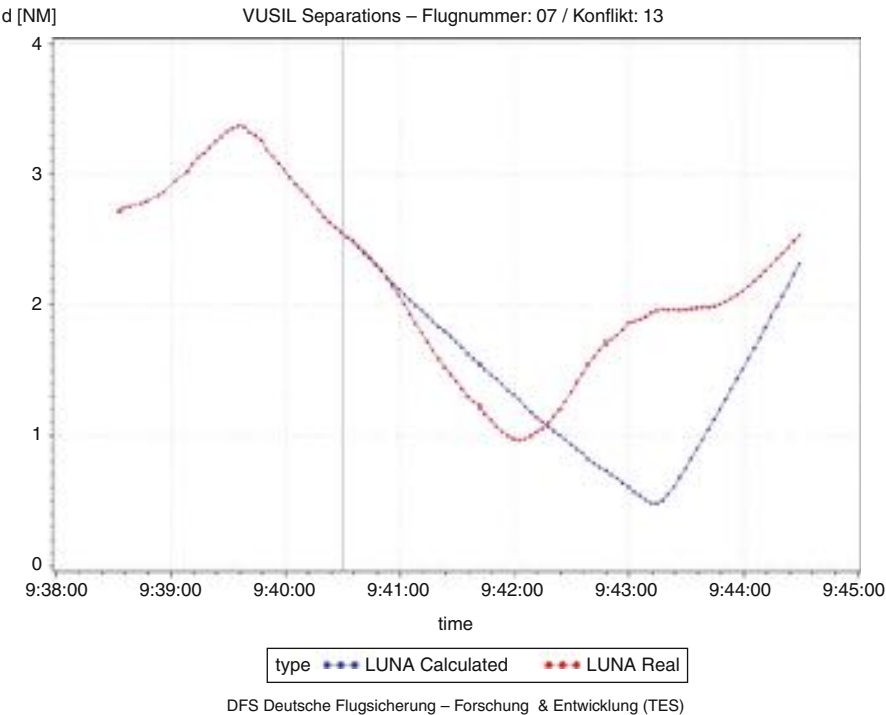


Fig. 78.8 Real distance and extrapolated (calculated) distance between UAS and intruder for one approach

maneuver to avoid the intruder, the extrapolated flight path without this maneuver was calculated (light blue). Figure 78.8 shows the result of the calculation of the real distance between the UAS and the intruder (red). In a second calculation, the closest proximity between the intruder’s flight path and the UAS’s flight path without avoidance maneuver was assessed (shown in blue).

In 22 of the approaches to the LUNA, the calculated closest distance between the LUNA and the intruder would have been less than the separation minimum of 0.5NM. In all cases, the LUNA pilot started an avoidance maneuver to prevent an infringement of the separation minimum. In all but one cases, the pilot increased the distance between the intruder and the LUNA and accomplished the separation minimum. In one case, the distance was below the separation minimum. A deeper analysis showed that due to a software problem, the SAAFu display was frozen and presenting false data to the pilot. The pilot reacted according to the false information and conducted a wrong maneuver.

The six approaches involving the NEO were analyzed in the same manner. In four of the cases, the calculated distance for the extrapolation of the flight path if no maneuver had been initiated was less than the separation minimum. In three cases, the pilot conducted an avoidance maneuver successfully; in one case, the maneuver

failed. The analysis of this case showed a general problem of the integration of UAS outside segregated airspace: The speed of the UAS used in the flight trials and especially the NEO was very low. The intruder was flying at around 90 knots. Due to the speed difference between UAS and the intruder the UA pilot could not always prevent an infringement of the separation minimum even though he reacted in an appropriate manner.

The flight trials also highlighted another problem for the integration of UAS. Both UAS were small and painted in white color. Both UAS were equipped with a transponder, so the two pilots in the helicopter could see the UAS on the airborne collision avoidance system (ACAS) display in the cockpit. Although the pilots knew the position of the UAS based on the ACAS display, in most of the cases, they could not see the UAS visually. Flying under VFR rules implies that the pilot has to follow the rules of the air (see ICAO 2005). Under defined circumstances, the pilot is committed to avoid traffic by flying an avoidance maneuver. This implies that the pilot must be able to see the opponent. In case of a UAS, this may not be possible. For these cases either the “rules of the air” (ICAO) have to be adapted or the visibility of the UAS has to be significantly enhanced.

78.6 Ability Requirements for UA Pilots

Another aim of the VUSIL II project was to conduct a first investigation of the ability requirements of UA pilots. This investigation was conducted by the German Aerospace Centre (DLR) in Hamburg. For the investigation, the participating UA pilots filled in the Fleishman Job Analysis Survey (Fleishman and Reilly 1992), which was adapted by DLR. This questionnaire is also used for the evaluation of the ability requirements of air traffic controllers or pilots (Eißfeldt and Heintz 2002). Altogether, six UA pilots filled in the questionnaire. Table 78.4 shows the ranking of the 15 most important ability requirements.

Spatial orientation resulted as the most required ability. Stress resistance and staying power were ranked secondly and thirdly. In comparison to similar results for controllers and pilots, the results of the UA pilots show higher similarity to the results of pilots, although the values of the required abilities for UA pilots were in tendency lower than for pilots of manned aircraft.

Due to the low number of UA pilots participating in the study, the results are not fully comparable to the results of the pilots and controllers. Nevertheless, first hints for the required abilities for UA pilots have been found.

78.7 Safety Analysis

One of the aims of the VUSIL II project was to demonstrate an equivalent level of safety (ELOS) for the developed concept of integration. To this end, it needs to be investigated whether the collision risk for the UAS in the integration concept is equal to or smaller than for VFR flights in manned aviation.

Table 78.4 List of the 15 most important ability requirements for UA pilots

| |
|-----------------------|
| Spatial orientation |
| Stress resistance |
| Staying power |
| Visual imagination |
| Multitasking |
| Reading maps |
| Teamwork |
| Reliability |
| Self-control |
| Speed of perception |
| Selective attention |
| Situational awareness |
| Speed/reaction |
| Decision making |
| Self-assurance |

This section begins with explaining the approach to answer this question, which involves comparing effectiveness of “detect and avoid” to effectiveness of “see and avoid.” Subsequently, the effectiveness of both of these processes is estimated, based on the flight trials conducted in the project, expert judgement, and results from the literature. At the end of this section, results are summarized, and conclusions are drawn.

78.7.1 Approach to Safety Analysis

In order to demonstrate an ELOS, the “detect and avoid” process, as applied in the integration concept, is decomposed into a subprocess regarding detection of traffic in the vicinity of the UAS and a subprocess regarding recognition and resolution of possible conflicts. The effectiveness of these subprocesses is quantified using conditional probabilities, multiplication of which then results in effectiveness of the “detect and avoid” process. The “see and avoid” process in manned aviation is analogously decomposed and quantified, which then allows a comparison. This approach is indicated in Fig. 78.9.

- In conducting the above comparison, the following assumptions have been made:
1. The flights of the UAS take place in a context where equipment performs as may be expected from operationally (instead of experimentally) used equipment. Consequently, equipment failures would occur as rarely as in a normal operational context. Flight trials conducted in the VUSIL project that were affected by equipment failures that would be rare in an operational context were therefore excluded. This approach is appropriate as the project aimed to evaluate the integration concept, not yet the equipment components.
 2. Manned flights are conducted by single pilots, as is not unusual for VFR flights.

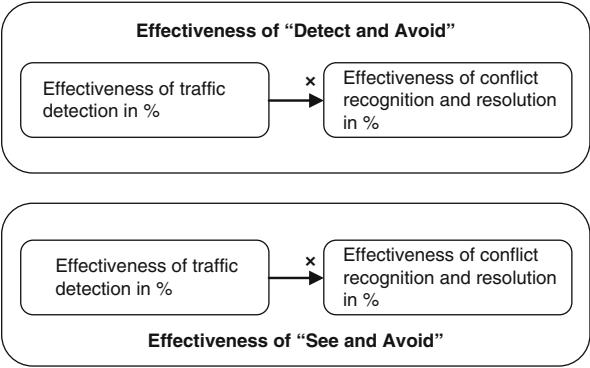
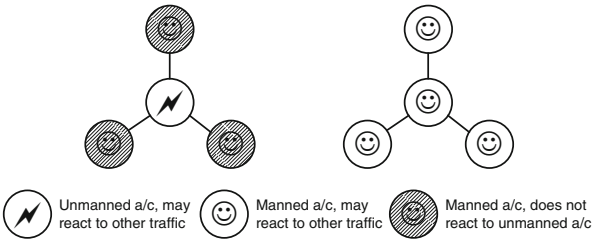


Fig. 78.9 Decomposition of “detect and avoid” and of “see and avoid” into traffic detection and conflict recognition and resolution

Fig. 78.10 Conservative assumption: manned aircraft do not react to unmanned aircraft (in manned aviation, both aircraft in a conflict may react to each other)

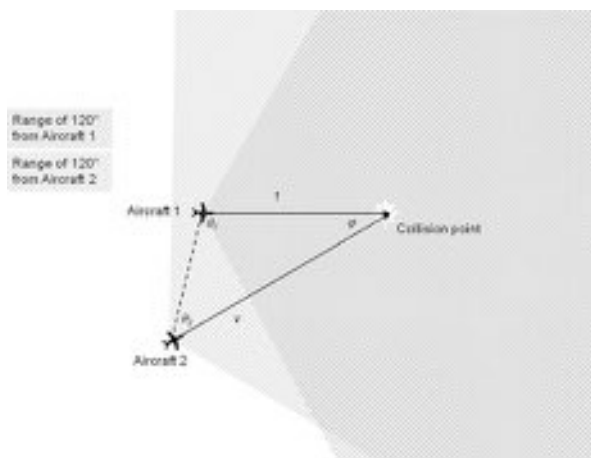


3. For the rest, the comparison has been made in a “conservative” manner, that is, in cases of uncertainty or lacking information, aspects of “detect and avoid” have been assessed pessimistically, whereas such aspects for “see and avoid” have been assessed optimistically. If ELOS can be still shown using a conservative approach, this gives increased confidence in the validity of this result. An example of this conservative approach is the assumption that UAS are principally not detected by manned aircraft (e.g., due to their small size), whereas in conflicts between manned aircraft, the pilots of both aircraft can detect each other and initiate a conflict resolving maneuver. In other words, it is assumed that conflicts involving UAS can only be resolved by the UA pilot. This conservative manner in comparing “detect and avoid” to “see and avoid” is indicated in Fig. 78.10.

78.7.2 Effectiveness of “See and Avoid”

The estimates regarding effectiveness of “see and avoid” are based on Andrews 1991, which reports on flight tests conducted to address the ability of pilots to visually acquire aircraft on a near collision course. More specifically, in the flight tests conducted, 24 on the average relative experienced pilots flew one of two

Fig. 78.11 Influence of conflict geometry on whether an aircraft on a conflicting course can be seen by a pilot from the other aircraft in a range of 120° around the forward direction



cross-country courses in a Beech 33 (Bonanza). During each of these flights, three encounters were planned with a Cessna 421, flying at an altitude differing 500 ft with that of the Beech Bonanza and directed to achieve as little horizontal distance as possible – usually a few tenths of a mile. The safety pilot also present in the Bonanza recorded the point of time at which the subject pilot detected the Cessna (see Andrews 1991).

The investigation restricted to encounters in which the Cessna was within 60° horizontally and 10° vertically from the forward direction of the Bonanza as the paper indicate that this is the angular region normally searched by pilots and considered most likely to contain aircraft on a threatening course.

A main result of Andrews 1991 is that only in 36 of 64 of such encounters, the subject pilot visually acquired the Cessna. On the basis of this result, the effectiveness of traffic detection by one pilot is estimated to be 56%. On the basis of the 2nd assumption that a VFR aircraft is flown by a single pilot, this is the effectiveness of traffic detection from the perspective of one of the manned aircraft involved in a conflict.

Depending on the geometry, the other aircraft may also detect the first aircraft and in this way contribute to the resolution of a conflict. The degree to which the pilots of both aircraft are able to do so depends on the geometry of the conflict. In particular, it depends on the relative flight directions and speeds of the aircraft whether the pilot in aircraft 1 can see aircraft 2 within a horizontal angle θ_1 smaller than 60° from the forward direction and vice versa. This is illustrated in Fig. 78.11.

Figure 78.12 indicates that depending on the proportion v between the speeds of aircraft 2 and aircraft 1, for an angle ϕ of less than 60° between the aircraft's horizontal directions, the pilot(s) in *at most* one of the aircraft can see the other aircraft within a range of 120° around the forward direction. For values of the angle ϕ between 60° and 120° , one or both of the aircraft are visible within this range

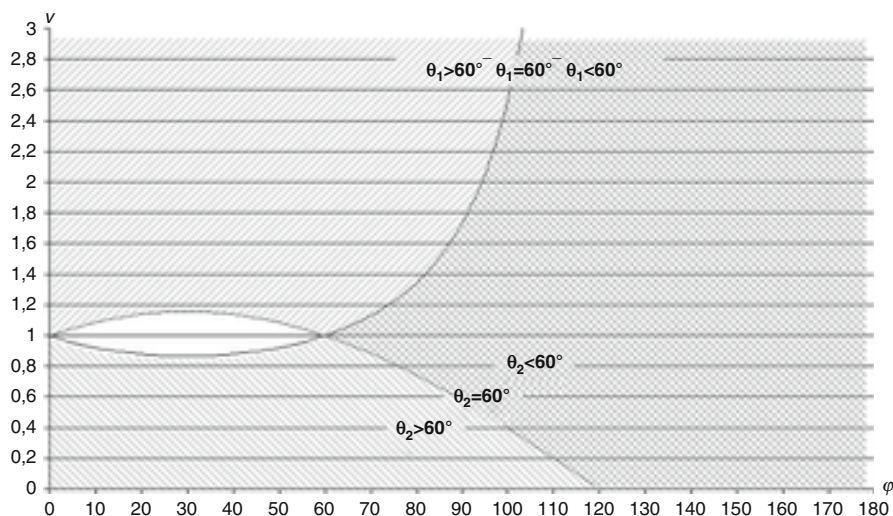


Fig. 78.12 How do the angles θ_1 and θ_2 indicating the horizontal angle of aircraft 2 as seen from aircraft 1 and vice versa depend on the angle ϕ between the aircraft's directions and the proportion v between the speeds of aircraft 2 and aircraft 1?

to the pilots of the other aircraft, and for angles over 120° , both aircraft are visible within this range.

In line with the conservative approach, it is optimistically assumed that the conflict geometry is always such that the pilots of both aircraft could see the other aircraft as if they were within 60° of the forward directions. Under this assumption, the probability that at least the pilot of one aircraft detects the other aircraft is $1 - (1 - 0.56)^2 \approx 0.8$.

Under the also optimistic assumption that given successful detection of traffic, conflict detection, and resolution is always successful, the overall effectiveness of “see and avoid” is estimated to be 80%.

For further information of the “see and avoid” concept, see also BEKLAS 2002 and Air Traffic Safety Board ATSB.

78.7.3 Effectiveness of “Detect and Avoid”

In contrast to the aforementioned manned flight tests, the flight tests conducted in the VUSIL project are not suitable to derive effectiveness of the traffic detection subprocess: The UA pilots knew that during the flight tests, probably several conflicts would be encountered, and they were therefore intensively paying attention to other traffic.

Nevertheless, the effectiveness of traffic detection is judged to be much better than in the VFR context in manned aviation: All traffic in the neighborhood is

indicated on the radar screen as well as the “detect and avoid” display such that aircraft cannot be overseen due to the conflict geometry or factors impeding visibility (such as clouds, small size of the aircraft, or glare). Detection of traffic by the UA pilot only fails if:

- One of the systems (radar, data link, track system, “detect and avoid” system, displays) necessary for providing the traffic picture fails (which under assumption 1 of operational grade equipment would be relatively improbable).
- Surrounding traffic is not equipped with a working transponder (which, under the assumption that surrounding traffic is equipped with a transponder is also improbable).
- The UA pilot does not pay sufficient attention to the radar screen and the “detect and avoid” display.

Regarding the last point, it was suggested to notify the UA pilot regarding new and/or conflicting traffic by means of acoustic signals. All in all, the experts involved in the project judged that under the assumptions made, the effectiveness of traffic detection would be at least 99%.

The effectiveness of the UA pilot, supported by a radar screen and the “detect and avoid” tool, to recognize and resolve conflicts can be derived from the test flights conducted in the VUSIL project. During these test flights, a manned helicopter generated potential conflicts, which had to be recognized and resolved by the UA pilot. To safeguard the helicopter and the UAS, the helicopter was always flying 500 ft above the UAS, and the conflict resolving maneuvers were restricted to the horizontal plane. In this context, conflicts were situations in which the minimum horizontal separation of 0.5 NM would be infringed if the UAS would not change its direction. This notion of conflict corresponds with the near collision courses generated in the flight tests in Andrews 1991, briefly discussed above.

In the analysis of the data generated by the flight tests, it was investigated how many conflicts had been generated and which proportion was recognized and resolved successfully by the UA pilot, such that in the end no horizontal separation infringement occurred. Separation infringements due to system failures that would be improbable for operational grade equipment have been excluded. Under these assumptions, $N = 26$ conflicts occurred of which $n = 1$ was not recognized and resolved successfully. Estimating the conditional probability p that given detected traffic, the recognition and resolution of a conflict fails, using the maximum likelihood (ML) method (see a textbook on statistics, e.g., Sivia and Skilling 2006) provides the following result:

$$p_{\text{ML}} = n/N = 1/26 \approx 4\%.$$

An alternative, conservative, estimate concerns the largest value of the conditional probability p for which the event that of 26 conflicts at most one conflict is not recognized and resolved successfully is still credible. Taking for “still credible” the value of 5%, the conservative estimate $p_{\text{conservative}}$ follows from the equation:

$$(1 - p_{\text{conservative}})^{26} + 26p_{\text{conservative}}(1 - p_{\text{conservative}})^{25} = 0.05,$$

which results in the following conservative estimate:

$$p_{\text{conservative}} \approx 0.22.$$

Combination of the estimates regarding the effectiveness of conflict recognition and resolution with the judgement that traffic is detected in at least 99% of the cases yields the following estimates for the effectiveness of “detect and avoid”:

ML estimate of “detect and avoid” effectiveness: $0.99(1 - 0.04) \approx 95\%$

Conservative estimate of “detect and avoid” effectiveness: $0.99(1 - 0.22) \approx 77\%$

78.7.4 Conclusion of Safety Analysis

Based on the flight tests conducted in the VUSIL II project and the assumptions made, the effectiveness of “detect and avoid” estimated by the maximum likelihood method is 95%, which is considerably larger than the analogously estimated effectiveness of 80% for “see and avoid,” based on flight tests reported on in the literature.

A statistically more conservative approach determining the value for which the result of at most 1 of 26 conflicts not being recognized and resolved successfully is still credible, results in an overall estimate of the “detect and avoid” effectiveness of 77%.

In other words, although the maximum likelihood estimate indicates that “detect and avoid” is considerably more effective than “see and avoid,” the results and the number of flight test performed within VUSIL are not yet large enough to state with 95% confidence that the effectiveness of “detect and avoid” is at least 80%.

The following activities should further strengthen these results:

- Detailed estimate of the effectiveness of traffic detection by the UA pilot and available equipment (radar, data link, and further developed “detect and avoid” tool).
- Extend the set of flight tests to increase confidence further.
- Verify whether the ELOS principle is sufficient for regulatory approval.

78.8 Conclusions

The VUSIL projects show that the developed concept for the integration of UAS outside segregated airspace is valid. The positive results of the flight trials with the real flying UAS in a context of simulated traffic could be validated further in the flight trials with the real intruder. It must however be considered that the concept is not valid for a complete integration of UAS into nonsegregated airspace but that it is limited to uncontrolled airspace where a transponder is mandatory.

The proposed separation minimum of 0.5 NM horizontally/500 ft vertically between the UAS and the intruder was considered sufficient during the flight trials.

High speed differences between the UAS and intruder appear to be a problem. Possibly, the concept must be supplemented by a minimum speed requirement for UAS.

Also problematic is the visibility of UAS for pilots in a manned aircraft. Due to the fact that the UAS are normally smaller than manned aircraft, their visibility is low. Perhaps additional requirements like eye-catching coloring or strobe lights are necessary to enhance the visibility of the UAS. Additionally, the rules and regulations for conflict avoidance between aircraft must be addressed. Due to the technical support of the UA pilot, it could be reasonable that UAS are always responsible for the conflict resolution. This is not in line with today's ICAO rules and regulations, so that the regulations possibly have to be adapted.

The safety analysis indicates an equivalent level of safety for "see and avoid" and "detect and avoid" as used in the VUSIL program. Even under conservative assumptions, the "detect and avoid" capabilities of a UAS pilot are comparable with the capabilities of a pilot in a manned aircraft. An important assumption underlying this statement is however that the "detect and avoid" equipment is developed to similar standards as equipment in manned aviation.

The next step for the VUSIL program is to realize UAS flights outside segregated airspace and with operational equipment. To that end, the SAAFu software must be programmed according to certification standards. Also the technical solutions for the transmission of the radar data must pass a safety analysis to fulfill the high standards of manned civil aviation.

Acknowledgments The two projects in the VUSIL program were lead and conducted by the Research and Development Centre of DFS. The first project (VUSIL I) was sponsored by the German Ministry of Transport ("Bundesministerium für Verkehr, Bau und Stadtentwicklung"); the second project was financed by the German Federal Police in cooperation with the Ministry of Internal Affairs ("Bundesministerium des Inneren"). Both projects would not have been realized without their support.

Partners in the VUSIL I project were:

- DFS Deutsche Flugsicherung GmbH (DFS)
- ESG GmbH (responsible for the SAAFu)
- Federal Office of Defence Technology and Procurement (BWB)
- EMT GmbH (manufacturer of the UAS "LUNA")

Based on the results of the VUSIL I project, the VUSIL II project was set up. The project started in October 2009 and finished at the end of 2010. The partners of VUSIL I were again participating in the project. In addition the German Aerospace Centre (DLR) with the "Institute of Flight Guidance" in Braunschweig and the "Institute of Flight Psychology" in Hamburg, as well as Swiss UAV providing a helicopter type UAS (called "NEO"), completed the team. As external partner, the NATO department "Centre of Excellence for Operations in Confined and Shallow Waters (CEO CSW)" organized the flight trials. From the contracting body, the Federal Police, a helicopter participated as intruder for the UAS. The German Federal Aviation Office (LBA) attended the projects as observer.

References

Air Traffic Safety Board (ATSB), Limitations of the See-and-Avoid Principle, Australian Transport Safety Bureau (ATSB), Reprinted November 2004, http://www.atsb.gov.au/media/32918/limit_see_avoid.pdf

- J.W. Andrews, Unalerted Air-to-Air Visual Acquisition, Report No. DOT/FAA/PM-87/34, Project Report ATC-152, Lincoln Laboratory, Massachusetts Institute of Technology, 26 November 1991
- BEKLAS, Erkennbarkeit von Segelflugzeugen und kleinen motorisierten Luftfahrzeugen, Abschlußbericht, FE-Nummer L – 6/2002 – 50.0300/2002, im Auftrag des Bundesministerium für Verkehr, Bau- und Wohnungswesen (2002)
- Beschluss des Bundesrates 816/09 (2009), Verordnung zur Änderung der Luftverkehrs-Ordnung und anderer Vorschriften des Luftverkehrs, 18.12.2009
- DFS Deutsche Flugsicherung GmbH, The multi-radar tracking and air situation display system “PHOENIX” developed by DFS, (2008). http://www.dfs.de/dfs/internet_2008/module/worldwide_solutions/deutsch/worldwide_solutions/download/phenix.pdf
- H. Eißfeldt, A. Heintz, Ability requirements for DFS controllers – Current and future, in *Staffing the ATM System – The Selection of Air Traffic Controllers* (S. 13–24), ed. by H. Eißfeldt, M.C. Heil, D. Broach (Ashgate, Aldershot, 2002)
- Eurocontrol, Eurocontrol specification for the use of military unmanned aerial vehicles as operational air traffic outside segregated airspace, Edition Number 1.0, 26.07.2007, Brussels, 2007
- E.A. Fleishman, M.E. Reilly, *Handbook of Human Abilities. Definitions, Measurements, and Job Task Requirements* (Consulting Psychologists Press, Palo Alto, 1992)
- R. Heidger, The Phoenix White Paper, DFS Deutsche Flugsicherung GmbH, System House, 2011
- ICAO, Annex 2 “Rules of the Air” 10th edn. (International Civil Aviation Organization, Montreal, 2005)
- ICAO, Unmanned Aircraft Systems (UAS), Circular #328, Montreal, 2011
- D. Sivia, J. Skilling, *Data Analysis: A Bayesian Tutorial*, 2nd edn. (Oxford University Press, 2006)
- A. Udovic, J. Vielhauer, DFS Deutsche Flugsicherung GmbH, VUSIL project: results and further development, in *UAS Unmanned Aircraft Systems, The Global Perspective*, UVS International, 7th Edition, 2009/2010 (Blyenburgh&Co, Paris, 2010)
- A. Udovic, H.H. de Jong, J. Vielhauer, Validation of unmanned aircraft systems’ integration into the airspace (VUSIL I and II). *J. Aerosp.* 4(2), 1216–1227 (2011)
- UK CAA, CAP 722, *Unmanned Aircraft System Operations in UK Airspace – Guidance* (Civil Aviation Authority UK, London, Norwich, 2010)

Allistair Moses, Matthew J. Rutherford, and Kimon P. Valavanis

Contents

| | | |
|--------|---|------|
| 79.1 | Introduction | 1896 |
| 79.1.1 | Why RADAR?..... | 1898 |
| 79.1.2 | Existing Challenges | 1898 |
| 79.2 | RADAR Hardware | 1899 |
| 79.2.1 | Continuous Wave vs. Pulsed Operation | 1899 |
| 79.2.2 | Transmit Frequency Selection | 1902 |
| 79.3 | The Proposed RADAR Sensor | 1904 |
| 79.3.1 | Generation 2 Microwave Section | 1904 |
| 79.3.2 | Generation 2 Analog Section | 1905 |
| 79.4 | FSK CW Simulation and Ranging Experiments | 1909 |
| 79.4.1 | Azimuth-Enabled RADAR Evaluation | 1913 |
| 79.4.2 | RADAR Targets | 1914 |
| 79.4.3 | Remarks | 1922 |
| 79.5 | Collision Detection and Evasion | 1928 |
| 79.5.1 | Collision Detection | 1928 |
| 79.5.2 | Collision Avoidance Maneuver Classes | 1930 |
| 79.5.3 | Collision Avoidance Maneuver Planning | 1933 |
| 79.5.4 | Airspace Simulation Software | 1939 |
| 79.5.5 | Collision Avoidance Algorithm Evaluation | 1941 |
| 79.5.6 | Remarks | 1943 |

A. Moses (✉)

University of Denver, Denver, CO, USA

e-mail: Allistair.Moses@gmail.com

M.J. Rutherford

Department of Computer Science, University of Denver, Denver, CO, USA

e-mail: mjr@cs.du.edu

K.P. Valavanis

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

79.6 Conclusion 1951

References 1952

Abstract

Unmanned aircraft systems (UAS) have become increasingly prevalent and will represent an increasing percentage of all aviation. These unmanned aircraft are available in a wide range of sizes and capabilities and can be used for a multitude of civilian and military applications. However, as the number of UAS increases, so does the risk of midair collisions involving unmanned aircraft. This chapter aims to present one possible solution for addressing the midair collision problem in addition to increasing the levels of autonomy of UAS beyond waypoint navigation to include preemptive sensor-based collision avoidance. The presented work goes beyond the current state of the art by demonstrating the feasibility and providing an example of a scalable, self-contained, RADAR-based, collision avoidance system. The work can be divided into two sections: RADAR hardware and collision avoidance algorithms. The technology described herein can be made suitable for use on a miniature (maximum takeoff weight <10 kg) UAS platform. This is of paramount importance as the miniature UAS field has the lowest barriers to entry (acquisition and operating costs) and consequently represents the most rapidly increasing class of UAS.

79.1 Introduction

Modern unmanned aircraft systems (UAS) are available in a wide range of sizes from the palm-sized “Black Widow” to the 39.8 m wingspan Global Hawk (AeroVironment 2013; U.S.DOD 2012). Most modern UAS are generally limited to autonomously following preprogrammed waypoints or executing preprogrammed commands while under the supervision of a human operator. Only recently are UAS beginning to operate with some level of independence from preprogrammed commands as demonstrated by Shen et al. (2011) in which a miniature quadrotor unmanned aerial vehicle (UAV) autonomously constructs a map of an indoor environment. However, these exercises have largely been limited to computer vision and laser rangefinder-based systems operating in an indoor environment. One of the next steps for UAV technology development is to enable unmanned systems to perform the same tasks in an outdoor environment with a similar level of safety and autonomy. This level of autonomy is challenging in many ways due to the different sensor arrangements required when transitioning to the outdoor environment. This sensor paradigm shift is typically necessitated by the fact that many of the sensors used in indoor laboratory conditions are not well suited for the demands of UAV operation in an outdoor environment due to the longer ranges, higher speeds, and environmental factors normally encountered. More importantly, the threats to safe operation are dramatically different in regulated airspace wherein the risk of a collision between a manned aircraft and a UAS exists.

There are a number of existing solutions to address the midair collision problem. These solutions are typically divided into two categories which may be combined

to form a complete collision avoidance solution: transponders and noncooperative sensors.

There are a number of transponder solutions including *Traffic Collision Avoidance System* (TCAS), *Portable Collision Avoidance System* (PCAS), *FLight ALARM* (FLARM), and *Automatic Dependent Surveillance and Broadcast* (ADS-B) (FAA 2012b; ZAON 2007; Flarm.Inc 2008; ADS-B 2011). The TCAS transponders (currently required in the United States for turbine-powered aircraft with more than 10 seats) function by interrogating other TCAS transponders to determine heading, velocity, and altitude information while simultaneously responding to TCAS interrogations from opposing aircraft (FAA 2011). If a collision is detected, the TCAS automatically determines a collision avoidance maneuver and presents the information to the pilot via a cockpit display. TCAS is an effective solution for manned aircraft; however, the cost of a typical installation is often prohibitive for many general aviation craft, which comprise a substantial portion of the aircraft population (AOPA 2013b; SAI.inc 2011). To address this issue, PCAS receivers have been made available for under 2000 USD (ZAON 2011). PCAS receivers achieve this cost reduction, in part, by the elimination of the transmit functionality present within TCAS transponders. In essence, PCAS receivers listen for TCAS signals and determine the risk of collision without transmitting their own location. In this fashion, PCAS-equipped aircraft can actively avoid TCAS-equipped aircraft. However, the passive nature of PCAS makes it unsuited for avoiding collisions with other PCAS-equipped aircraft, nor does it allow TCAS-equipped aircraft to detect or avoid PCAS-equipped aircraft. There are a wide range of flight regimes present throughout aviation including many cases where aircraft routinely operate in close proximity to each other without the risk of a midair collision. Manned gliders are useful examples of this type of flight. The FLARM transponder was developed to provide a collision avoidance solution for aircraft operating under these conditions. FLARM utilizes barometric pressure and GPS data to estimate the host vehicle's location and velocity vector. It then broadcasts this data to the airspace while listening for position and velocity information from other FLARM devices. If a midair collision situation arises, the FLARM transponder alerts the pilot who can then take action if necessary. The final transponder system, ADS-B, is currently poised to supersede secondary surveillance RADAR (which is based on RADAR transponders located onboard aircraft) as the primary air traffic control method. ADS-B operates in a manner similar to FLARM but adds additional features such as weather and terrain data broadcast by ground stations (ADS-B 2011). While ADS-B use is not currently required, the FAA aims to make it mandatory for all aircraft by January 1, 2020 (FAA 2010). The common theme with all the transponder solutions is the need for cooperative infrastructure if collisions are to be successfully avoided.

In contrast to the cooperative nature of transponders, sensor-based collision avoidance otherwise known as "sense-and-avoid" (SAA) systems typically do not require cooperation between aircraft to effect a useful collision avoidance solution. There are a number of prototype systems utilizing a wide range of sensor technologies. These technologies include acoustic sensors, laser rangefinders, and camera systems (both visible and infrared wavelengths). This dissertation describes the development of an additional SAA system based on radio detection and ranging

(RADAR) (Selex [2011](#); SARA.inc [2011a, b](#); Watanabe et al. [2007](#)). More specifically, this dissertation investigates the use of RADAR technology for the detection and identification of miniature (MTOW <10 kg) UAVs. Additionally, vertical and horizontal collision avoidance maneuvers are compared and a computationally efficient method for collision avoidance is proposed and evaluated.

79.1.1 Why RADAR?

RADAR has a well-established history dating back to Heinrich Hertz's initial experiments in 1886. The publication of Heinrich Hertz's book documenting his experiments forms the foundation of modern radio concepts (Hertz [1893](#)).

The use of RADAR for active detection of objects was first demonstrated by Christian Hulsmeyer who created a device he called the Telemobiloscope. This device, patented in 1904, transmitted broadband radio frequency (RF) energy in a wide pattern while observing reflections with a narrow-beam antenna which could be directed in order to make observations of a specific area. Originally intended for use in an anti-collision role in a maritime environment, the device never saw widespread acceptance (Bauer [2005](#)).

RADAR was significantly advanced during the 1930s and 1940s in the United Kingdom, Germany, and the United States. The demands of World War II facilitated many innovations, and RADAR technology advanced rapidly during this period. Previously, only limited to ground installations, RADARs soon became small enough to install in aircraft (Goebel [2013](#)). After World War II RADAR applications in the civilian domain continued to increase, particularly with regard to civilian aviation. This trend continues to this day (Britannica [2013](#)).

RADAR-based sensors have a number of distinct advantages when compared with other sensor types such as LASER rangefinders, computer vision/focal plane arrays, and acoustic sensors. Unlike optical systems, RADARs are not affected by sunlight, smoke, fog, dust, and other factors that typically affect optical wavelengths. Furthermore RADARs typically have improved directionality and range characteristics when compared with acoustic systems; additionally, RADAR systems can be used onboard aircraft with high levels of acoustic noise and can detect aircraft with little to no acoustic noise emissions (something that is increasingly important as the number of UAS using electrical propulsion increases). At present, no other noncooperative sensor type provides all these capabilities (Tait [2006](#)).

79.1.2 Existing Challenges

There are a number of existing challenges that preclude the widespread adoption of collision avoidance RADARs on UAS. The first challenge is the physical volume and mass of many RADAR systems as most currently available RADAR systems are far too large for widespread UAS implementation.

Furthermore, most airborne RADAR systems currently installed on UAS are designed for use in air-to-ground modes either for mapping or for target detection/tracking purposes. An example of such a system is ImSAR's NanoSAR line of synthetic aperture RADARs (SAR) (ImSAR.inc 2011; Grumman 2013). This dedication to the air-to-ground operating mode often inhibits effective use when attempting to use the RADAR for air-to-air operations. This is because additionally air-to-air RADARs must often incorporate ground clutter rejection features not typically present in dedicated air-to-ground RADARs.

Power consumption is another area that must be addressed before widespread collision avoidance RADAR becomes a reality. Many airborne RADARs require hundreds of watts of electrical power to operate. This is often unavailable on small UAS. There are, however, some small RADAR systems with low electrical power consumption. The NanoSAR B utilizes 30 W of electrical power.

Finally, one of the more important issues that must be addressed is the prohibitive cost of many airborne RADAR solutions. Existing systems can easily cost upwards of 100,000 USD and greatly exceed the cost of UAS platforms with MTOWs <50 kg (ImSAR.inc 2011).

The remainder of this chapter is organized as follows: Sect. 79.2 describes a number of critical decisions that must be made when designing a RADAR; in addition, some example hardware designed at the University of Denver is presented. Section 79.4 details the ability of the RADAR prototype to localize targets and experimental results are presented. Section 79.5 describes the collision detection and pulsed evasion method, and simulations are provided to demonstrate the efficacy of the proposed approach.

79.2 RADAR Hardware

There are a number of critical design decisions that must be made regarding any RADAR architecture used in various aircraft scales. These decisions have a direct impact on operating parameters such as maximum range, minimum range, range resolution, input power requirements, and required timing resolution. This section will detail the critical design decisions and provide justifications for each.

79.2.1 Continuous Wave vs. Pulsed Operation

RADAR systems can generally be divided into two broad categories depending on the nature of the transmit energy: pulsed and continuous wave (CW). In a pulsed RADAR architecture, the transmitter generates RF pulses of a specified duration and transmits this energy towards the region under observation. These pulses then reflect off an object (if present) and a portion of the energy returns to the receive antenna. The range to the target object is determined by the following equation:

$$R = \frac{cT}{2} \quad (79.1)$$

Table 79.1 Example pulsed
RADAR performance

| Parameter | Value (m) |
|-------------------|-----------|
| Unambiguous range | 7,494.8 |
| Range resolution | 79.6 |
| Minimum range | 74.9 |

where the range, R , is defined in terms of the speed of light, c , and the elapsed time between the transmission of an individual pulse and its reception, T .

The rate at which these pulses are transmitted is referred to as the pulse repetition frequency (PRF). In pulsed RADAR systems, any subsequent pulse must be transmitted after the arrival of the previous pulse. Failing to do so will lead to range ambiguities. Therefore, the PRF has a direct impact on the maximum distinguishable range (otherwise known as the unambiguous range), R_{unamb} , which is described by

$$R_{unamb} = \frac{c}{2(PRF)} \tag{79.2}$$

Pulsed RADARs are typically limited in the minimum detectable range as the receive circuitry is deactivated during the transmit pulse, since close-range reflections (i.e., high-power reflections) would saturate and possibly destroy the highly sensitive receiver electronics. The lower bound on this minimum detectable range is defined by

$$R_{min} = \frac{c(T_{tx} + T_{config})}{2} \tag{79.3}$$

where T_{tx} is the duration of the transmit pulse and T_{config} is the time required to enable the receiver. Thus, for moderate pulse widths, the minimum detectable range can easily be in the region of tens of meters.

The range resolution of pulsed RADARs, ΔR_{min} (in the absence of more advanced techniques, e.g., intra-pulse modulation and pulse compression), is shown below:

$$\Delta R_{min} = \frac{cT_{tx}}{2} \tag{79.4}$$

Given (79.1)–(79.4) above, a theoretical RADAR system with a minimum T_{tx} of 500 ns, a T_{config} of 11 ns, and a PRF of 20 kHz will have the properties described in Table 79.1.

There are a large number of UAS with dimensions significantly smaller than the critical parameters: minimum range and range resolution. Furthermore, the design of a pulsed RADAR system with acceptably small minimum range and resolution characteristics is prohibitively expensive. This is due to the high power needed during the short pulses and the fast timing requirements of such a system.

Continuous wave (CW) RADARs operate by continuously illuminating a target object with RF energy. Unlike pulsed RADAR systems, CW RADARs tend to operate using lower overall power levels (up to three magnitudes lower power for similar range performance (Wallace et al. 2010)) and determine range to

objects using some form of frequency modulation as opposed to pulse timing. This allows for simpler circuitry due to the elimination of tight timing requirements. Additionally, CW RADARs are capable of providing arbitrary range resolutions and no limit on the theoretical minimum range.

The simplest CW RADAR configuration, Doppler RADAR, broadcasts a continuous signal towards the area of interest. If an object is present in the area of interest, a portion of the transmit energy is reflected back towards the receiver where it is then down-converted (multiplied in the frequency domain) to produce the frequency of interest, otherwise known as the “beat frequency” or “intermediate frequency” (IF) and is described by

$$F_{\text{beat}} = F_{tx} - F_{rx} \quad (79.5)$$

where F_{rx} is the frequency of the received signal and F_{tx} is the frequency of the transmitted signal. F_{rx} is determined by the radial velocity v of the target object relative to the RADAR system (assuming colocated transmit and receive antennas). F_{rx} is generally defined by

$$F_{rx} = F_{tx} \left(\frac{1 + \frac{v}{c}}{1 - \frac{v}{c}} \right) \quad (79.6)$$

Simplifying (79.6) for terrestrial applications where generally $v \ll c$ results in

$$F_{\text{beat}} \approx 2v \frac{F_{tx}}{c} \quad (79.7)$$

Unmodulated CW RADARs have no means to determine the range between the RADAR antenna and the target object. Furthermore, unmodulated CW RADARs are incapable of detecting stationary targets as F_{beat} results in a direct current (DC) signal which is lost due to the alternating current (AC) coupling between the RF front end and the amplifier stages. Furthermore, most CW RADAR front ends typically produce a DC signal regardless of the presence of a target due to leakage between the receiver and transmitter.

Nevertheless, a number of methods exist for range measurement in CW RADARs and are generally divided into two categories: frequency-modulated continuous wave (FMCW) and stepped frequency continuous wave (SFCW) (of which frequency-shift-keyed continuous wave (FSK CW) is a narrowband subset). FMCW RADARs operate by modulating the transmit frequency, typically in a triangular fashion. Due to the finite speed of light, F_{tx} will be offset from F_{rx} by an amount proportional to the range. In general terms, the range between the RADAR antenna and the target object is defined by

$$R = \frac{2cT F_{\text{beat}}}{BW} \quad (79.8)$$

where T is period of the frequency modulation and BW is the bandwidth. Unlike unmodulated CW RADARs, FMCW RADARs are capable of detecting both the

Table 79.2 CW RADAR comparison

| Parameter | Doppler | FMCW | FSKCW | X-band |
|----------------------------|------------------|------------------------|--------|--------|
| Range measurement | – | Yes | Yes | K-band |
| Velocity – range ambiguity | – | Yes | No | |
| # of receiver channels | 1 | 1 | 2 | |
| Bandwidth requirements | Single frequency | 500 MHz | 5 MHz | |
| Modulation type | – | Triangular or sawtooth | Square | |

presence and range of stationary targets. However, background clutter such as the terrain is also present in the output data.

FSKCW RADARs address this issue by replacing the linear, triangular modulation of FMCW RADARs with a square-wave modulation of the transmit frequency. F_{beat} is then sampled synchronously with the modulation. The transmit frequency modulation combined with the synchronous sampling of the received signal effectively produces two unmodulated CW RADAR channels separated by a few MHz (BW): IF_A and IF_B . (79.7) indicates that for any given moving target, F_{beat} will be marginally different for IF_A and IF_B . This marginal difference in frequency appears the same when used for velocimetry purposes but is fundamental for ranging as it manifests itself as a phase shift in the time domain. The range to the target object can then be inferred from the phase difference between IF_A and IF_B ; see (79.9) where ϕ_n is the phase of the IF_n (Skolnik 1981):

$$\text{Range} = \frac{c(\phi_A - \phi_B)}{4\pi BW} \tag{79.9}$$

FSKCW RADARs offer the same benefits of other CW RADAR types, namely, no minimum range and excellent range resolution, in addition to the benefits offered by unmodulated CW RADAR types (excellent ground clutter rejection and low overall transmit power requirements). However, the maximum practical range (in the absence of transmit power and receiver sensitivity limitations) is defined by the range corresponding to a 180° phase shift. As with pulsed RADAR systems, this range is defined as the unambiguous range and is described, in this case, by (79.10) (Skolnik 1981):

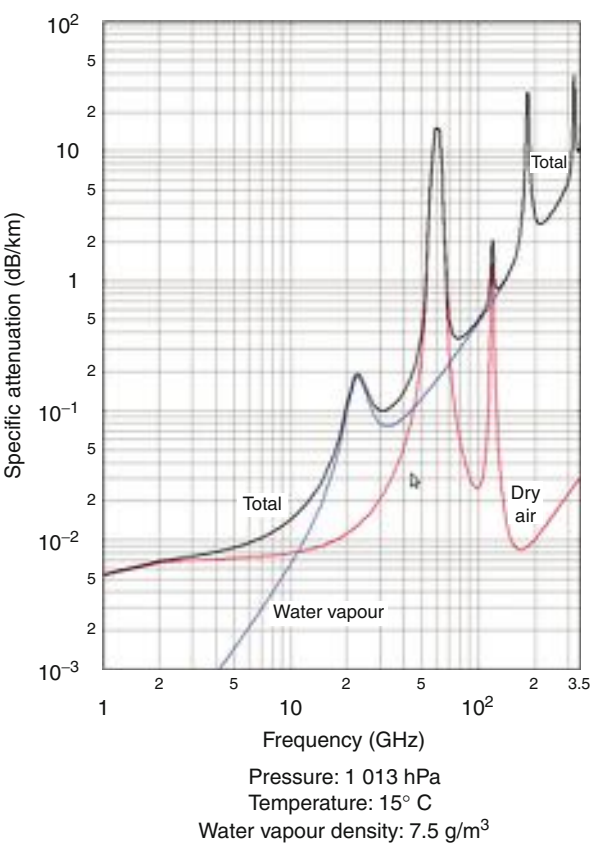
$$R_{\text{unamb}} = \frac{c}{2BW} \tag{79.10}$$

A comparison of the various CW RADAR types discussed in this section is shown in Table 79.2.

79.2.2 Transmit Frequency Selection

The transmit frequency, F_{tx} , is selected based on a compromise between the following parameters: atmospheric absorption, ratio between the wavelength and

Fig. 79.1 Atmospheric absorption of RF energy (ITU 2012)



the target object size, availability of components, desired range resolution, desired antenna size, and processing speed requirements of the intermediate frequency.

Atmospheric absorption of microwave frequencies (see Fig. 79.1) plays a role in frequency selection since absorption increases dramatically as a function of frequency. Countering this effect requires a substantial increase in power and consequently the size of most RADAR components. A cursory examination of Fig. 79.1 indicates that an X-band RADAR system (center frequency 10.5 GHz) will have a round-trip atmospheric absorption of 0.2 dB/km, while a comparable K-band RADAR system (center frequency 24 GHz) will have an absorption figure of over 2 dB/km. This effectively means that if all other system parameters are equal, then an X-band RADAR system will have a range 10 times greater than a comparable K-band RADAR. In practice, however, there are more contributing factors.

Transmit frequency selection also has a direct impact on the velocity resolution and data acquisition time. As characterized by (79.7), the Doppler shift, F_{beat} , is directly proportional to F_{tx} . If F_{tx} increases, maintaining the same velocity resolution requires that the ADC sampling rate also increase by the same proportion.

This has the effect of reducing the time required to acquire the relevant signal samples required for processing the RADAR data.

Microwave antenna sizes are also highly dependent on F_{tx} . A RADAR system designed for a mobile platform typically has limits placed on the amount of electrical power available and the overall volume and mass allocated to the RADAR payload. To reduce the required RF transmit power while maintaining the same range performance, an antenna with a narrow main lobe must be selected. However, to reduce the size of the main lobe, the physical antenna size must increase. An example of this relationship (in the case of a parabolic antenna) is described by (79.11) (Minoli 2009):

$$\Theta_{-3\text{ dB}} = \frac{k\lambda}{\text{diameter}} \approx \frac{70\lambda}{d} \quad (79.11)$$

where $\Theta_{-3\text{ dB}}$ is the -3 dB main lobe beamwidth, k is the steepness taper constant (typically 70 for parabolic antennas), λ is the operating wavelength, and d is the diameter of the antenna's reflector. Therefore, to achieve the same beamwidth, a RADAR operating at 10.5 GHz must have an antenna diameter approximately 2.3 times larger than a comparable RADAR operating at 24 GHz.

79.3 The Proposed RADAR Sensor

The University of Denver's research group has developed a number of generations of miniature RADAR sensors. A description of the earlier revisions of these systems can be found in (Moses et al. 2011). This section provides a description of the second-generation RADAR sensor hardware.

The second-generation RADAR prototype (see Fig. 79.2) makes several of significant improvements over the first generation with the most important improvement being the addition of FSKCW range detection. Other important improvements include the addition of variable gain amplifiers to expand the dynamic range of the analog section, variable cutoff frequency filters to facilitate a wide range of target velocities, and improved noise rejection due to revisions in the analog section layout. The specifications of the complete generation 2 RADAR sensor are shown in Table 79.3.

79.3.1 Generation 2 Microwave Section

The microwave section of the generation 2 RADAR sensor is comprised of a commercially available, FSKCW microwave front end and antenna module (Microwave Associates MDU4210) (Solutions 2013). This module is based on a dielectric resonant oscillator (DRO) and is electronically tunable using a varactor diode. In many commercial applications, these modules are used for motor vehicle traffic detection.

Fig. 79.2 Generation 2
RADAR sensor

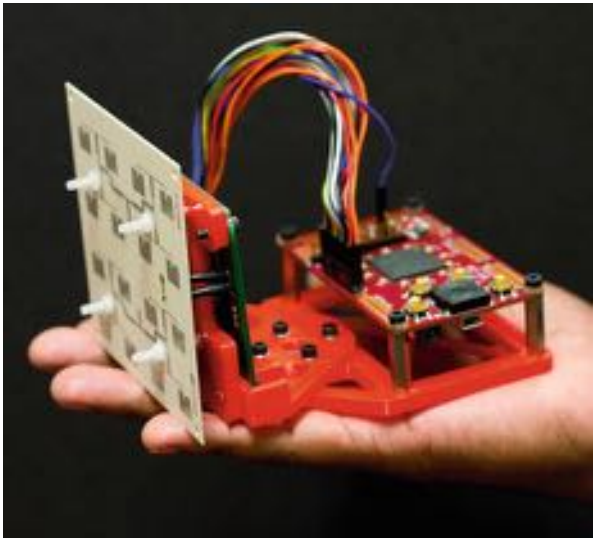


Table 79.3 Generation
2 specifications

| | |
|--------------------|-------------------|
| System mass | 150 g |
| System dimensions | 10 × 10 × 10 (cm) |
| Power consumption | 4.5 W |
| Input voltage | 5–6 VDC |
| Transmit frequency | 10.587 GHz |
| Transmit bandwidth | 5 MHz |
| Transmit power | 0.4 mW |

79.3.2 Generation 2 Analog Section

The generation 2 analog electronics (see Fig. 79.3) is comprised of three separate printed circuit boards of two categories: signal switching and signal amplification. Since FSK modulation requires rapid switching between two adjacent frequencies (typically separated by less than 5 MHz), two amplifier trains are required due to the finite bandwidth of the electronics. These amplification chains (located on separate PCBs to aid in serviceability and cross-talk reduction) function as follows: The RADAR IF signal first passes through a band-pass filter to strip away the higher-order mixing harmonics and the DC offset. The relatively weak IF signal is amplified by an operational amplifier-based non-inverting amplifier with a digital potentiometer feedback element. The output of this amplifier is AC coupled with another identical variable gain amplifier. This output signal is then passed through a variable cutoff frequency low-pass filter tuned by a digital potentiometer.

Up to this point, the hardware described is capable of obtaining target information while the target is within the antenna main lobe which is fixed relative to the RADAR sensor. The fixed main lobe of the first two generations is problematic since

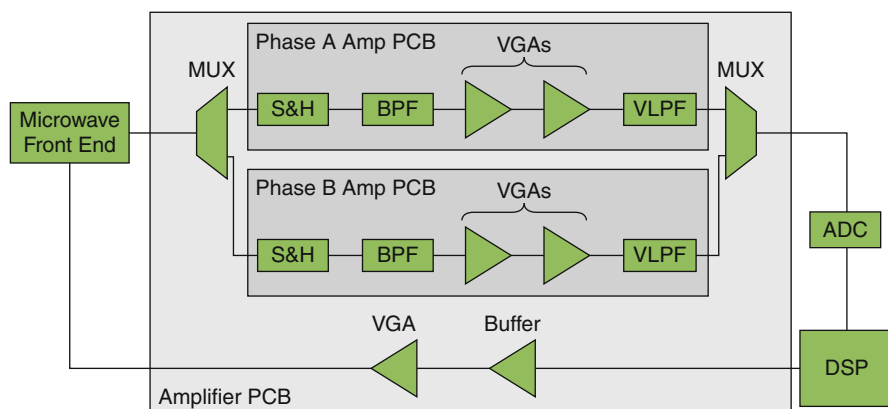


Fig. 79.3 Generation 2 electronics block diagram

they are not able to scan an area to provide a wider field of view. More importantly, they are not able to provide information regarding the relative bearing of any objects of interest.

There are a number of popular methods for obtaining target bearing information using RADAR systems including electronic beam steering, monopulse, interferometry, and mechanical scanning. Electronic beam steering utilizes an array of emitters with variable phase shifters to electronically steer the antenna main lobe. While there are a variety of methods that can be used to accomplish this, electronic beam steering is typically the most complex and most expensive solution for obtaining relative bearing information on RADAR targets (Tait 2006).

The second method, amplitude comparison monopulse (ACM) RADAR, is constructed using between 2 and 4 receive antennas arranged in a row for single axis measurement or quadrant for dual axis measurement, respectively. The target bearing information manifests as a difference in received signal strength in the antenna array. ACM RADAR types can obtain very accurate relative bearing information, but typically have limited observation angles due to high antenna gain. Therefore, ACM RADARs are most often used to track targets which have already been acquired with a search RADAR. The third method, phase comparison monopulse (PCM), is similar to the ACM architecture in that it requires an array of receive antennas. However, while ACM measures the difference in received signal strength, PCM RADARs measure the phase shift caused by the difference in the free space path length between the antenna array elements and the target of interest. Larger off-boresight angles increase the difference in the path length. A diagram of the PCM RADAR operation is shown in Fig. 79.4.

The final method, mechanical scanning, is when the entire antenna system is mechanically rotated throughout the observation angles in order to obtain target bearing information. Mechanical scanning may be used in combination with any of the above methods or may be used by itself. The primary disadvantage is the increased time required to obtain target bearing information. An additional

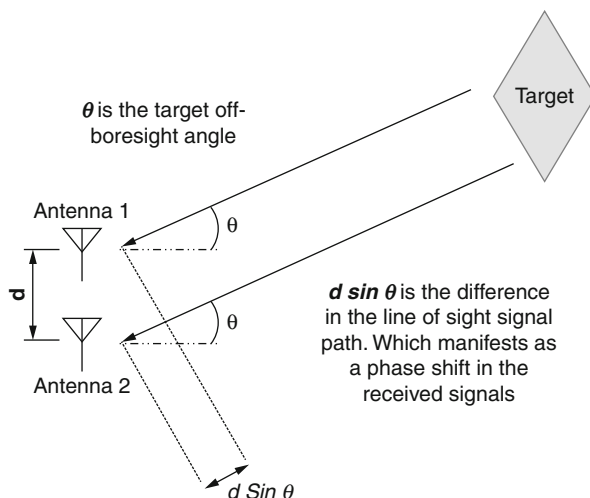


Fig. 79.4 Phase comparison monopulse diagram

disadvantage is the addition of a moving component (the antenna and the antenna gimbal/servo assembly). However, mechanical scanning allows the RADAR prototype described in this chapter to be utilized without completely redesigning the antenna, microwave, electronic, and software components. Mechanical scanning has an additional advantage in that narrow-beam (and consequently high-gain) antennas can be used without decreasing the sensor's field of regard as would be necessary using interferometry or monopulse architectures.

Given the above, mechanical scanning is used for the prototype to obtain target azimuth information. The result is the generation 2a hardware shown in Figs. 79.5 and 79.6. The antenna, microwave, and computing hardware of the generation 2a RADAR sensor are identical to the generation 2 hardware, the only addition being the mechanical components and electrical modifications required for mechanical scanning.

Mechanical scanning is accomplished by first separating the XMOS processor board (XMOS XC-1a) from the antenna, microwave, and analog components. This reduces the reciprocating mass by 50 g and improves antenna pointing agility. Additional benefits include the physical separation of the high-gain electronics from the high-frequency digital signals present on the XC-1a board.

The antenna, microwave, and analog components are then mounted on a plastic structure (henceforth referred to as the antenna mount) which provides mechanical stability and maintains the relative positions of the aforementioned components. This plastic structure is then directly attached (using two M3 screws) to a Hobbico CS-80 servo (1999) which serves to physically rotate the antenna mount through a 90° range in as little as 0.29 s. The addition of the servo increases the overall sensor mass by 150 g and the total power consumption by 1.3 W.

As with most radio control (RC)-type servos, the position feedback in the CS-80 is internal to the servo, and users provide position commands to the servo using

Fig. 79.5 Generation 2a
RADAR sensor front view
(with a 6'' (15.2 cm) size
reference)

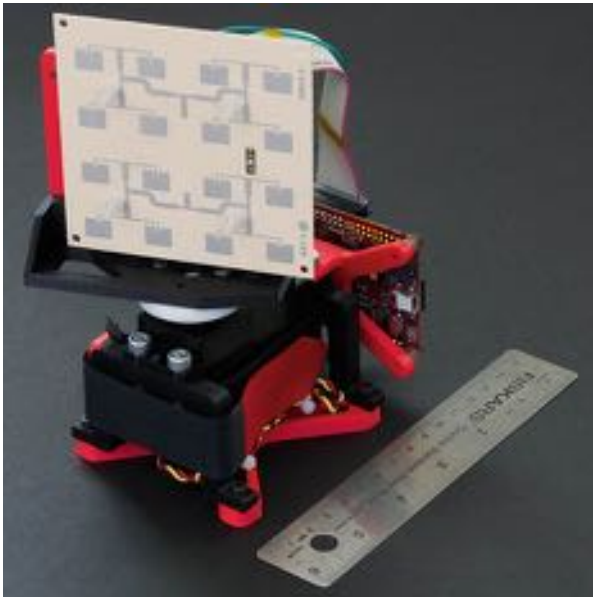
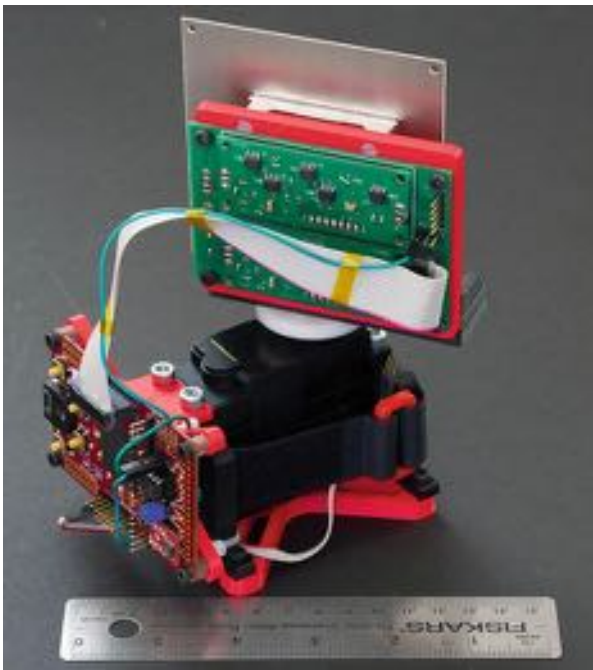


Fig. 79.6 Generation 2a
RADAR sensor rear view
(with 6'' (15.2 cm) size
reference)



pulse position modulation (PPM) signals with no feedback regarding the servo's actual position. Allowing the RADAR processor to monitor the position of the servo using the servo's own position encoder eliminates the need for an additional, external antenna position encoder and reduces the system complexity and overall mass. The CS-80 servo, as with most RC servos, utilizes a potentiometer directly coupled to its mechanical output shaft. The potentiometer is configured in such a way to produce an analog voltage directly proportional to the physical position of the shaft. Extracting this signal involves drilling a hole through the servo case and then routing a wire from the potentiometer's wiper to the RADAR processor's analog to digital converter (ADC). The 16-bit LTC1865 ADC used in the prototype has a voltage resolution of $76\text{ }\mu\text{V}$ which results in an antenna position resolution of 0.01° .

79.4 FSKCW Simulation and Ranging Experiments

As mentioned earlier, unmodulated CW (otherwise known as Doppler) RADARs are incapable of determining range. To address this limitation, some form of modulation must be added; the generation 2a RADAR hardware described in the previous section utilizes FSKCW modulation as described in Sect. 79.2. Range to the target is determined by comparing the phase of the two IF signals at the peak frequency. This process is illustrated in Fig. 79.7.

In practice, however, the phase measurements output by the RADAR hardware are noisy and, therefore, must be filtered prior to use. The targets of interest typically appear to be constricted by Newtonian mechanics and a simple scalar Kalman filter can be used to filter the phase measurements. Prior to hardware implementation, some amount of information regarding the sensor noise parameters must be determined. This is done through repeated executions of a non-iterative MATLAB simulation of the FSKCW ranging process. The simulation aims to simulate the processing that must be performed on the IF signals to determine the range.

The simulation begins by generating two sine waves representing the IF output corresponding to a point target. The frequency is determined by the target velocity, and the phase difference is determined by the target range. These signals are digitized and a Hamming window is applied to suppress side lobes. The resulting signals are shown in Fig. 79.8. Additive white Gaussian noise (AWGN) is then introduced to both phases such that the signal to noise ratio (SNR) is 3 dB. The result is shown in Fig. 79.9.

The methodology described in Fig. 79.7 is applied and the resulting data is analyzed over a number of non-iterative simulation runs to determine the variance of the measured phase noise. This value is used to tune the gains of an online scalar Kalman filter running in-line with the FSKCW simulation. Figure 79.10 shows comparison of the filtered (green) and unfiltered (blue) phase data for 60 simulation runs. The filter converges after 8 phase measurements; if the IF signal is sampled at

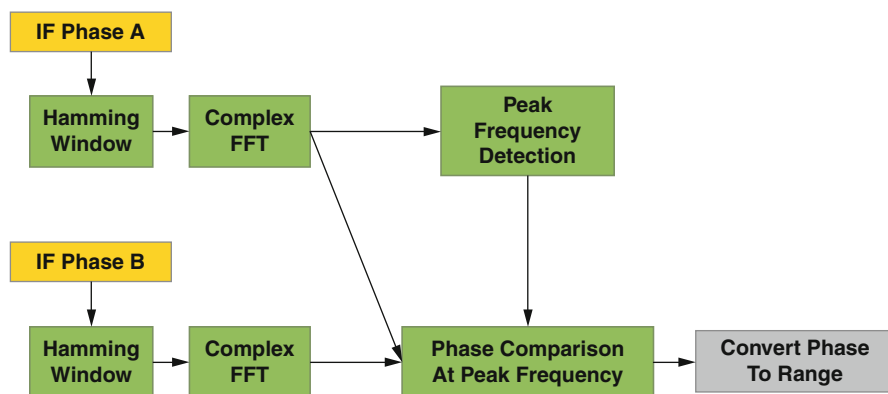


Fig. 79.7 FSKCW ranging methodology

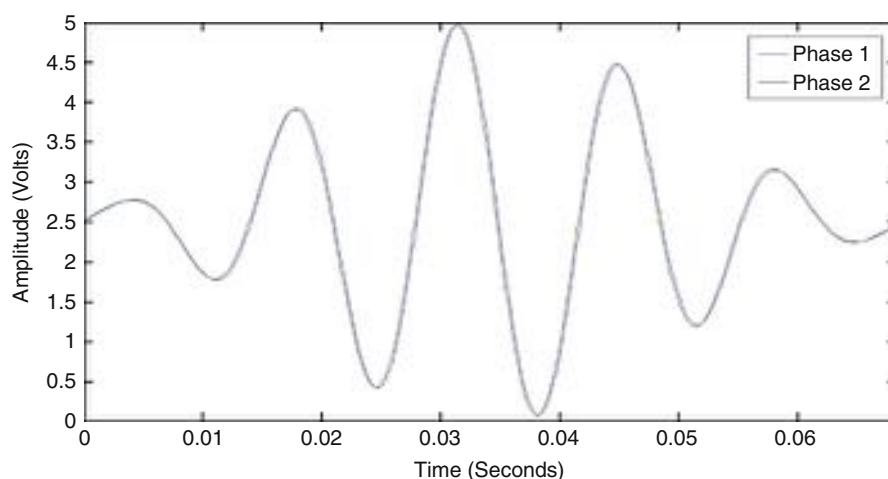


Fig. 79.8 Simulated IF signal without noise

15 kHz, this would take approximately 0.55 s. After convergence, the typical phase error is -0.00536 radians. This is equivalent to a range error of 0.26 m.

Experimental validation of the ranging method involves the use of a moving target as relative motion of the target is a requisite of FSKCW ranging. In this case an aluminum dihedral reflector with an RCS of 2 m^2 is mounted on an O-scale model train; see Fig. 79.11. The train traverses along an oval track, while the RADAR is arranged such that the main lobe illuminates a linear section of the track. The test setup is illustrated in Fig. 79.12.

The test scenario is complicated by the presence of multipath reflections from objects within the testing room and the detection of the target by both the main and side lobes when the target is outside the area of interest. These conditions

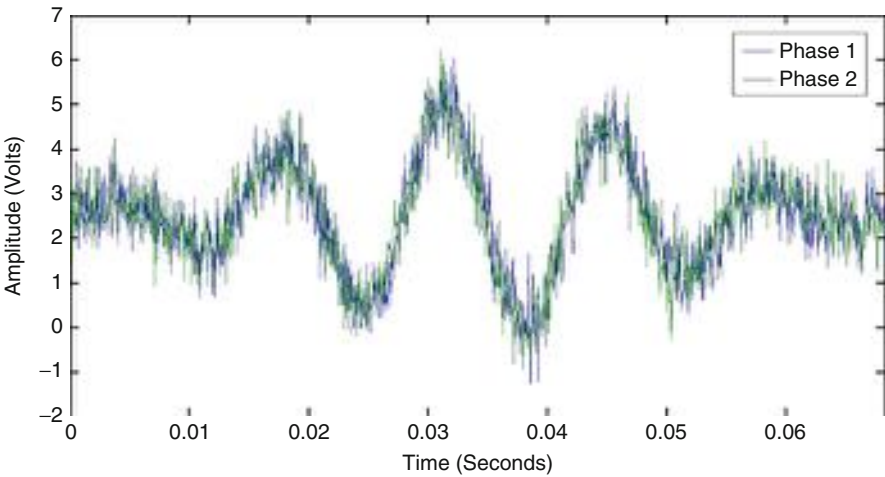


Fig. 79.9 Simulated IF signal with noise

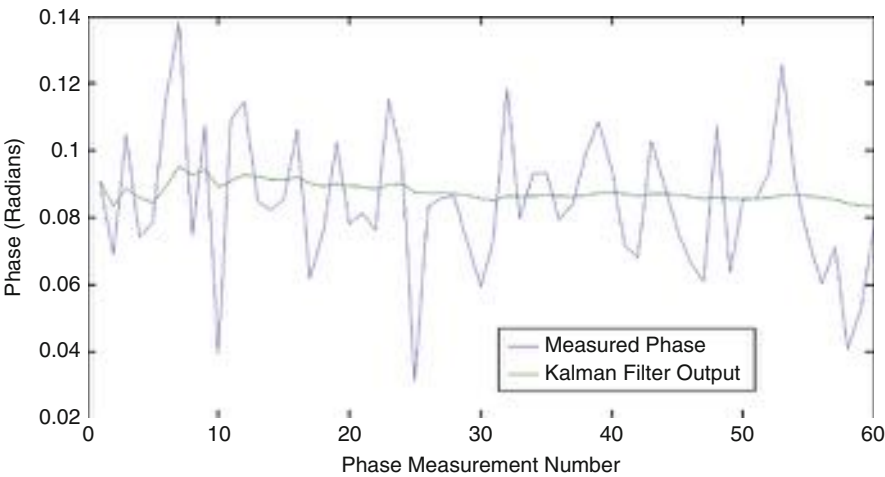


Fig. 79.10 Simulated phase data

necessitates the use of target discrimination to preselect the phase samples that are admitted into the Kalman filter. This target discrimination process consists of two primary metrics: target direction and velocity.

Throughout the testing, the train target traverses the track in a clockwise fashion. From the RADAR’s perspective, the target is always heading away from the antenna. This receding motion gives rise to a negative phase measurement (while the converse will give rise to a positive phase shift). This filter effectively eliminates the half of the track through which the train is heading towards the RADAR.

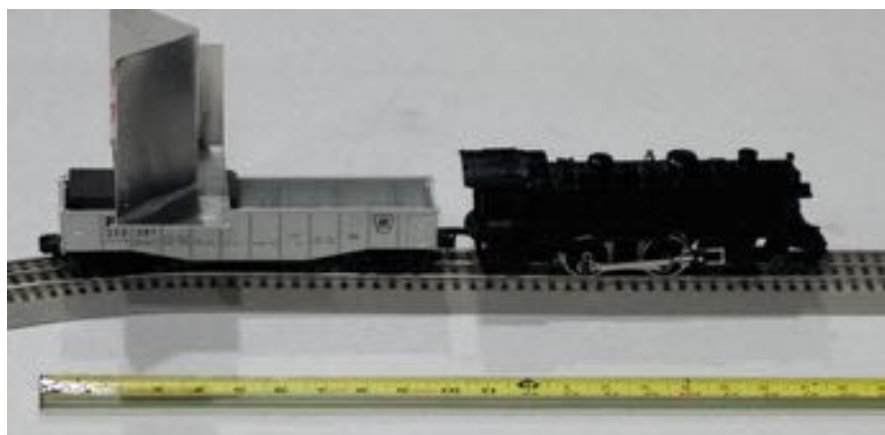


Fig. 79.11 O-scale train with Doppler target for FSKCW experimental validation

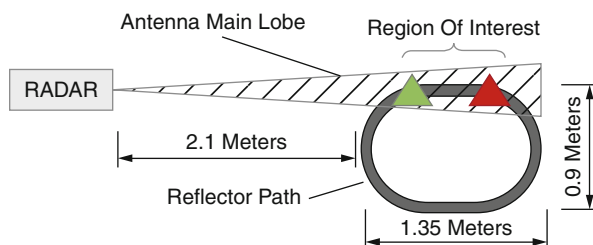


Fig. 79.12 FSKCW ranging experimental validation: test setup

Velocity filtering further reduces the measurement selection to periods in which the target's Doppler shift falls within a particular FFT bin. This condition is met when the target is traversing the region of interest and the antenna Poynting vector is colinear with the target motion vector.

The test begins by configuring the train's drive system to move the train and target assembly at 0.5 m/s, the velocity is then verified by a stopwatch. The RADAR system is then aligned to view the region of interest. The filtered phase data is then collected and stored for further analysis.

Figure 79.13 shows 36 range measurements gathered using this measurement setup. The region of interest bounds (represented as phase) are displayed as green and blue horizontal lines. The red trace is the measured phase data after the real-time target discrimination and range filtering process. The sinusoidal behavior of the measurement is explained by the target discrimination process. When the target enters the region of interest, the measured phase is relatively low. The target then moves away from the RADAR, thus causing the range to increase monotonically. This process ends when the target leaves the region of interest. The target goes around the track and the phase measurements are rejected due to the target

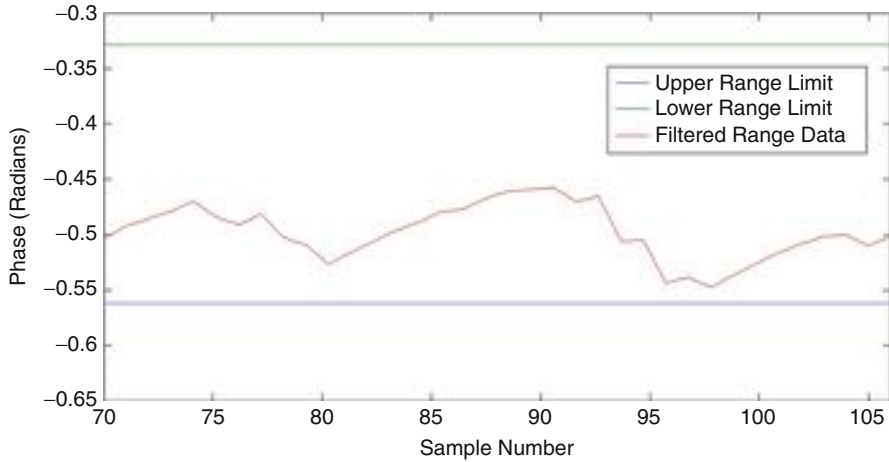


Fig. 79.13 FSKCW ranging experimental validation: results

discrimination process. When the target completes its circuit and reenters the region of interest, the measured phase value is low again and the filtered value is pulled down to reconcile the filter output with the new measurements before following the target's increasing range again.

It should be noted that as the region of interest is small, the prediction stage of the Kalman filter is omitted in these tests. This essentially assumes a stationary target. When transitioning to targets of a more dynamic nature, the Doppler shift in addition to the measurement timestamp can be used to provide a more accurate estimate of the target's current range. The phase estimate would then be of a form described by

$$\hat{\phi} = \frac{4\pi \left(R_{(k-1)} + c \left(\frac{F_{IF}}{2F_T} \right) (T_k - T_{(k-1)}) \right)}{c} \quad (79.12)$$

where $\hat{\phi}$ is the predicted phase shift and $R_{(k-1)}$ represents the previous range value taken at time $T_{(k-1)}$ ($R_{(k-1)}$ is obtained from the output of previous time steps).

79.4.1 Azimuth-Enabled RADAR Evaluation

Previous sections within this chapter have detailed the hardware and software methods utilized for performing FSKCW ranging against moving targets. However, relative bearing information is unavailable without one of the methods discussed in Sect. 79.3.2. This section first describes the hardware developed to test the generation 2a RADAR sensor. The software methods for determining target azimuth are then discussed. Finally, the accuracy of the RADAR is quantified.

79.4.2 RADAR Targets

The evaluation performed in Sect. 79.4 demonstrates the FSKCW concept; however, more accurate evaluation benefits from the use of RADAR targets with a more concentrated distribution of reflecting surfaces (by comparison, the reflecting surface of the train target is distributed over the length of the target which is approximately 0.53 m long). To address these issues and to evaluate the RADAR using multiple targets, two quad-dihedral reflectors are used; see Fig. 79.14.

The use of well-documented geometries (see Figs. 79.15 and 79.16) in the construction of these reflectors allows for the application of standardized formulas to determine the reflector RCS; see (79.13) and (79.14) (Gallman 2005):

$$\sigma_{\text{panel}} = \frac{4\pi a^2 b^2}{\lambda^2} \quad (79.13)$$

$$\sigma_{\text{dihedral}} = \frac{8\pi a^2 b^2}{\lambda^2} \quad (79.14)$$

where a is the width of the panels, b is the height, λ is the wavelength of incoming radiation, and σ is the RCS.

Applying Eqs. 79.14 and 79.13 to the reflectors shown in Fig. 79.14 produces the RCS values shown in Table 79.4.

Since FSKCW RADARs can only detect objects with some motion relative to the RADAR antenna, the reflectors in Fig. 79.14 are mounted on a Pololu “3pi” line following robot; see Fig. 79.17. This arrangement allows for reflector motion paths to be created using any high-contrast line, which the vehicle follows using optical sensors at speeds up to 0.9 m/s (Pololu.Inc 2013).

Within this research, two oval tracks are created using 19 mm wide electrical tape on a white poster board substrate. The dimensions of the tracks are shown in Fig. 79.18.

79.4.2.1 Azimuth Measurement Methodology and RADAR Interface

The generation 2a RADAR hardware uses mechanical scanning to direct the main lobe of the antenna. Throughout the research, the angular width of the path swept by the antenna is 90° with the period of the sweep being set to 2 s. RADAR range measurements are not synchronized with the motion of the antenna, and samples are continually gathered at the RADAR’s maximum rate. The maximum range measurement rate is limited by the FSKCW processing method which requires the use of floating point FFTs. Unfortunately, the XMOS hardware does not possess a floating point unit, and floating point operations are emulated using the XMOS’ fixed-point hardware. This causes a performance reduction, thus limiting the RADAR sample update rate to approximately 7 Hz (a measurement period of 0.14 s). As a result, during each measurement period, the antenna rotates 6.3° and each 90° antenna sweep has, on average, 14.3 range measurements. Before and



Fig. 79.14 Quad-dihedral RADAR reflectors (with 6'' (15 cm) size reference)

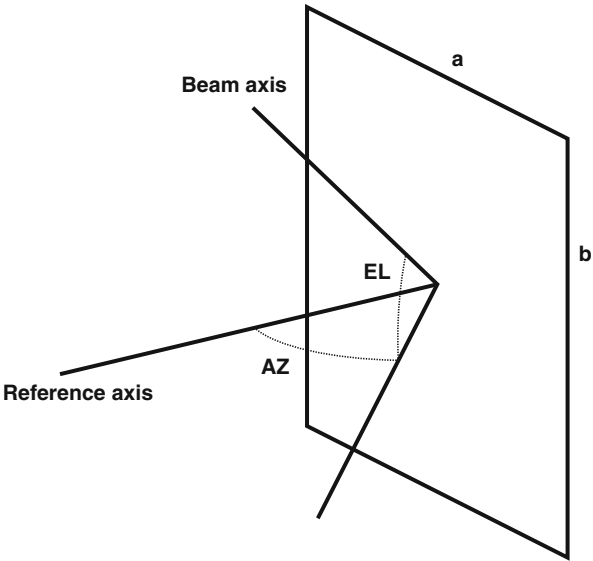


Fig. 79.15 Panel RCS geometry (Gallman 2005)

after each range measurement period, the antenna position is sampled and the range measurement is associated with the linear interpolation of these angles.

Each range measurement also has an associated “activity metric” which is simply the sum of the FFT bin magnitudes (excluding bin 0 which represents the DC component of the IF signal). When the activity metric values are arranged spatially and plotted, the local maxima of the plot represents individual targets if the local

Fig. 79.16 Dihedral RCS geometry (Gallman 2005)

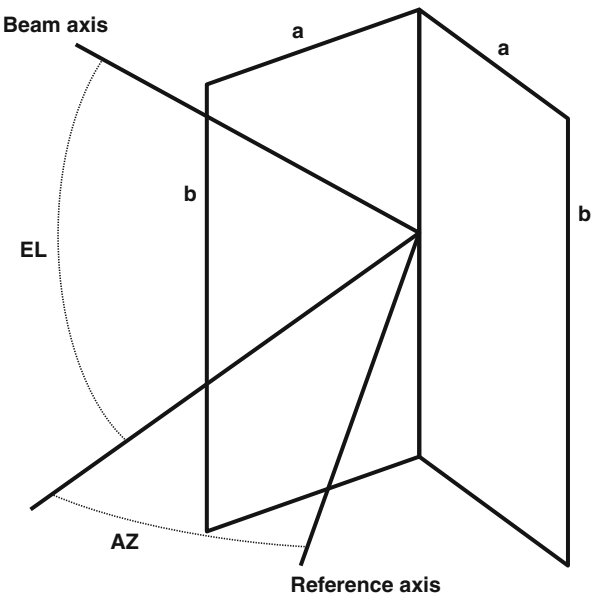


Table 79.4 RADAR reflector RCS

| | Large reflector | Small reflector |
|-------------------------------------|-----------------|-----------------|
| Panel peak RCS (m ²) | 3.73 | 2.23 |
| Dihedral peak RCS (m ²) | 1.87 | 1.02 |

maxima amplitude exceeds the noise threshold for the activity metric values. The noise threshold is determined experimentally by directing the RADAR towards an area with no Doppler activity and increasing the threshold value to eliminate false detections. For the purposes of this research, each local maxima is associated with a particular target, and the azimuth values are filtered using a scalar Kalman filter in the same manner that the FSKCW phase values are filtered.

Figure 79.19 is a screen capture of the laptop-based RADAR interface developed for this research and shows a plot (in the lower left) of the activity metrics for a scenario with two targets. Written in the *Processing* programming language, the RADAR interface software serves to provide a real-time visual representation of the RADAR data and record the RADAR data to a file for off-line analysis.

79.4.2.2 Combined Range and Azimuth Measurement

Combined range and azimuth measurements are made using a series of small- scale scenarios in which the target vehicles described in Sect. 79.4.2 are arranged and then set in motion about oval tracks also described in Sect. 79.4.2. Throughout all scenarios, the target vehicle velocity is approximately 0.5 m/s and is measured by using a quartz chronograph (Seiko SDNA57) to determine the time the target vehicles require to make a circuit of their respective tracks.

Fig. 79.17 Line following
RADAR target (with 6''
(15 cm) size reference)

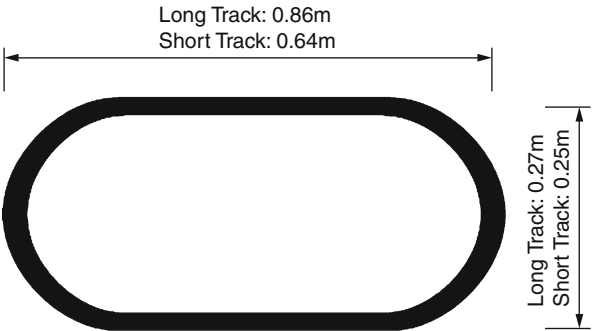


Fig. 79.18 RADAR
reflector path dimensions

Figure 79.20 shows the physical layout of the RADAR and the target vehicles during the first evaluation scenario. Scenario 1 is performed with all test elements located in a reinforced concrete room with no effort to reduce multipath reflections. For this scenario three tests are performed. First, target 1 is placed within the environment and data is gathered for 5 min, while target 1 is traversing the track in a clockwise direction. The second test is performed by introducing target 2 into the environment along with target 1. Throughout the second test, target 1 is stationary and target 2 is moving around its track in a counterclockwise direction. As with the first test, data is gathered for 5 min. The third and final test is performed with both target 1 and target 2 moving. As with the previous tests, data is gathered for 5 min.

Figure 79.21 displays the results from scenario 1, test 1, with the view centered around target 1. The path made by the target vehicle is indicated by the dashed

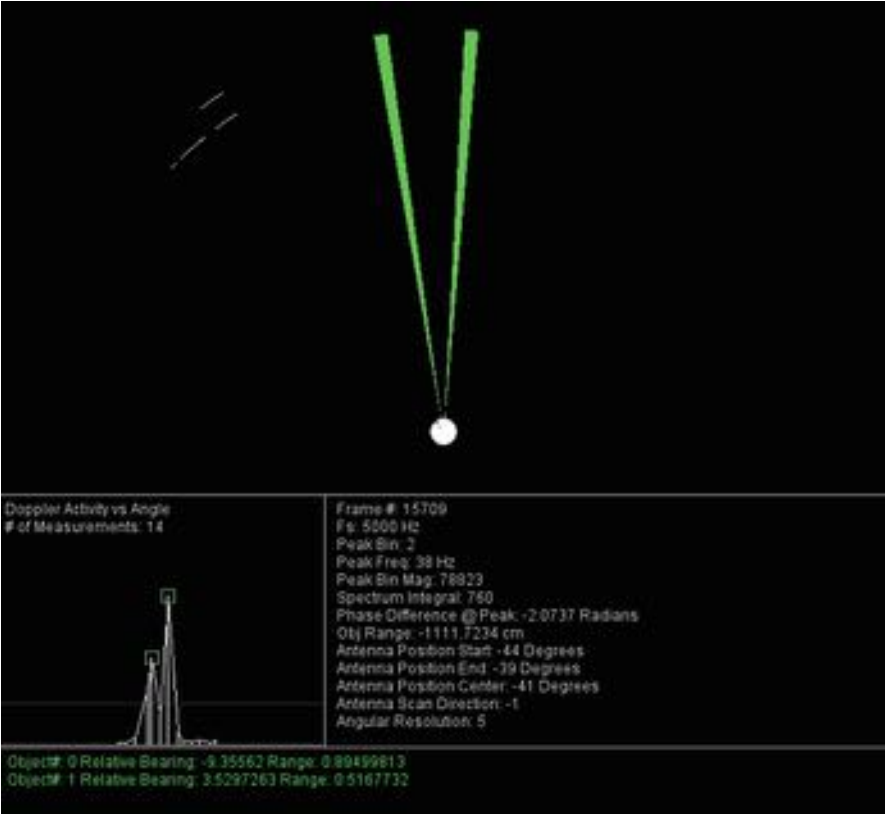


Fig. 79.19 RADAR interface screenshot

oval. However, the reflector is not a point target and physically sweeps the area bounded by the two solid ovals. The RADAR measurements are shown (after filter convergence) as a scatter plot with each measurement being the result of a single 90° antenna sweep. The mean and standard deviation of the measurements are also plotted with the X and Y axis mean values shown in green and the standard deviation shown in magenta and presented numerically in Table 79.5. Figure 79.22 presents the aforementioned information for scenario 1, test 2.

Figure 79.22 displays the results from scenario 1, test 2, with the view centered around target 2. A numerical analysis of the test is shown in Table 79.10. Figure 79.23 shows the results for scenario 1, test 3, which involves both target vehicles moving simultaneously. A numerical analysis is shown in Table 79.7.

Scenario 1 demonstrates the sensitivity of the FSKCW ranging technique to multipath signal propagation. Multipath signal propagation is defined within this research as any signal path not of the same length and direction as the LOS vector

Fig. 79.20 Generation 2a test scenario 1

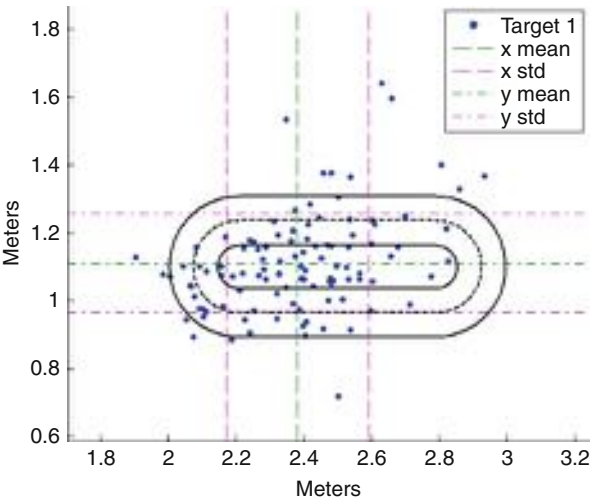
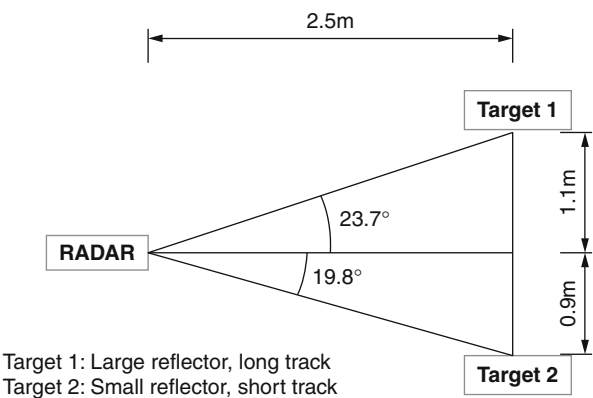


Fig. 79.21 Scenario 1, target 1 only (the RADAR is positioned at the origin, (0,0))

between the RADAR antenna and any given target. Within scenario 1, the primary cause of multipath errors is the addition of another RADAR reflector within the area swept by the antenna. As Tables 79.5 through 79.7 show, the average angular measurement error is 0.49° and the majority of the position measurement error is due to range measurement inaccuracies. For scenario 1, test 1, the average range measurement error is 0.1 m. Throughout the test, the transmit frequency modulation bandwidth is 4.45 MHz; thus a 0.1 m error is equal to a phase error of 0.5329° , which is, in turn, 0.3 % of the 180° unambiguous range limit. Table 79.8 shows the range error analysis for the tests performed in scenario 1. While range measurement errors are ideally 0 %, the range measurement errors throughout scenario 1 are comparable with other sensor technologies. For example, the Hokuyo URG-04LX-UG01 scanning laser rangefinder has a rated accuracy of ± 3 % the measurement range (Hokuyo.Inc 2009).

Table 79.5 Scenario 1, test 1: numerical analysis

| Parameter | Physical layout | RADAR measurement | Error |
|------------------------------|-----------------|-------------------|--------|
| X axis mean (m) | 2.50 | 2.38 | 0.12 |
| X axis std. dev. | – | 0.21 | – |
| Y axis mean (m) | 1.10 | 1.11 | 0.01 |
| Y axis std. dev. | – | 0.15 | – |
| Mean relative bearing | 23.70° | 25.03° | –1.33° |
| Mean line of sight range (m) | 2.73 | 2.63 | 0.1 |

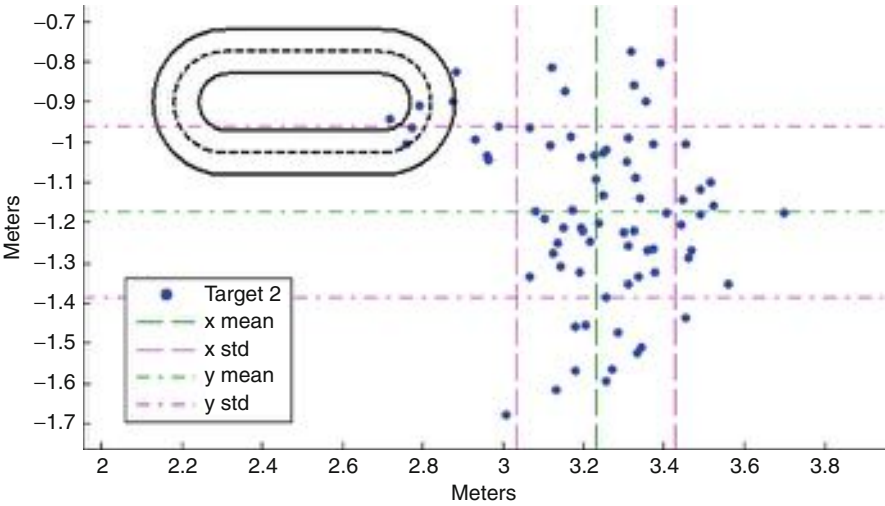


Fig. 79.22 Scenario 1, target 2 moving, target 1 stationary

Figure 79.24 shows the physical layout of the RADAR and the target vehicles during the second evaluation scenario. All measurements in scenario 2 are made within a steel hanger and no multipath mitigation steps are taken. For this scenario, two tests are performed. For the first test, target 2 is placed within the environment and RADAR measurements are made for 5 min (0.5 Hz update rate). During the second test, target 1 is introduced and RADAR measurements are made in an attempt to gather position information regarding both targets. A graphical representation of scenario 2, test 1, RADAR measurements is shown in Fig. 79.25, while a numerical analysis of the test results is shown in Table 79.9. Figure 79.22 and Table 79.10 provide the same information for scenario 2, test 2. As with scenario 1, the RADAR is located at the origin throughout scenario 2 (coordinate 0,0).

Scenario 2 demonstrates another vulnerability of FSKCW RADARs: target masking. During test 2, target 1 is masked by target 2 in that target 2 interrupts the line of sight path between the RADAR and target 1. Due to the longer measurement range, target 1 returns less power to the RADAR than target 2 despite target 1

Fig. 79.23 Scenario 1, both targets in motion

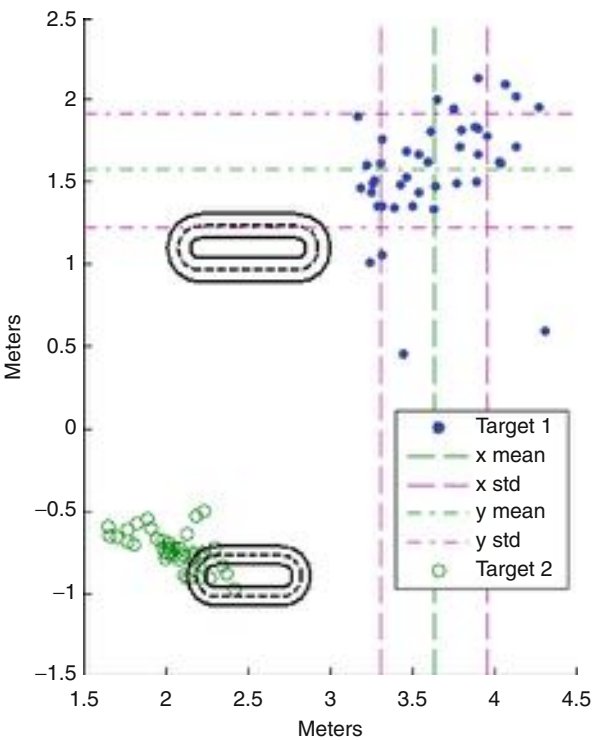


Table 79.6 Scenario 1, test 2: numerical analysis

| Parameter | Physical layout | RADAR measurement | Error |
|------------------------------|-----------------|-------------------|-------|
| X axis mean (m) | 2.50 | 3.23 | 0.73 |
| X axis std. dev. | – | 0.20 | – |
| Y axis mean (m) | –0.9 | –1.17 | 0.27 |
| Y axis std. dev. | – | 0.21 | – |
| Mean relative bearing | –19.80° | –19.91° | 0.11° |
| Mean line of sight range (m) | 2.25 | 3.44 | –1.21 |

having a larger RCS. Nevertheless, the range filter converges for target 1 rather than target 2. This is due to the more rapid phase reversals observed for target 2 that are not as prevalent in target 1; explained differently, the sign of the phase difference between the two IF channels used for FSKCW ranging depends on whether or not the target is heading towards or away from the RADAR. When utilizing the oval tracks described in Fig. 79.18, this phase reversal occurs twice for every lap of the target vehicle. Since both target vehicles are traveling at the same speed, and target 2 utilizes a shorter track, the RADAR measurements of target 2 contain more phase reversals which serve to corrupt the range measurement and prevent convergence of the range filter (Fig. 79.26). An analysis of the range errors for scenario 2 is shown in Table 79.11.

Table 79.7 Scenario 1, test 3: numerical analysis

| Parameter | Physical layout | RADAR measurement | Error |
|------------------------------|-----------------|-------------------|--------|
| Target 1: | | | |
| X axis mean (m) | 2.50 | 3.63 | −1.13 |
| X axis std. dev. | – | 0.32 | – |
| Y axis mean (m) | 1.10 | 1.56 | −0.46 |
| Y axis std. dev. | – | 0.35 | – |
| Mean relative bearing | 23.7° | 23.35° | 0.35° |
| Mean line of sight range (m) | 2.73 | 3.971 | −1.24 |
| Target 2: | | | |
| X axis mean (m) | 2.50 | 2.18 | 0.32 |
| X axis std. dev. | – | 0.18 | – |
| Y axis mean (m) | −0.9 | −0.73 | −0.17 |
| Y axis std. dev. | – | 0.11 | – |
| Mean relative bearing | −19.8° | −19.62° | −0.18° |
| Mean line of sight range (m) | 2.25 | 2.18 | 0.07 |

Table 79.8 Scenario 1 range error analysis

| Data set | Range error (m) | Phase error (°) | % of R_{unamb} |
|------------------|-----------------|-----------------|------------------|
| Test 1 | 0.10 | 0.5329 | 0.3 |
| Test 2 | 1.21 | 6.4659 | 3.6 |
| Test 3, target 1 | 1.24 | 6.6262 | 3.7 |
| Test 3, target 2 | 0.07 | 0.3741 | 0.2 |

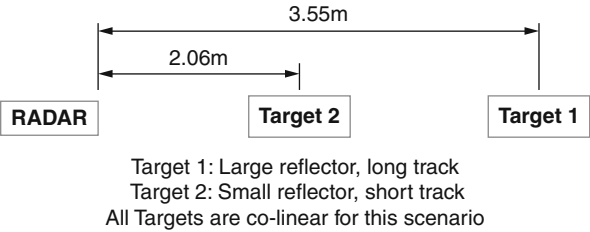


Fig. 79.24 Generation 2a test scenario 2

79.4.3 Remarks

Within this section, the FSKCW ranging process presented in this chapter is first simulated in Sect. 79.4 wherein simulations indicate that with a 3 dB SNR, the range errors should be, on average, 0.26 m. Throughout the hardware evaluation process, the average measured range error is 0.62 m or 138 % of the range error encountered during the simulation. While there are a number of factors that cause this discrepancy, there are three dominant causes: multipath error, frequency setting error, and the direction reversals of the RADAR targets. As multipath errors are

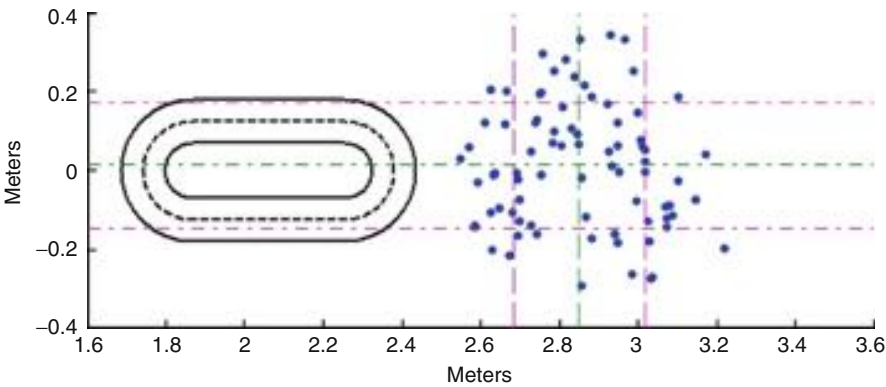


Fig. 79.25 Scenario 2, target 2 only

Table 79.9 Scenario 2, test 1: numerical analysis

| Parameter | Physical layout | RADAR measurement | Error |
|------------------------------|-----------------|-------------------|-------|
| X axis mean (m) | 2.06 | 2.851 | 0.79 |
| X axis std. dev. | – | 0.17 | – |
| Y axis mean (m) | 0.00 | 0.01 | 0.01 |
| Y axis std. dev. | – | 0.16 | – |
| Mean relative bearing | 0.00° | 0.28° | 0.28° |
| Mean line of sight range (m) | 2.06 | 2.86 | 0.78 |

Table 79.10 Scenario 2, test 2: numerical analysis

| Parameter | Physical layout | RADAR measurement | Error |
|------------------------------|-----------------|-------------------|--------|
| Target 1: | | | |
| X axis mean (m) | 3.55 | 3.67 | –0.12 |
| X axis std. dev. | – | 0.45 | – |
| Y axis mean (m) | 0 | 0.02 | –0.02 |
| Y axis std. dev. | – | 0.20 | – |
| Mean relative bearing | 0.00° | 0.33° | –0.33° |
| Mean line of sight range (m) | 3.55 | 3.67 | –0.12 |
| Target 2: | | | |
| X axis mean (m) | 2.06 | 2.87 | –0.81 |
| X axis std. dev. | – | 1.22 | – |
| Y axis mean (m) | 0 | –0.20 | 0.20 |
| Y axis std. dev. | – | 0.12 | – |
| Mean relative bearing | 0.00° | –4.50° | 4.50° |
| Mean line of sight range (m) | 2.06 | 2.89 | –0.83 |

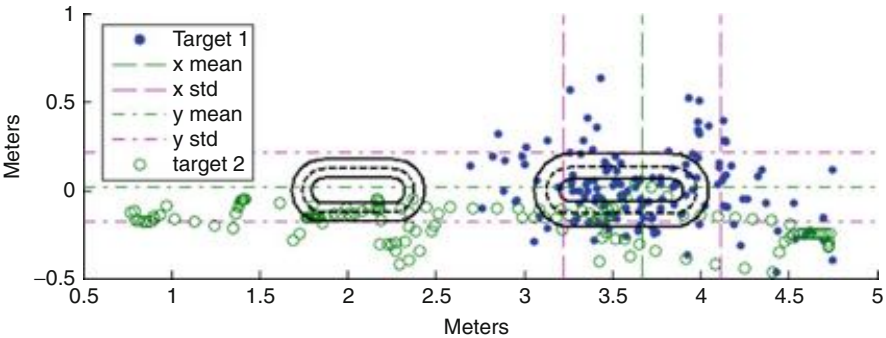


Fig. 79.26 Scenario 2, both targets

Table 79.11 Scenario 2 range error analysis

| Data set | Range error (m) | Phase error (°) | % of R_{unamb} |
|------------------|-----------------|-----------------|------------------|
| Test 1 | 0.78 | 8.4298 | 4.68 |
| Test 2, target 1 | 0.12 | 1.2969 | 0.72 |
| Test 2, target 2 | 0.83 | 8.9702 | 4.98 |

adequately explained above, this section will provide more in-depth information regarding frequency setting and direction reversals.

Frequency setting error is, in itself, comprised of two parts: the unmodulated carrier error and the frequency modulation error. The first item, unmodulated carrier error, is defined as an error in the assumed transmit frequency of the RADAR without the square-wave modulation used for determining the range to the target. For the RADAR front end used in the generation 2 and 2a prototypes, the unmodulated carrier frequency is 10.587 GHz. However, due to the construction technology, this frequency is temperature dependent and can vary by up to 6.5 MHz throughout the operating temperature range. Assuming a linear relationship between frequency and temperature, this equates to approximately 230 kHz/°C and is a relatively minor error source accounting for no more than 0.06 % of the unmodulated carrier error (Solutions 2013).

The frequency modulation error is markedly more important and is defined as the error between the expected and actual frequency shifts caused by the modulation process. The maximum frequency modulation range of the RADAR front end is theoretically as 5 MHz (at room temperature) and is obtained by applying a square wave between 0 and 5 V to the tuning pin of the RADAR front end (Solutions 2013). However, during the initial evaluation, the actual frequency range was determined to be 4.45 MHz. This value was obtained by assembling a test scenario with a moving RADAR reflector which passes through a marker located at a specific distance from the RADAR. An oscilloscope then used to measure the relative phase between the two IF signals as the RADAR target passes the marker. Since the speed of light, measured phase, and range to the target are known quantities, the modulation

bandwidth is determined by rearranging (79.9) to the form shown in (79.15) and solving for the bandwidth, BW :

$$BW = \frac{c(\phi_A - \phi_B)}{4\pi \text{Range}} \quad (79.15)$$

The dependence of the accuracy of the range measurements on the accurate knowledge of the modulation bandwidth is demonstrated by noting that for an object with a measured phase difference of 20° , the calculated range is 1.87 m for a modulation bandwidth of 4.45 MHz. If however, the modulation bandwidth is 5 MHz, the calculated range is 1.67 m. This is an error of 0.2 m or 10 % of the original measurement and is therefore a more significant source of range measurement errors than the unmodulated carrier error mentioned above.

In addition to errors caused by the uncertainty in the microwave section, there exist errors caused by the nature of the motion of RADAR targets used throughout this research. In order to simplify the experimental validation of the RADAR prototypes, it is necessary to have RADAR targets whose average location is a known constant. By definition, this conflicts with the need for the mobile targets required by the FSKCW ranging method. Therefore, throughout the evaluation process, the RADAR reflectors travel about oval tracks. Circular tracks would allow more convenient error measurement metrics to be used; however, their use would exacerbate the phase reversal problem briefly mentioned during the analysis of scenario 2.

Since the sign of the phase difference between the two IF signals used for FSKCW ranging is dependent on the direction of the target motion, a reversal of the apparent target motion direction results in a phase reversal. Figure 79.27 shows an oscilloscope capture of both RADAR IF signals while a target within the RADAR's field of view undergoes three direction reversals. The phase reversal is indicated by the regions with low signal amplitudes.

Figure 79.28 shows an enlarged view of a phase reversal. On the left side of the image, the green trace is leading the yellow trace, while on the right side, the yellow trace is leading the green trace. The figure shows 1 second of IF data; however, the RADAR system is sampling at 512 samples at a time at 5 kHz giving rise to a data packet length of 0.1024 s. Thus, over Fig. 79.28 the RADAR would make roughly 10 phase measurements (ignoring the 7 Hz processing speed limit). The phase reversal of the signals takes approximately 300 ms to complete and therefore for the data set shown in Fig. 79.28, 30 % of the phase measurements are erroneous. The use of oval tracks (as opposed to circular) reduces the percentage of erroneous measurements by increasing the percentage of time the RADAR reflector spends traveling either towards or away from the RADAR. Nevertheless, the routine phase reversals are the driving factor behind the gathering of data for 5 min for each test within the scenarios. This allows adequate time for the range filters to converge and filter the noise caused by phase reversals. Furthermore, the data is only analyzed after the convergence of the filters, which on average, require 10 samples to converge on a useful range value.

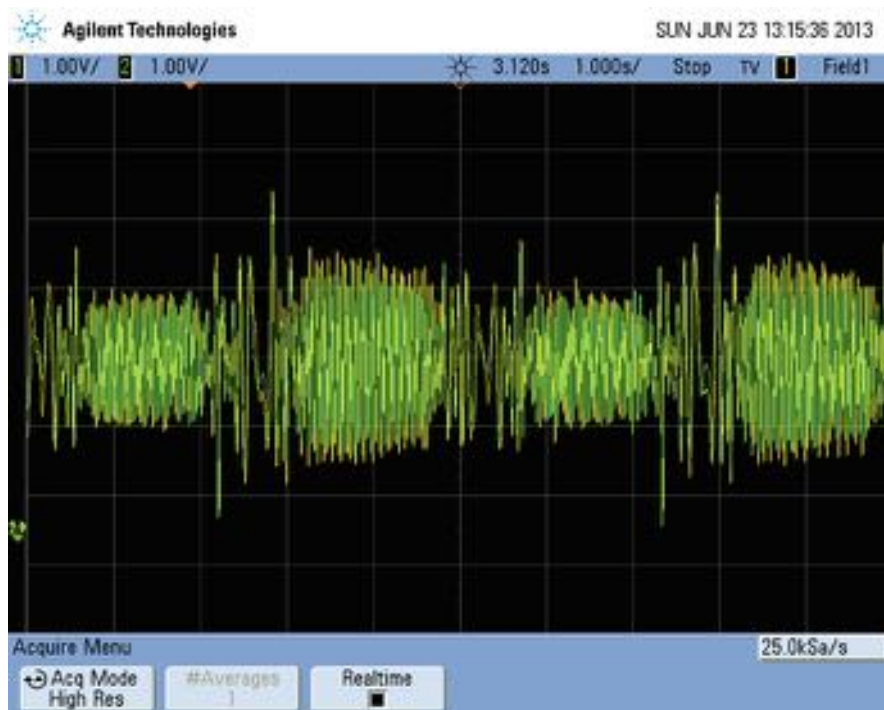


Fig. 79.27 Oscilloscope capture of both IF signals for three target direction reversals

Besides range, the other measurement required to localize targets is the target relative bearing. Throughout scenarios 1 and 2, the average relative bearing error is 1.01° . However, it should be noted that for all but two of the tests, the angular errors are below 0.36° . This is achieved by not synchronizing RADAR sampling with the motion of the antenna. The lack of synchronization serves to dither the angle measurements and reduce the measured angular error over time.

Also important is the horizontal antenna beamwidth (see Fig. 79.29) which shows the generation 2/2a antenna radiation patterns. The vertical pattern is shown on the left, while the horizontal pattern is shown on the right. The -3 dB beamwidths are 36° and 18° , respectively. Throughout each measurement, the antenna rotates 6.3° . Since this is less than the horizontal -3 dB beamwidth, the entire 90° sector swept by the antenna is covered by the main lobe with no gaps. If the antenna main lobe beamwidth were to be narrower than the angle traversed during the measurement period, there would be gaps in the main lobe coverage throughout any given antenna sweep and targets that would otherwise be detectable would remain undetected.

This chapter evaluates the hardware developed in Sect. 79.2 and provides a number of metrics regarding the RADAR's accuracy. While the hardware developed is not suited for flight testing due to the low transmit power, limited range, and slow scan speed, it does provide a valuable proof of concept. In the next

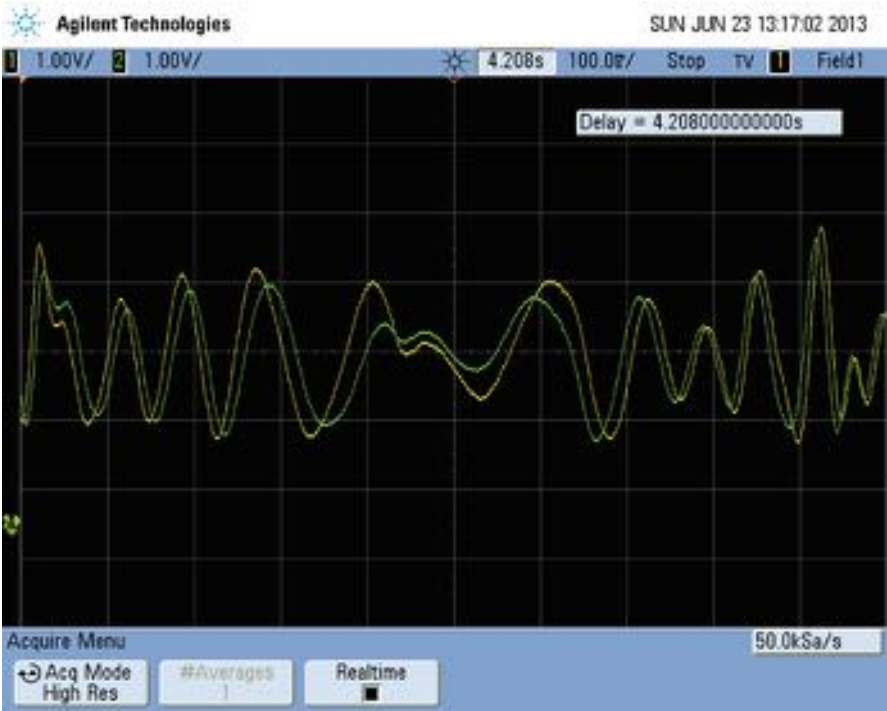


Fig. 79.28 Enlarged view of IF signals during a target direction reversal

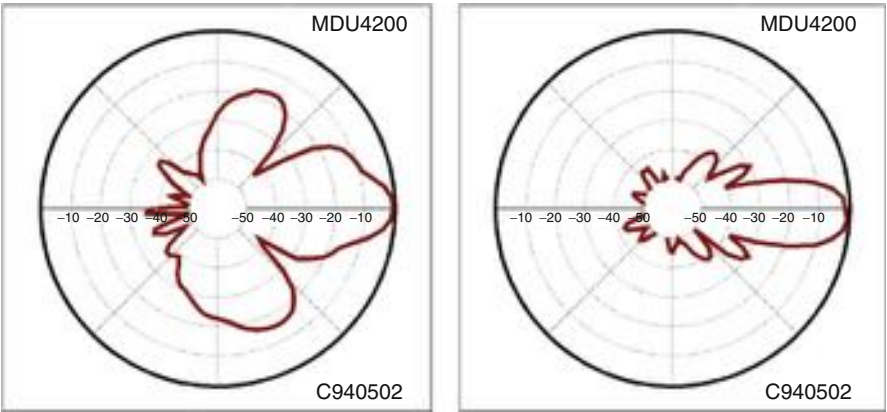


Fig. 79.29 Generation 2/2a antenna radiation pattern (Solutions 2013)

chapter, a collision avoidance algorithm that is capable of utilizing the RADAR's identification functionality is presented and evaluated.

79.5 Collision Detection and Evasion

There exist a number of different methods for avoiding collisions between aircraft. This chapter first describes the method this research uses for detecting collisions. Collision avoidance maneuvers are then divided into two categories, and a method for determining the lower energy maneuver is presented. Finally, this chapter describes the method this research uses to calculate collision avoidance maneuvers, and an empirical evaluation of the method is performed.

Stated differently, the collision mitigation process consists of a number of operations: target detection, target identification, collision detection, and host motion planning. The first two operations are described in Sect. 79.4. This chapter is concerned with the remaining two: collision detection and host motion planning.

79.5.1 Collision Detection

Collisions between two objects can be described in terms of relative bearing and range. A perfect collision between two point objects involves maintaining a constant relative bearing while the range between the two objects decreases. Aircraft have a nonzero volume and cannot be considered to be point objects. Furthermore, there exists a region around most aircraft that should remain free from other aircraft due to both safety and aerodynamic concerns. Throughout this dissertation, the shape of this exclusion region is assumed to be circular as viewed along the gravity vector. The dimensions of these regions are defined in terms of their radii, r_n . Figure 79.30 illustrates an arrangement involving two aircraft.

A minimum distance miss is achieved when the exclusion regions intersect at a single point and is defined as $r_1 + r_2$. The boundary dividing a collision condition and safe trajectories is defined in (79.16), where Φ is the line of sight (LOS) angle and R is the LOS range. Figure 79.31 displays the collision boundary for $r_1 = 5$ m and $r_2 = 10$ m.

$$\frac{d\Phi}{dR} = \frac{r_1 + r_2}{R^2 \sqrt{1 - \frac{(r_1 + r_2)^2}{R^2}}} \quad (79.16)$$

This plot is divided into several regions. The first region is the area of the plot in which the LOS range is less than or equal to $r_1 + r_2$. For values within this region, a collision has already occurred and (79.16) is invalid. Likewise, for $R = r_1 + r_2$ the required change in angular rate approaches infinity. The second region is represented by the region above the collision boundary for all $R > r_1 + r_2$. Within this region,

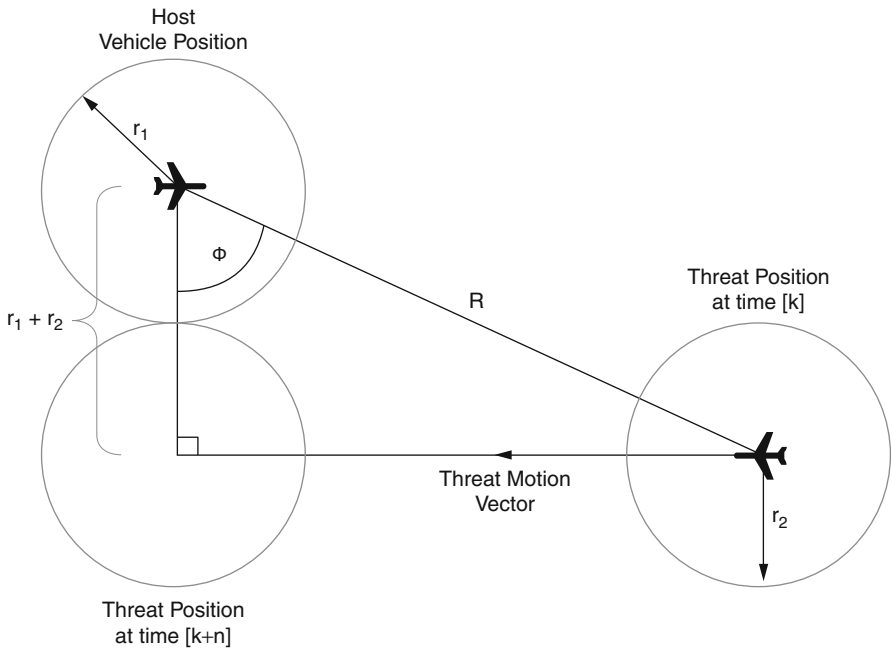


Fig. 79.30 Collision geometry

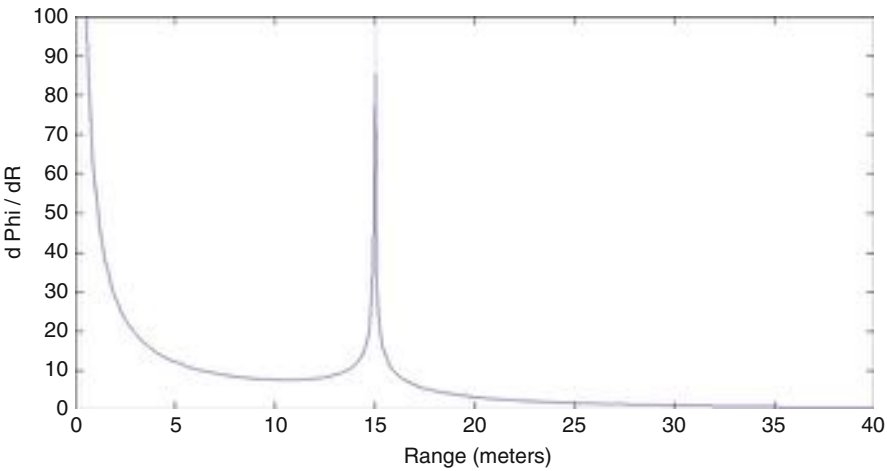


Fig. 79.31 $\frac{d\Phi}{dR}$ plot for $r_1 = 5$ m and $r_2 = 10$ m

$\frac{d\Phi}{dR}$ is high enough to avoid a collision. The final region is the region below the collision boundary for all $R > r_1 + r_2$. Within this region, $\frac{d\Phi}{dR}$ is low enough to cause a breach of the clearance regions.

For every time step, R , r_1 , and r_2 are entered into Eq. 79.16 to determine a value of $d\Phi$ against which the RADAR data is compared. If the measured $d\Phi$ value is below the calculated threshold $d\Phi$, then a collision is detected.

79.5.2 Collision Avoidance Maneuver Classes

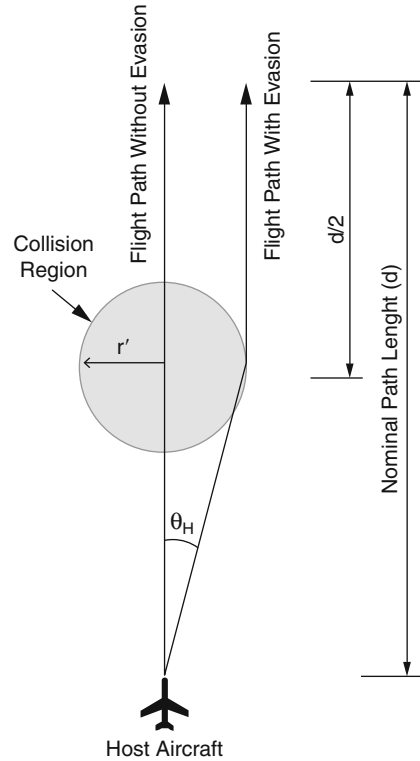
Once a collision has been detected using the methodology described in Sect. 79.5.1, some action by the host vehicle is required to avoid the collision. The proposed method divides collision avoidance actions into two categories: purely vertical maneuvers (altitude changes) and horizontal maneuvers with no vertical component (heading changes). Complex paths, incorporating both vertical and horizontal maneuvers, are not investigated in an effort to simplify the analysis. Furthermore, as the currently implemented transponder technology discussed in Sect. 79.1 prescribes vertical collision avoidance maneuvers, this section aims to examine the efficacy of collision avoidance maneuvers within the horizontal plane when the energy consumed in such a maneuver is less than or equal to the energy consumed in a vertical maneuver.

There are two key assumptions made for this analysis. First, it is assumed that the host vehicle maintains a constant forward velocity throughout the scenario, i.e., throughout the analysis, the rate at which the vehicles follow their trajectories is a constant. This is because smaller vehicles are able to rapidly vary their velocity, while, larger, faster moving, vehicles are unable to adjust their velocities on similar timescales. Furthermore, rapidly varying forward velocities while maintaining a constant altitude makes poor use of a vehicle's available energy by not trading kinetic energy (velocity) for potential energy (altitude).

The second assumption is that collision avoidance maneuvers in the horizontal plane require more physical separation than maneuvers in the vertical plane. This is representative not only of the physical dimensions of most aircraft which are typically wider and longer than they are tall but also of Federal Aviation Administration (FAA) regulations (Order JO 7110.65U) which requires more vertical separation than horizontal separation. In some cases under Instrument Flight Rules and RADAR guidance rules, 1,000 ft vertical separation or 3 miles lateral separation is required (FAA 2012a).

A collision avoidance scenario for a single host and a single opposing aircraft, for both horizontal and vertical avoidance maneuvers, is shown in Figs. 79.32 and 79.33, respectively. The host aircraft is shown on the bottom of the figures with the goal of traversing towards some goal at the top of the figures. Between the current host aircraft location and the goal is a region where, at some point in the future, a collision will occur between the host aircraft and some airborne threat. This "collision region" is modeled as an oblate spheroid with a minor semidiameter equal to r'/a and a major semidiameter equal to r' . a therefore determines the aspect ratio of the collision region and may be made to conform to arbitrary dimensions on a case-by-case basis.

Fig. 79.32 Horizontal evasion geometry (top down view)



The energy required for the vehicle to perform a horizontal maneuver is described by (79.17), while the energy required for a vertical maneuver is described by (79.18):

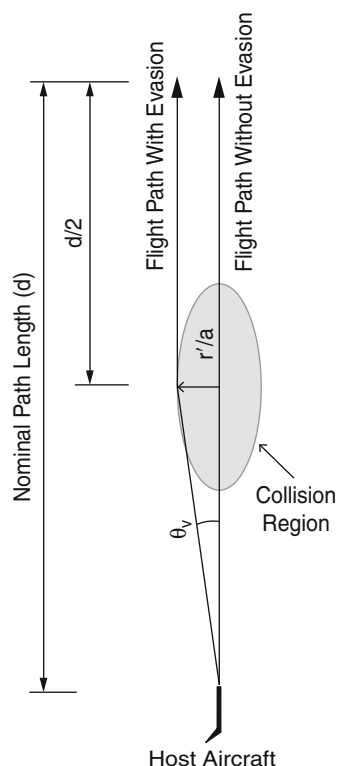
$$E_H = b \left(0.5d + \frac{0.5d}{\cos \left(\arctan \left(\frac{2r'}{d} \right) \right)} \right) \quad (79.17)$$

$$E_V = b \left(0.5d + \frac{0.5d}{\cos \left(\arctan \left(\frac{2r'}{da} \right) \right)} \right) + mg \left(\frac{r'}{a} \right) \quad (79.18)$$

where E_H and E_V are the energies required for horizontal and vertical evasion, respectively. b is the baseline energy consumed by the aircraft for every unit of distance traveled and is determined by dividing the engine output power while cruising by the aircraft cruising velocity. d is the nominal path distance as shown in the figures, a is the aspect ratio of the collision region, m is the mass of the vehicle, g is acceleration due to gravity, and r' is the lateral or vertical displacement required to avoid the collision region.

For any given aircraft, (79.17), and (79.18), can be used to determine when a vertical maneuver is superior from an energy standpoint to a horizontal maneuver. Furthermore, this comparison allows practical limits to be placed on horizontal

Fig. 79.33 Vertical evasion geometry (side view)



maneuvers so that the energy consumed in an in-plane horizontal maneuver will not exceed the energy of a vertical maneuver. Figure 79.34 shows the energy required for a horizontal maneuver with a heading change up to 30° compared with an equivalent vertical maneuver.

The parameters used to obtain Fig. 79.34 for a theoretical aircraft are shown in Table 79.12. For comparison, the baseline energy of a 1.5 kg radio-controlled aircraft (Eflite Ultra Stick 25e) is approximately 18 J/m (Hobby 2013).

In the case described by Fig. 79.34, horizontal collision avoidance maneuvers are more energy efficient than their vertical counterparts, provided that the heading is not modified by more than 19.7° .

Figure 79.35 shows the same analysis performed for a Cessna 172 fixed-wing airplane, given the parameters in Table 79.13. The parameters are selected under the assumption that the aircraft is equipped with a standard Lycoming IO-360-L2A engine operating at 75 % of its maximum power rating. This results in the aircraft cruising at 122 knots (62.76 m/s) at an altitude of 8,000 ft (Company 1977). As shown in Fig. 79.35, horizontal maneuvers are more energy efficient than vertical maneuvers for heading changes up to 28° .

Compared with fixed-wing aircraft, helicopters have higher baseline energy consumption rates. Due to the longer path length, horizontal collision avoidance

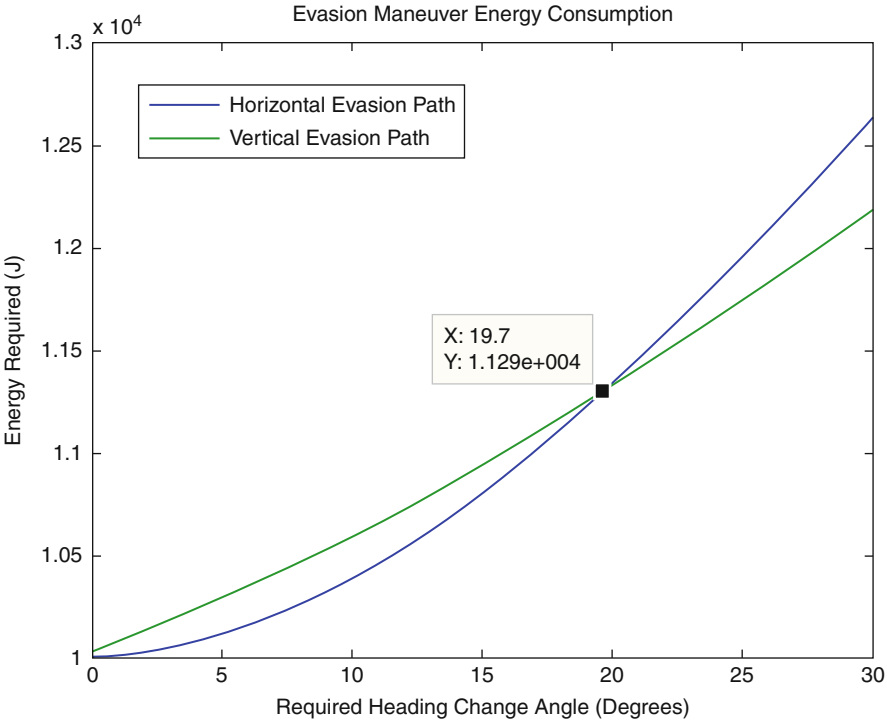


Fig. 79.34 Horizontal vs. vertical collision avoidance energies for a theoretical aircraft

Table 79.12 Parameters for a theoretical aircraft. See Fig. 79.34

| Parameter | Value |
|--------------------------------|----------------------|
| Vehicle mass (m) | 5 kg |
| Baseline energy (b) | 100 J/m |
| Aspect ratio (a) | 2 |
| Nominal path distance (d) | 100 m |
| Gravitational acceleration (g) | 9.8 m/s ² |

maneuvers require lower energies only over a narrower angle than fixed-wing aircraft. Figure 79.36 displays the results of an evasion maneuver energy analysis for a Bell 206 helicopter. The analysis specifications are shown in Table 79.14 (Inc 2000). In this case, horizontal collision avoidance maneuvers require lower energies for heading modification angles up to 12°.

79.5.3 Collision Avoidance Maneuver Planning

As mentioned earlier, the algorithm proposed in this section focuses on investigating the efficacy of collision avoidance maneuvers in the horizontal plane. Section 79.5.2

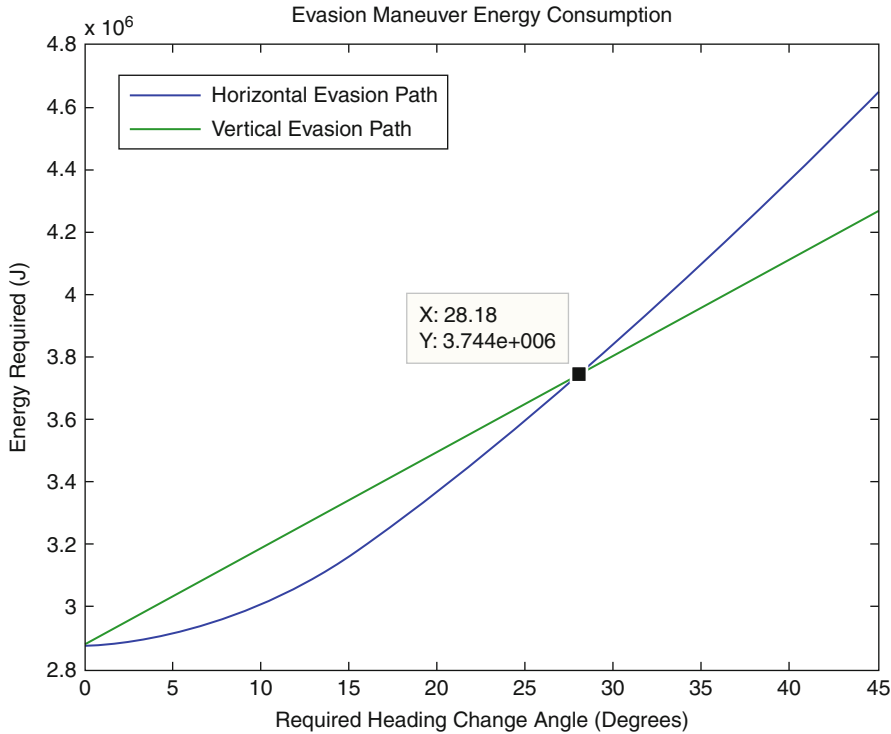


Fig. 79.35 Horizontal vs. vertical collision avoidance energies for a Cessna 172

Table 79.13 Parameters for a Cessna 172

| Parameter | Value |
|---------------------------|-----------|
| Vehicle mass (m) | 1,111 kg |
| Baseline energy (b) | 1,438 J/m |
| Aspect ratio (a) | 15.8 |
| Nominal path distance (d) | 2,000 m |

demonstrates that such maneuvers are only more energy efficient over an specific angle whose size is dependent on the host vehicle properties and the aspect ratio of the collision region, while this section describes the collision avoidance method. The description begins with a single opposing aircraft, and then the method is generalized for an arbitrary number of threats.

Section 79.5.1 describes a geometric method for detecting collisions, where a collision is detected when the line of sight (LOS) vector’s rate of change as a function of range is not greater than the boundary described by Eq. 79.16. This condition occurs when the angular separation between the LOS vector and the relative velocity vector between the host and any given threat is not sufficiently large. Such a condition is illustrated by Fig. 79.37, where Ψ is the angular distance between the LOS and relative velocity vectors.

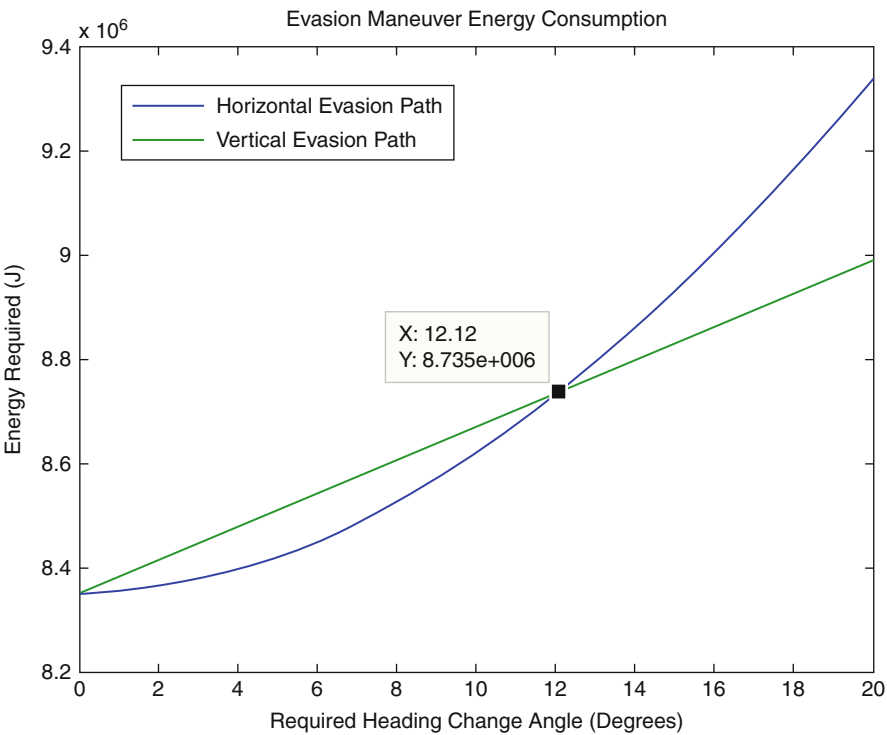


Fig. 79.36 Horizontal vs. vertical collision avoidance energies for a Bell 206

Table 79.14 Parameters for a Bell 206 helicopter

| Parameter | Value |
|---------------------------|-----------|
| Vehicle mass (m) | 1,400 kg |
| Baseline energy (b) | 4,175 J/m |
| Aspect ratio (a) | 15.8 |
| Nominal path distance (d) | 2,000 m |

The collision avoidance algorithm begins by creating a series of alternate LOS vectors for each threat. This is accomplished by rotating the LOS vector about the threat in uniform increments throughout an angular range centered about the current LOS vector known as the “search range.” The alternate LOS vectors are shown in Fig. 79.38. Where the alternate LOS vectors are displayed in red, the original LOS vector is shown in green, and the relative velocity vector is shown in blue.

The absolute magnitude of the angular distances between each alternate LOS vector and the relative velocity vector is then calculated and stored in an array, henceforth referred to as the “vector_distance” array. This has the effect of creating an array where each element represents one possible host vehicle heading change. As Fig. 79.38 shows, the alternate vectors are generated from the opposing aircraft’s reference frame. This is transformed to the host aircraft reference frame by shifting

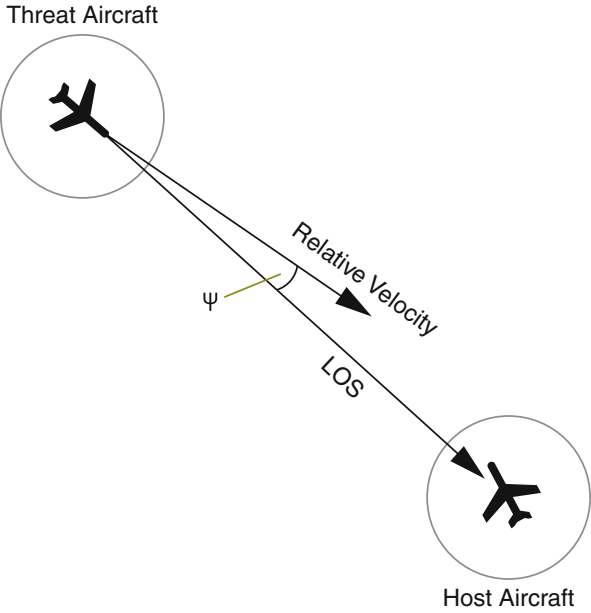


Fig. 79.37 Collision geometry

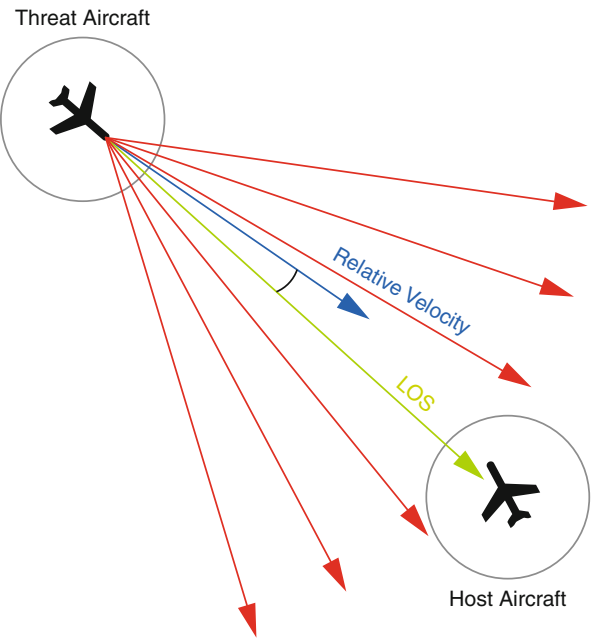
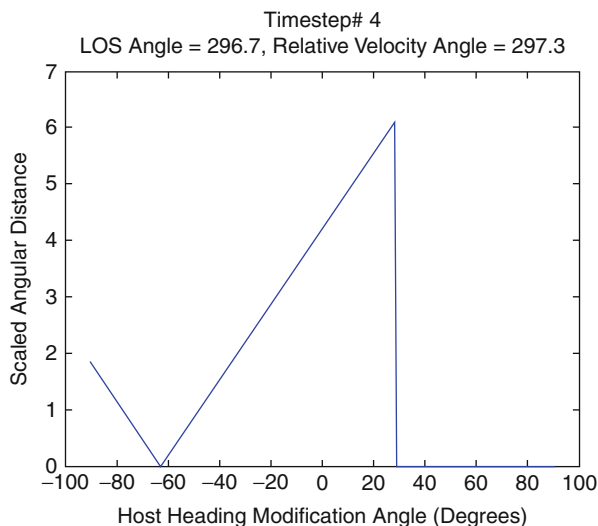


Fig. 79.38 Collision geometry with alternate LOS vectors

Fig. 79.39 Plot of “vector_distance” array after the shifting and weighting operations



the vector_distance array element values to the left for threats on the left side of the host vehicle (relative to the host longitudinal axis) and to the right for threats on the right side of the host vehicle. The magnitude of this shifting operation is such that the angular displacement of the threat LOS vector from the host vehicle boresight vector is equal to the shift distance multiplied by the angular resolution of each element. The vector_distance array element magnitudes are then weighted so that the magnitudes are inversely proportional to the range between the host and the opposing aircraft.

A plot of the vector_distance array for a single opposing aircraft is shown in Fig. 79.39. Figure 79.40 shows the vector_distance array evolution over time for a threat on a collision course. For both figures, the opposing aircraft's relative LOS angle is 296.7° , and the relative velocity vector angle is 297.3° .

Once a vector_distance array is created for a particular opposing aircraft, it is added to a global array (known as the global_heading_distance array) which incorporates the vector_distance array values for all detected opposing aircraft (regardless of whether or not the aircraft in question is on a collision course). The heading modification angle is determined by finding the index of global maximum of the global_heading_distance array multiplied by the angular resolution of the generated alternate LOS vectors.

Throughout this section, 180 alternate LOS vectors are utilized for each opposing aircraft. The alternate LOS vectors are uniformly distributed throughout a 180° range resulting in an angular resolution of 1° .

The collision avoidance algorithm proposed in this section is described by the following pseudocode.

It is important to note that this algorithm does not take into account any heuristics regarding which aircraft has the “right of way” (as defined in FAA 14 CFR

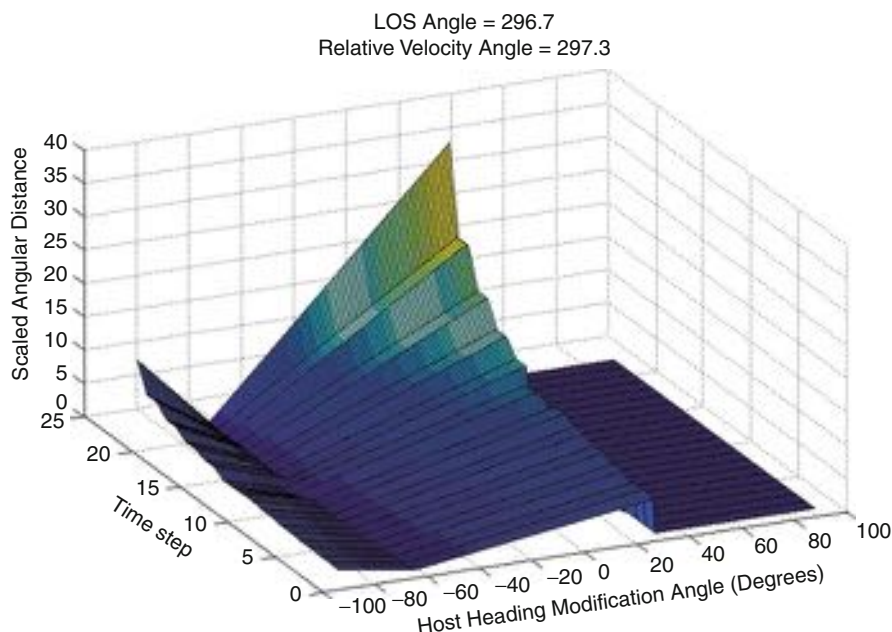


Fig. 79.40 Plot of “vector_distance” displayed over time

91.113 or 91.115). This is because the algorithm described herein is a last resort collision avoidance methodology, rather than a high level path planner. Ideally “right of way” considerations would be handled by the higher level guidance software in addition to human operators such that this algorithm will never be needed. However, when the need for a collision avoidance maneuver arises, the priority is placed on avoidance rather than “right of way.”

The performance of the algorithm is evaluated by translating the method into C and then executing the algorithm on the XC-1A development board used for RADAR control, as described in Sect. 79.2. Execution time is indicated by setting an I/O pin to *logic 0* before execution of the algorithm and then setting the same I/O pin to *logic 1* upon completion of the algorithm. Execution time is measured by observing the state of the I/O pin using an oscilloscope (Agilent DSO5012A). The execution time measurement is performed for up to 20 opposing aircraft with 180 alternate LOS vectors per aircraft. The results are shown in Fig. 79.41.

Require: Opposing Aircraft Velocity Vectors, LOS Vectors, Range
while $n < \text{number_of_opposing_vehicles}$ **do**
 $\text{global_heading_distance}[all] \leftarrow 0$
 Generate_Alternate_LOS_Vectors_For_Vehicle $n(n)$
 while $i < \text{number_of_alternate_LOS_vectors}$ **do**
 $\text{vector_distance}[all] \leftarrow 0$


```

vector_distance[i] ← abs( $\angle$ alternate_LOS[i] -  $\angle$ velocity_vector)
i = i + 1
end while
if  $\angle$ LOS[n]  $\geq 0^\circ$  and  $\angle$ LOS[n]  $\leq 180^\circ$  then
  shift_vector_distance_array_right( $\angle$ LOS[n])
else
  shift_vector_distance_array_left( $\angle$ LOS[n])
end if
vector_distance[all] ← vector_distance[all]  $\left( \frac{\text{scaling\_factor}}{\text{range}(n)} \right)$ 
global_heading_distance[] + vector_distance []
n = n + 1
end while
while j < size(global_heading_distance[]) do
  if global_heading_distance[j] = global_maxima then
    break
  end if
  j = j + 1
end while
required_heading_modification_angle = jxalternate_LOS_vector_resolution

```

79.5.4 Airspace Simulation Software

Due to the expense and difficulty of evaluating midair collision scenarios in hardware, algorithm evaluation takes place in simulation environment. To this end, a custom simulation software package was written in *Processing*, a branch of the Java programming language. This simulation software is comprised of a number

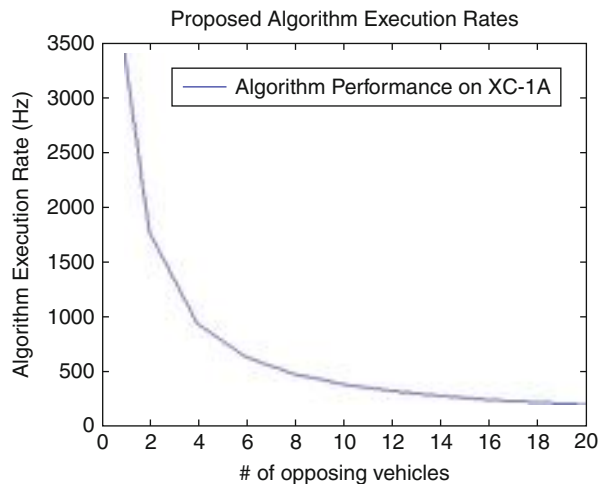


Fig. 79.41 Algorithm execution rate on the RADAR processor hardware

of modules each responsible for a certain portion of the simulation. The modules each act on a global state matrix that stores information on all aircraft within the simulation such as current coordinates, destination coordinates, velocity, exclusion region size, and aircraft maneuverability. The modules acting upon this data are as follows:

Initialization

The *initialization* module is executed upon the start of any given simulation sequence. While the module is routinely executed, the effect of the module is selectable to provide randomization of the positions, destinations, sizes, and velocities of all aircraft within the simulation.

Motion

The *motion* module's primary function is to update the current positions of all aircraft within the simulation based on the destination waypoint and the current velocity. As the global state matrix describes the vehicle position using latitude and longitude with precision of 0.111 m, relatively fine positioning control over the various simulated aircraft is possible.

Pilot Input

At any point in the simulation, a pilot may take control of the host aircraft using keyboard inputs. This has the effect of updating the "destination waypoint" for the host aircraft. Updates to this parameter are effected during the next simulation time step.

RADAR Track Generator

This module serves to simulate the data as gathered by the RADAR sensor by sampling the global state matrix and generating RADAR track data. This module is capable of simulating the real-world parameters including limited range, azimuth, and range errors.

Performance Monitoring

The primary metric used for evaluating the effectiveness of the collision avoidance algorithms is the minimum distance between the host aircraft and any other aircraft in the simulation. This module records this parameter for further investigation.

Graphical Representation

As the name implies, the *graphical representation* module is responsible for parsing the global state matrix and displaying a graphical representation of the airspace. Also represented are pertinent values regarding the host vehicle parameters and the performance of the collision detection and avoidance software; see Fig. 79.42. Due to the additional computational overhead involved in the rendering process, this module is only active during debugging when visual feedback is beneficial.

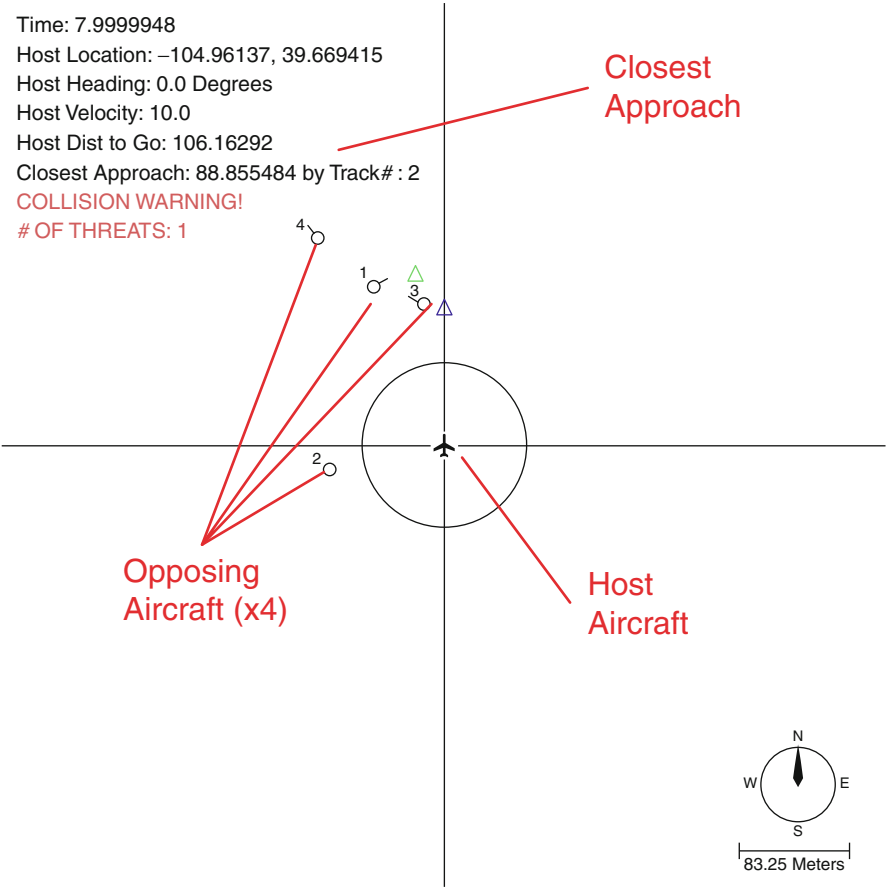


Fig. 79.42 Airspace simulation software screenshot

79.5.5 Collision Avoidance Algorithm Evaluation

The collision avoidance algorithm described in Sect. 79.5.3 is evaluated using the collision avoidance simulation software described in the previous section. The goal of this evaluation is to examine the efficacy of collision avoidance maneuvers in the horizontal plane. There are, however, a number of key constraints placed upon the simulation environment. They are as follows:

Opposing Vehicle Initial and Final Range

Throughout the algorithm, the *initialization* module is configured to generate opposing aircraft with random initial locations and destinations. The random generation is constrained such that, upon initialization, the latitude or longitude of the start and end points is no closer than 200 m and no further than 300 m from the host aircraft.

Thus, the initial LOS range between any aircraft and the host is between 282.8 and 424.3 m. This prevents the spawning of opposing vehicles colocated with the host; additionally it provides time for the host vehicle to effect a collision avoidance maneuver.

Opposing Vehicle Dispersion Time

Due to the constraints placed upon the locations of opposing aircraft, initial simulations indicated that opposing aircraft's starting locations may sometimes be generated closely together. Therefore, upon initializing any given scenario, collision avoidance maneuvers are inhibited for 2 s, thus allowing the opposing aircraft to disperse. Failure to do so artificially improves the performance metrics of the algorithm since there are large swaths of empty airspace.

Vehicle Collision Radii

Throughout the simulation, all aircraft have a collision avoidance radii of 10 m. This results in a collision being registered when the range between the host and any opposing aircraft within the simulation is less than 20 m; see Fig. 79.30. Ten meters is justified as a collision avoidance radius since it provides adequate clearance between many small UAVs and the maximum dimension of the Cessna 172 which has an 11 m wingspan (Company 1977). The Cessna 172, while not the most challenging threat to evade, is the most likely opposing aircraft in many municipalities since more Cessna 172s have been built than any other manned aircraft (more than 43,000 examples have been produced) (Niles 2007).

Single Collision Avoidance Maneuver

Throughout the evaluation process, a single collision maneuver is permitted. This is to avoid a meandering, in-plane path which would consume more energy than a vertical maneuver.

Throughout the evaluation process, the simulated area represents a horizontal slice of extremely high-density airspace. In this simulation, the slice is a square with an area of 0.36 km^2 , although airspace density values are normalized to represent aircraft/ km^2 . For every major parameter variation, a batch of simulations are run to determine the effect for various airspace densities up to $55.6 \text{ aircraft/km}^2$. For each incremental increase in airspace density, the simulation is executed 10,000 times (10,000 random scenarios). A scenario is defined as lasting from the end of the randomization process used during the beginning of the simulation to the time when all aircraft have reached their destination waypoints. Although the duration of any given scenario will vary due to the random nature of the waypoints and the velocities of the aircraft involved, the average duration of a scenario is 36.76 s. The results presented below represent 2.09 million scenarios (approximately 21,341 h of simulated flight time). For all efficacy plots, the vertical axis represents the probability of a collision, while the horizontal axis represents the airspace density.

Figure 79.43 displays the probability of a collision with and without the collision avoidance algorithm. Throughout the simulation, the velocities of all opposing

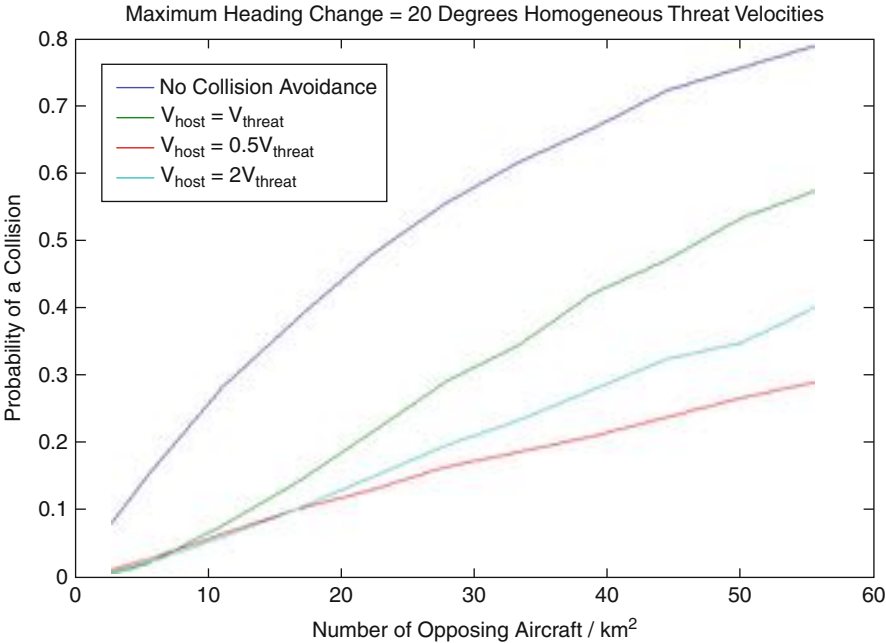


Fig. 79.43 Twenty degree heading modification angle

aircraft are 20 m/s. Furthermore, the maximum heading modification angle is limited to 20° to approximate the theoretical vehicle described in Table 79.12.

Figure 79.44 displays the effect of varying the maximum heading modification angle from the 20° used in Fig. 79.43. All vehicle velocities are 20 m/s.

Figure 79.45 shows the efficacy of the algorithm utilizing a range of homogeneous opposing aircraft velocities and a fixed host aircraft velocity. The maximum heading modification angle is 20°.

Figure 79.46 shows the efficacy of the algorithm with a maximum heading modification angle of 20° and randomized opposing aircraft velocities between 10 and 40 m/s.

Figures 79.47–79.50 show the data from Fig. 79.46 with and without collision avoidance enabled.

79.5.6 Remarks

Within this chapter, a geometric collision detection and evasion method is proposed and evaluated for a nontrivial number of random scenarios. Throughout the evaluation process, a number of key parameters are varied for each sweep of airspace densities: the maximum heading modification angle, velocities of both the host

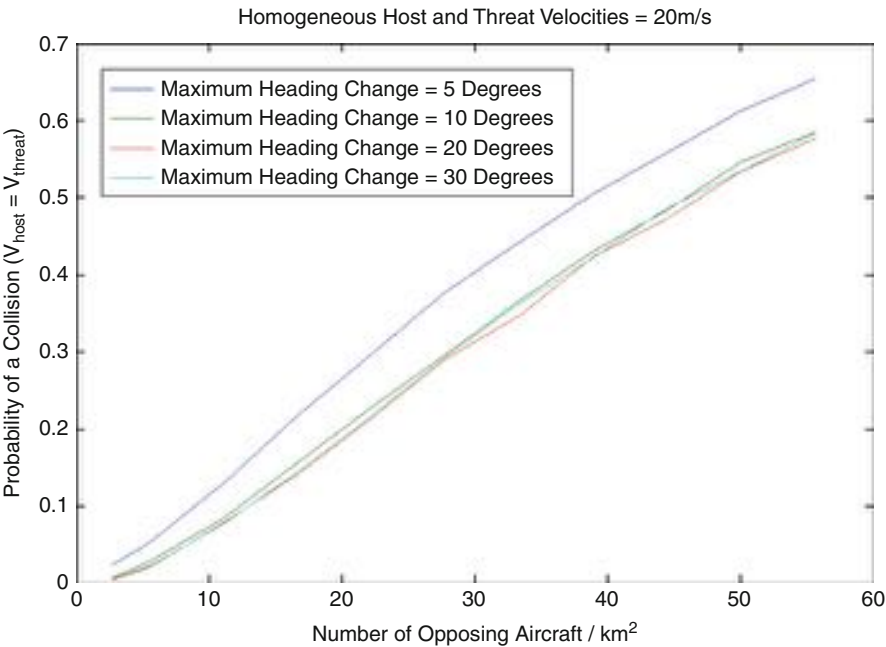


Fig. 79.44 Various heading modification angles

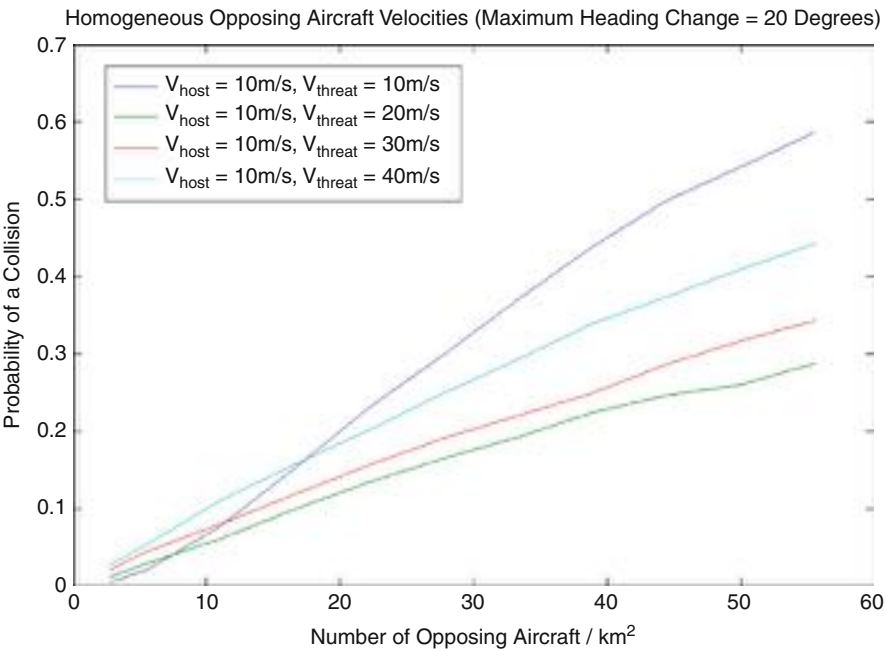


Fig. 79.45 Various homogeneous opposing aircraft velocities with constant host velocity

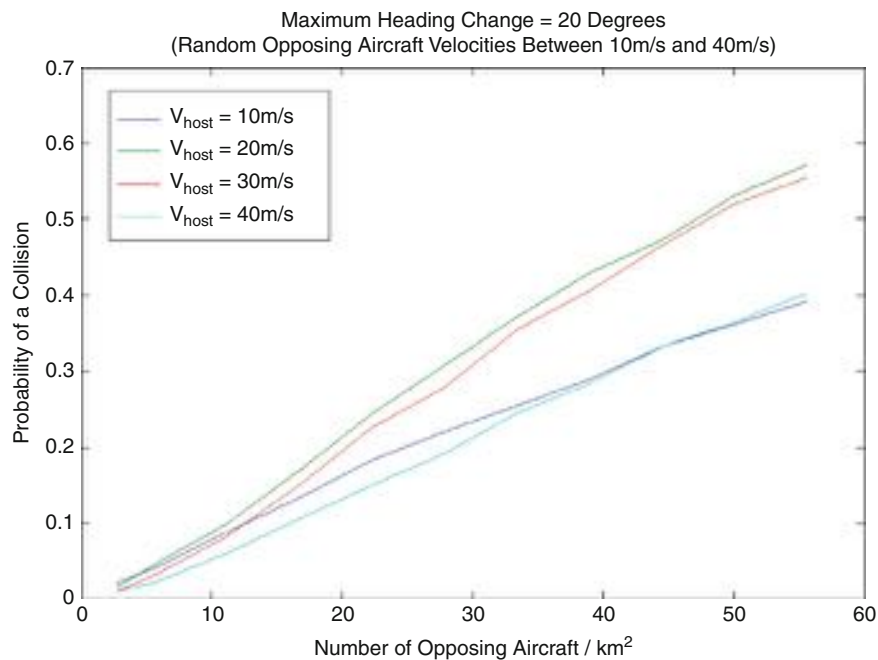


Fig. 79.46 Random opposing aircraft velocities with host velocity sweep

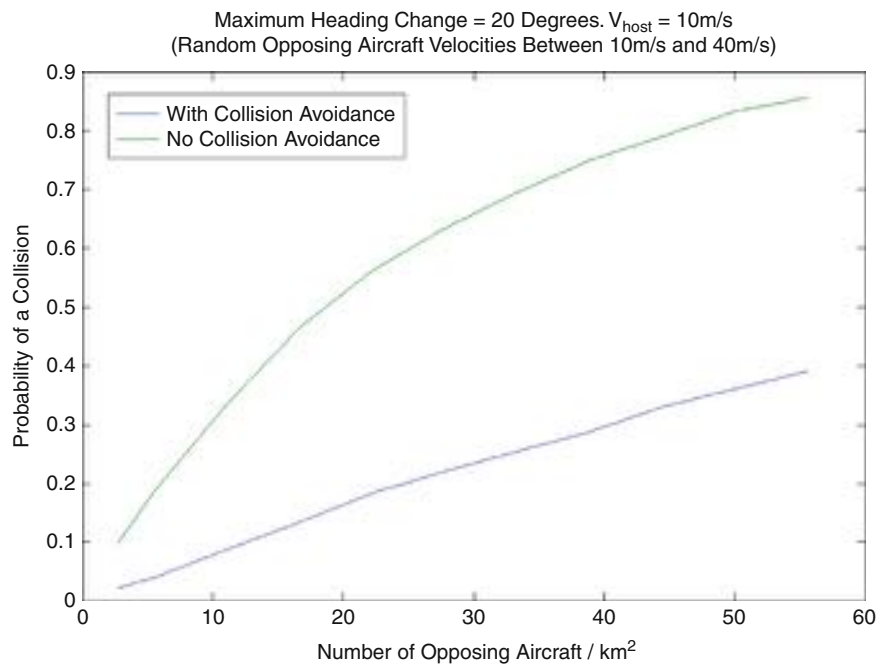


Fig. 79.47 Random opposing aircraft velocities with 10 m/s host

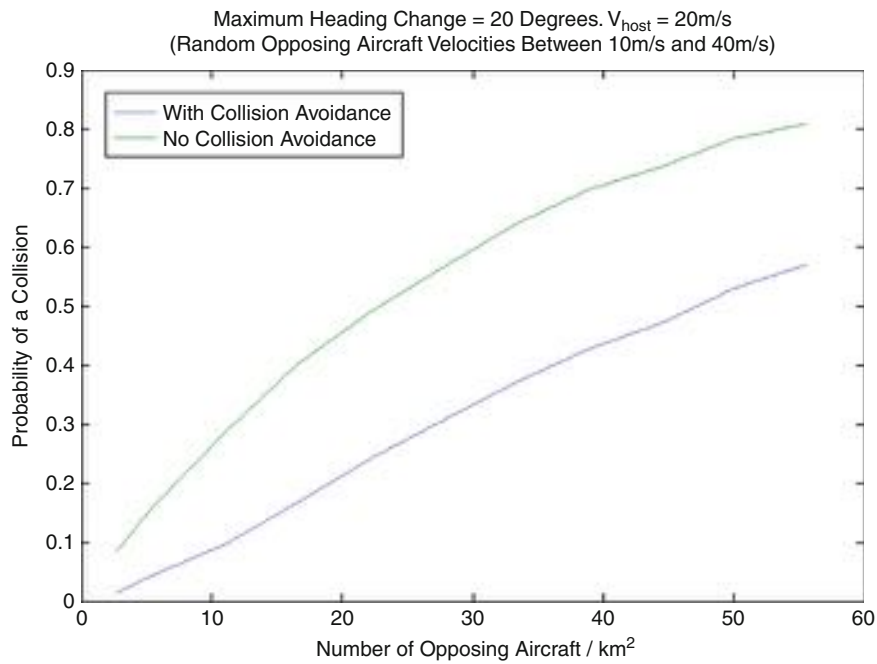


Fig. 79.48 Random opposing aircraft velocities with 20 m/s host

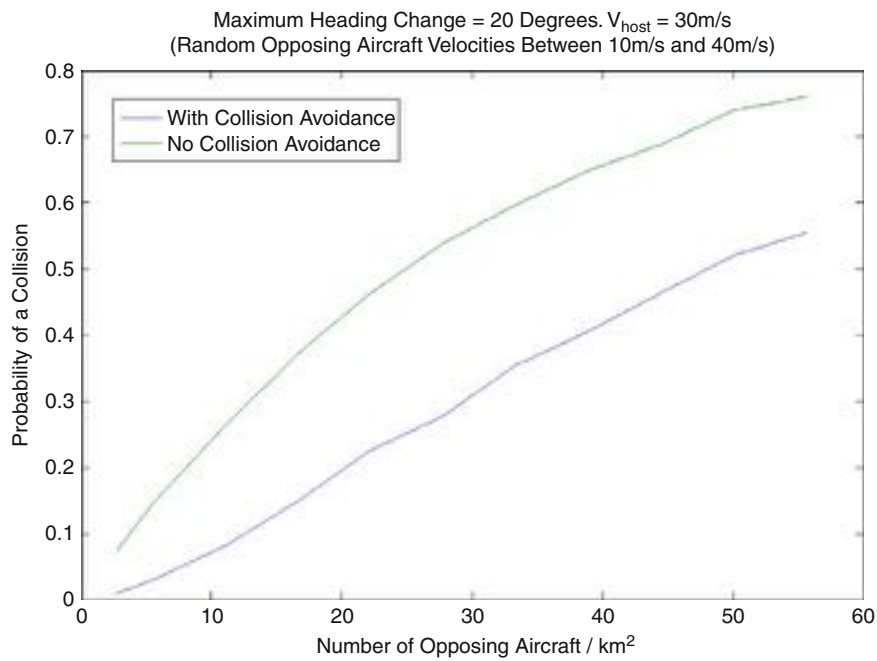


Fig. 79.49 Random opposing aircraft velocities with 30 m/s host

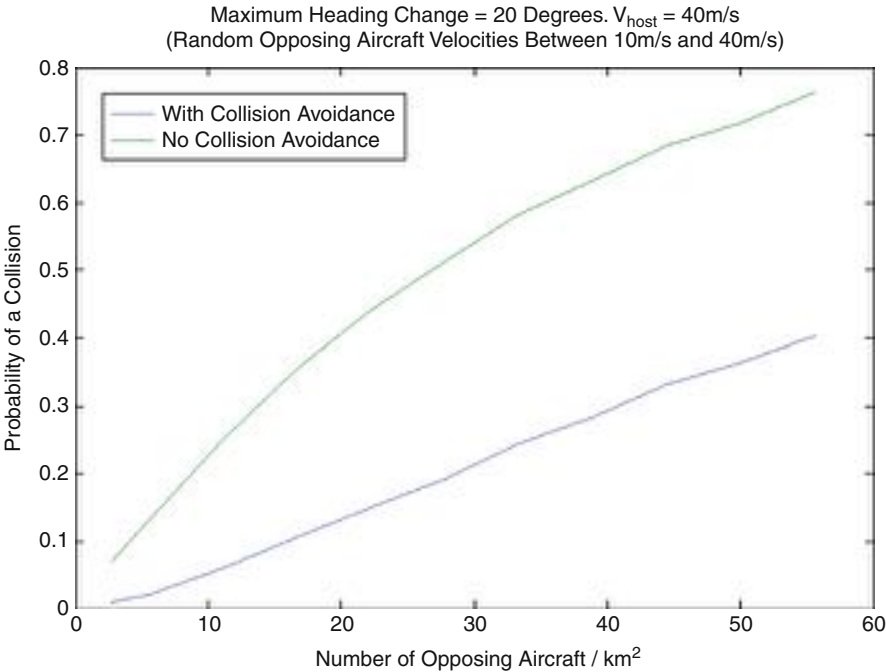


Fig. 79.50 Random opposing aircraft velocities with 40 m/s host

and the threat, and whether or not the scenario utilizes homogeneous or random opposing aircraft velocities.

Figure 79.44 shows the efficacy of the algorithm as a function of various maximum heading modification angles. The best performance within this data set occurs when the maximum heading modification angle is 20°. For values below 20°, the host aircraft is unable to avoid flying into the collision region described in Fig. 79.32. For values greater than 20°, the host aircraft is able to clear the original collision region, but often initiates a collision with another aircraft in the process.

The efficacy of the algorithm is also affected by the velocities of the vehicles involved and the opposing aircraft velocity distribution. With homogeneous opposing aircraft velocities, the simulations show that out of the examined configurations, the best efficacy is achieved when the host vehicle velocity is half the opposing vehicle velocity. With this velocity configuration, there is adequate time for the “cloud” of opposing aircraft to pass by the host vehicle. Increasing the host vehicle speed is detrimental to the performance in that it increases the closing speeds between the host and any opposing aircraft. This increased closing speed reduces the amount of time available for the host to clear the various collision regions.

The situation is reversed (for airspace densities up to 44 aircraft/km²) once random opposing aircraft velocities are introduced. Under these conditions, the best efficacy achieved during this simulation occurs when the host vehicle velocity

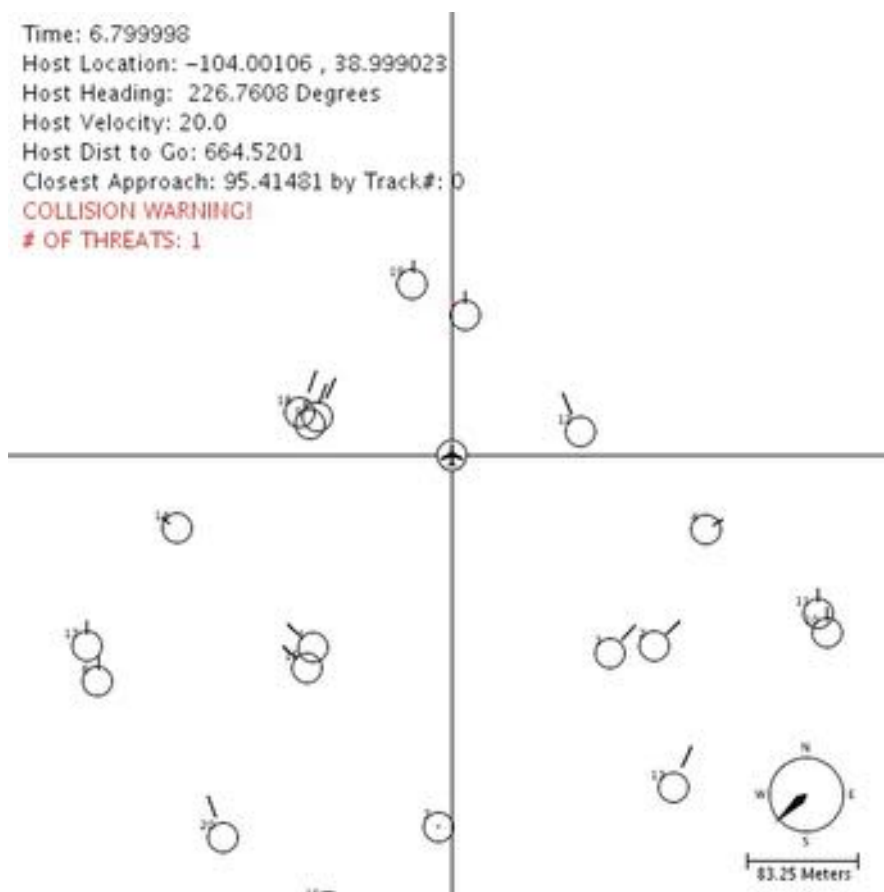


Fig. 79.51 Airspace simulation screenshot with 55.56 aircraft/km²

is equal to the upper bound of the opposing aircraft velocity distribution. This is explained by the “threat cloud” not dispersing due to the low-speed vehicles. Simultaneously, the host vehicle lacks the velocity to escape high-velocity opposing aircraft which penetrate the cloud of low-speed vehicles.

The efficacy of the algorithm is evaluated for airspace densities up to 55.56 aircraft/km² (see Fig. 79.51) at which point the probability of a collision exceeds 30 % for many of the scenarios in Sect. 79.5.5. It should be noted, however, that such airspace densities are highly improbable. For the sake of example, it can be assumed that the highest airspace density occurs within the airspace surrounding airports (the FAA classifies these regions density as “class B” airspace) (FAA 2001). The busiest airport in the world (according to the number of takeoffs and landings) is the Hartsfield-Jackson Atlanta International Airport (ATL) in Atlanta, Georgia, with 950,119 such events in 2010 (ACI 2011). While ATL’s class B

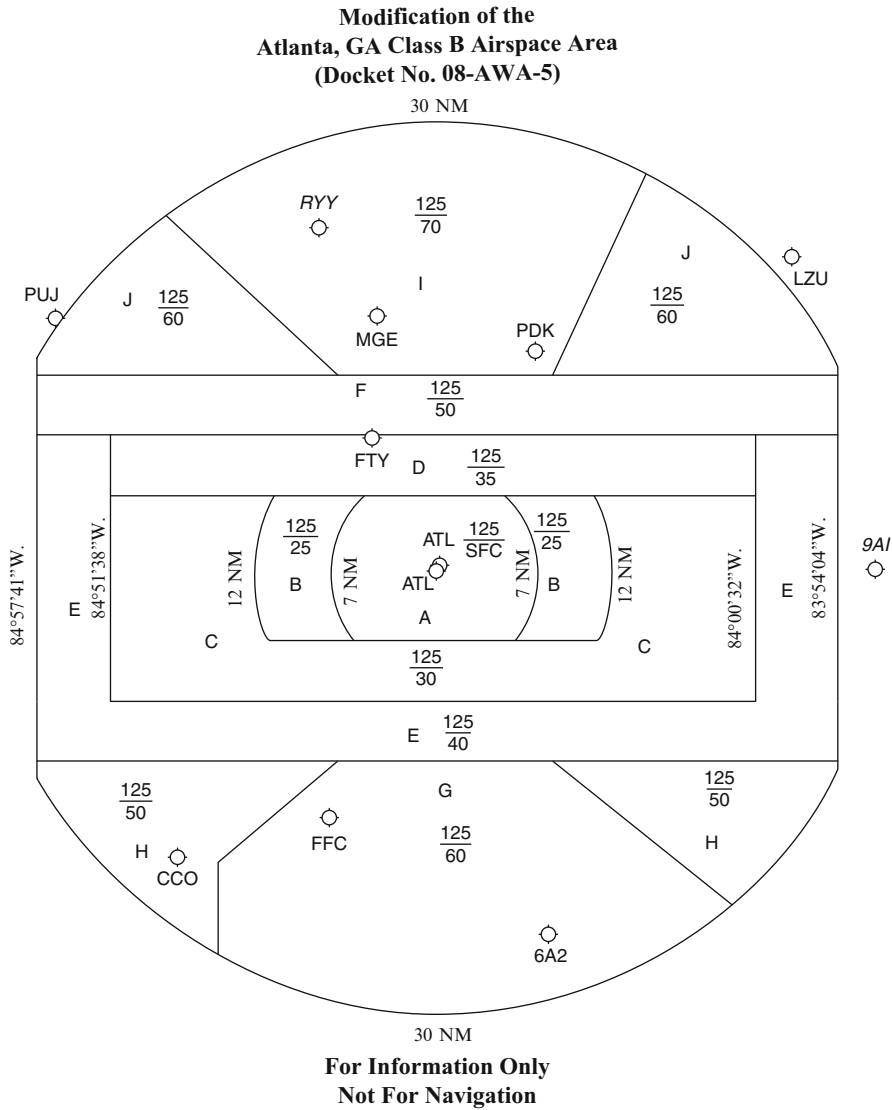
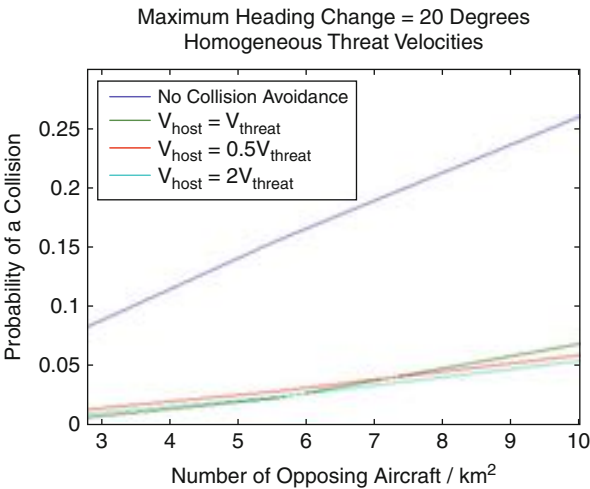


Fig. 79.52 Atlanta International Airport region airspace map (AOPA 2013a)

airspace (see Fig. 79.52) extends up to 56 km from the airport itself, the physical area within which the runways are contained can be modeled as a rectangle 4.64 km wide by 3.7 km long; see Fig. 79.53 (AOPA 2013a; FAA 2013). This results in an area of 17.17 km² through which it is assumed all aircraft serviced by ATL must pass. The average duration of a scenario in Sect. 79.5.5 is 36.76 s and there are 858,458.3 such time periods within a year. Assuming a uniform distribution

Fig. 79.54 Data in Fig. 79.43 for lower airspace densities



of aircraft arrival and departures, throughout any given day, there must be 1.1 flying aircraft over ATL every simulation period. This gives rise to an average airspace density of 0.0641 aircraft/km² which is 866.8 times less dense than the maximum airspace density simulated. Stated another way, if ATL operated under the maximum airspace density used in this research, an aircraft would be either taking off or landing every 39 ms (25.9 aircraft/s).

Figure 79.54 shows the data in Fig. 79.43 enlarged to provide more detail throughout lower airspace densities. For cases where the host and the opposing aircraft have homogeneous velocities and the heading modification angle is limited to 20°, the probability of avoiding a collision is 99.995 % when the airspace density is 1.66 aircraft/km². As the airspace density continues to decrease, the probability of avoiding a collision approaches 100 % when the airspace density is 1.5 aircraft/km², nearly 23 times the airspace density of ATL.

79.6 Conclusion

This chapter presents two components which aim to improve the safety of unmanned aircraft systems and ease their future integration into the national airspace. These components are as follows: a lightweight RADAR sensor and a computationally efficient method for determining collision avoidance maneuvers.

Due to the limitations of the available hardware, the prototype RADAR sensor is evaluated on a small scale using mobile RADAR reflectors to simulate opposing aircraft. This process revealed a number of limitations of the developed hardware, primarily its vulnerability to multipath reflections in addition to the limited range and measurement update rates. Nevertheless, the construction and experimental evaluation of a miniature RADAR sensor capable of localizing moving targets

demonstrates the feasibility of RADAR-based collision avoidance. While the hardware is evaluated on a small scale, the collision avoidance algorithm is evaluated extensively in order to characterize its efficacy well beyond the current maximum airspace density.

Flying implementations will, however, require more than is presented here. First, higher RADAR transmit power levels are required to achieve the detection ranges required for collision avoidance. Next, the collision avoidance algorithm should be expanded to incorporate three-dimensional collision avoidance maneuvers. Finally, collaborative collision avoidance must be incorporated to address the issues that arise when numerous vehicles are equipped with this technology. Nevertheless, this chapter provides a foundation to support the idea of RADAR-based collision avoidance for unmanned aerial vehicles.

References

- ACI, Traffic movements 2010 final. Technical report, Airports Council International, 2011
- ADS-B, *ADS-B Explained* (2011)
- AeroVironment, UAS advanced development: Black Widow (2013)
- AOPA, *Atlanta Class B Airspace Changes Take Effect March 7* (Aircraft Owners and Pilots Association, 2013a)
- AOPA, What is general aviation (2013b)
- A. Bauer, *Christian Hulsmeyer and About the Early Days of Radar Inventions* (2005)
- E. Britannica, Radar (2013)
- C.A. Company, *Pilot's Operating Handbook: Skyhawk 1978 Model 172N* (Cessna Aircraft Company, 1977)
- FAA, *Instrument Flying Handbook* (US Department of Transportation, 2001)
- FAA, 14 cfr part 91 (2010)
- FAA, *Intorduction to TCAS II* (Federal Aviation Administration, version 7.1 edn., 2011)
- FAA, Order jo 7110.65u. Technical report, Federal Aviation Administration, 2012a
- FAA, *TCAS Home Page* (2012b)
- FAA, *Atlanta Hartsfield Jackson International Airport Diagram* (Federal Aviation Administration, 2013)
- Flarm.Inc, Flarm: motivation. Technical report, FLARM Inc., 2008
- P.G. Gallman, *RADAR Reflectors for Cruising Sailboats*, chapter 6 (Ulyssian Publications, Canada, 2005)
- G. Goebel, *Microwave Radar at War*, chapter 4 (2013)
- N. Grumman, An/zpy-1 starlite small tactical radar. Technical report, Northrop Grumman, 2013
- H. Hertz, *Electric Waves Being Researches on the Propagation of Electric Action with Finite Velocity Through Space* (Dover, New York, 1893)
- Hobbico CS-80 Servo, *Hobbico CS-80 Giant Scale 2BB Servo Product Description* (Tower Hobbies, 1999)
- H. Hobby, Ultra stick 25e arf product description. Technical report, Horizon Hobby Inc., 2013
- Hokuyo.Inc, *Scanning Laser Range Finder URG-04LX-UG01 Specifications* (Hokuyo Automatic Co. Ltd., 2009)
- ImSAR.Inc, Imsar product page (2011)
- T. Inc, *Bell Jetranger III Rotorcraft Flight Manual* (Bell Helicopter Textron Inc, version 8 edn., 2000)
- ITU, *Recommendation P.676-9: Attenuation of Atmospheric Gases* (International Telecommunication Union, 2012)

- D. Minoli, *Satellite Systems Engineering in an IPv6 Environment*, chapter 3 Auerbach Publications Taylor & Francis Group 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487–2742 (2009)
- A. Moses, M.J. Rutherford, K.P. Valavanis, Radar-based detection and identification for miniature air vehicles, in *IEEE International Conference on Control Applications*, Denver (IEEE, 2011), pp. 933–940
- R. Niles, Cessna to offer diesel skyhawk (AVweb, 2007)
- Pololu.Inc, *Pololu 3pi Robot User Guide* (Pololu Corporation, 2013)
- SAI.Inc, Garmin gts850 tcas (2011), <http://sarasotaavionics.com/avionics/gts850-tcas-two-antennas>
- SARA.Inc, Optical collision avoidance (2011a)
- SARA.Inc, Uav acoustic collision-alert system (2011b)
- Selex, Loam: Laser obstacle avoidance system (2011)
- S. Shen, N. Michael, V. Kumar, Autonomous multi-floor indoor navigation with a computationally constrained mav, in *International Conference on Robotics and Automation*, Shanghai (IEEE, 2011)
- M.I. Skolnik, *Introduction to Radar Systems* (McGraw-Hill, New York, 1981)
- M. Solutions”, *MDU4210 Datasheet* (Microwave Solutions, 2013)
- P. Tait, *Introduction to Radar Target Recognition* (Institution of Engineering and Technology, 2006)
- U.S.DOD, Rq-4 global hawk. Technical report, U.S. Air Force, 2012
- T.V. Wallace, R.J. Jost, P.E. Schmid, *Principles of Modern RADAR: Basic Principles*, chapter 10, Dulles, VA (2010)
- Y. Watanabe, A.J. Calise, E.N. Johnson, Vision-based obstacle avoidance for uavs, in *Guidance, Navigation and Control Conference*, Hilton Head (American Institute of Aeronautics and Astronautics, 2007)
- Zaon Flight Systems Inc. Addison, TX (2007)
- ZAON, *PCAS MRX Overview* (2011)

Assessment of Detect and Avoid Solutions for Use of Unmanned Aircraft Systems in Nonsegregated Airspace

80

Joram Verstraeten, Martijn Stuip, and Tom van Birgelen

Contents

| | | |
|--------|--|------|
| 80.1 | Introduction | 1956 |
| 80.1.1 | Background and Scope | 1956 |
| 80.1.2 | Objective | 1957 |
| 80.1.3 | Approach | 1957 |
| 80.2 | Detect and Avoid Requirements | 1958 |
| 80.2.1 | Generic Requirements of the DAA Solution | 1958 |
| 80.2.2 | Requirements Under Development by European Organizations | 1961 |
| 80.3 | Detect and Avoid Solutions | 1964 |
| 80.3.1 | Sensor Suite | 1965 |
| 80.3.2 | Candidate Detect and Avoid Solutions | 1966 |
| 80.4 | Assessment of Detect and Avoid Solutions | 1968 |
| 80.5 | Conclusions and Recommendations | 1978 |
| | References | 1979 |

Abstract

Unmanned Aircraft Systems (UASs) emerge as new possibility for civil and military aircraft applications, without having the need for a pilot onboard the aircraft. Although there is great potential, there is also the need to show that the introduction in nonsegregated airspace can be done without endangering other (manned) traffic. In this chapter the most promising candidate solutions for Detect and Avoid (DAA) for use of UAS in nonsegregated airspace are assessed against a set of DAA requirements. A DAA solution is the combination

J. Verstraeten (✉) • T. van Birgelen
National Aerospace Laboratory of the Netherlands – Air Transport Safety Institute, Amsterdam,
The Netherlands

e-mail: Joram.Verstraeten@nlr-atsi.nl; Tom.van.Birgelen@nlr-atsi.nl

M. Stuip

National Aerospace Laboratory of the Netherlands, Amsterdam, The Netherlands

e-mail: Martijn.Stuip@nlr.nl

of the sensor suite, the avoidance algorithms, and the method of operation. The functions of a DAA solution are collision avoidance and, in uncontrolled airspace, separation provision. The focus of this chapter is the detection of conflicting traffic by the sensor suite. The sensor suite needs to be able to detect different classes of conflicting traffic in varying environments. Requirements for DAA solutions have to be set, and DAA systems must then be developed to meet these requirements.

ATM requirements for use of UAS in nonsegregated airspace have been drafted by EUROCONTROL (2007). This set includes a subset of requirements that are directly or indirectly of influence on the sensor suite used to Detect and Avoid other traffic. This subset is used as baseline for the establishment of DAA requirements and expanded by development of five additional generic requirements based on the main tasks of a DAA solution: (1) detection of other traffic, (2) tracking of other traffic and assessing if there is a conflict, (3) if there is a conflict, determining which evasive maneuver is to be executed, and (4) executing the selected maneuver. Five candidate solutions are assessed against the requirements: a noncooperative solution, a cooperative solution, and three solutions mixing cooperative and noncooperative sensors.

Overall it is concluded that it is a great challenge to develop a collision avoidance solution for UAS with a satisfactory level of safety. It is even more difficult to develop a DAA system that is capable of both collision avoidance and separation provision. Therefore, collision avoidance is considered a first step in developing DAA solutions for UAS, restricting the operations to flights in controlled airspace with ATC providing separation at all times. It is recommended to continue to develop suitable methods to perform the safety analysis for introduction of UAS equipped with DAA solutions in nonsegregated airspace.

80.1 Introduction

80.1.1 Background and Scope

Historically, pilots have to use their eyes as the sensors to detect other aircraft. It is commonly agreed that like manned aircraft, UAS should adhere to the general rules of the air. According to ICAO Annex 2 (ICAO 2005), all pilots need to exercise vigilance for the purpose of detecting potential collisions and are responsible for taking collision avoidance action when in flight, irrespective whether the flights is under instrument flight rules (IFR) or visual flight rules (VFR) and irrespective of Air Traffic Control (ATC) services. UAS also needs to adhere to the right-of-way rules, as can be found in ICAO Annex 2. For a UAS to adhere to the rules of the air and the right-of-way rules, a DAA solution is needed. The DAA solution should include a collision avoidance function at all times. A separation provision function is needed where this is not provided by ATC. The DAA solution must detect at a sufficient range to allow a timely maneuver to avoid a collision.

In the future both manned and unmanned traffic are expected to be integrated in nonsegregated airspace according to established procedures. This study investigates the current technological capabilities to support UAS with the detection and avoidance of potential traffic conflicts in nonsegregated airspace, thereby building on public references to studies on DAA solutions in Europe. A DAA solution is the combination of the sensor suite, the avoidance algorithms, and the method of operation of the UAS. The functions of a DAA solution are collision avoidance and, in uncontrolled airspace, separation provision. The sensor suite needs to be able to detect different classes of conflicting traffic in varying environments. The avoidance algorithms must assure that the conflicting traffic is avoided by taking appropriate avoidance action. The method of operation determines the autonomy of the DAA solution and is therefore of influence on the required performance of the systems. Requirements for DAA solutions have to be set, and technical systems must be provided to meet these requirements.

The mode of operation of the UAS considered for this study involves a designated UAS operator (DUO). It is assumed the detection of traffic and assessment of collision course is executed automatically, informing the DUO about the conflicting traffic. The focus is the detection of conflicting traffic by the sensor suite which is part of the DAA solution. This choice is made because most European research on detect and avoid focus on the needed sensors. It is assumed that avoidance will be executed automatically with human approval or veto. The type of UAS considered are those categorized as medium range or larger (i.e., a maximum take-off weight (MTOW) of 1,250 kg and up (van Blyenburgh 2011)). Due to this scope of UAS categories, it is believed the accommodation of sensors is not a major issue and is not further discussed in this chapter. The type of aircraft considered as traffic to be detected and avoided is UAS or manned aircraft with a take-off weight of 1,250 kg and up. The airspace classes considered are A–G. Airspace classes A–E are referred to as controlled airspace by ICAO, and subsequently airspace F and G are referred to as uncontrolled airspace. Only technologies to detect and avoid other airborne aircraft are considered. Technologies to avoid terrain, objects on the ground, and adverse weather are not considered. Legal aspects of UAS, DAA, and non-nominal events other than traffic on collision course are considered out of scope.

80.1.2 Objective

The objective of this study is to determine what the most promising candidate technological solutions are for detect and avoid solutions for UAS in nonsegregated airspace considering the requirements imposed.

80.1.3 Approach

The approach of this study is to match the most feasible DAA requirements with the most feasible DAA solutions. Therefore, the trend of requirement definition and

the viability and feasibility of these requirements is determined. The key technical issues while developing these requirements and the research efforts to resolve these issues are identified. This leads to a final set of requirements that is deemed most likely. This set of requirements is then matched to five potential DAA solutions, so as to determine which candidate solution is most feasible and viable in terms of satisfying the requirements. The five solutions are determined considering the technologies available and European studies performed to assess these technologies. As Detect and Avoid is considered a “dual” technology that can be used for military and civil UAS, both civil and military initiatives will be considered in this study.

80.2 Detect and Avoid Requirements

This section discusses the requirements for the detect and avoid solution of UAS. It first discusses the generic requirement for the DAA solution to fulfill its primary functions. A set of generic requirements are drafted to be used in the assessment of DAA solutions as given in the final part of this chapter. Next, the requirements in development at by various European standardization and regulatory organizations are described and discussed. A relevant set of requirements is obtained to use for assessment of DAA solutions.

80.2.1 Generic Requirements of the DAA Solution

The two primary functions of a DAA solution are the provision of collision avoidance and, in uncontrolled airspace, separation provision. Table 80.1 lists the responsibilities regarding collision avoidance and separation provision per airspace class and flight rules for manned aviation. It is assumed these responsibilities are unaltered for UAS, where the pilot’s role will be fulfilled by either a DUO with the help of a DAA solution or by the DAA solution itself. Therefore, collision avoidance will always be a function of the DAA solution. In case ATC is not responsible for

Table 80.1 Responsibilities related to airspace and flight rules

| Flight rules <i>responsibility</i> | Airspace classes | | | | | | |
|--|------------------|-------|------------|------------|------------|------------|-------|
| | A | B | C | D | E | F | G |
| IFR/IFR <i>Separation provision</i> | ATC | ATC | ATC | ATC | ATC | Pilot w/TA | Pilot |
| IFR/VFR <i>Separation provision</i> | – | ATC | ATC | Pilot w/TA | Pilot w/TA | Pilot | Pilot |
| VFR/VFR <i>Separation provision</i> | – | ATC | Pilot w/TA | Pilot w/TA | Pilot w/TA | Pilot | Pilot |
| All <i>Collision avoidance</i> | Pilot | Pilot | Pilot | Pilot | Pilot | Pilot | Pilot |

TA ATC traffic advisory

separation provision, the DAA solution needs to fulfill both the separation provision and collision avoidance function. In the latter case ATC, if available, can aid the operator of the UAS (or the UAS itself) by giving traffic advisories, although it has not been established how to provide a UAS traffic advisories.

For both separation provision and collision avoidance, there are four generic tasks of a DAA solution. The requirements that arise from these tasks can be different for separation provision and collision avoidance. The four tasks are

- 1. Detect of other traffic.
- 2. Track other traffic and assess if there is a conflict.
- 3. If there is a conflict, determine which evasive maneuver is to be executed.
- 4. Execute the selected maneuver.

As indicated in the introduction, the focus of this study is the detection of conflicting traffic; task 1 and task 2 combined. Task 3 must be done by algorithms and communication between traffic. Task 4 is either executed by a DUO or UAS itself again using algorithms. For the purpose of drafting the requirements, task 1 and task 2 are modeled in an event tree, given in Fig. 80.1.

Traffic in proximity of the UAS needs to be detected. Each DAA solution can detect traffic in the volume of airspace that it scans, the surveillance volume. To be able to detect other traffic in time, the DAA solution needs to scan a sufficient large part of the airspace; in other words the surveillance volume needs to be sufficiently large. The surveillance volume is made up by two components: the field

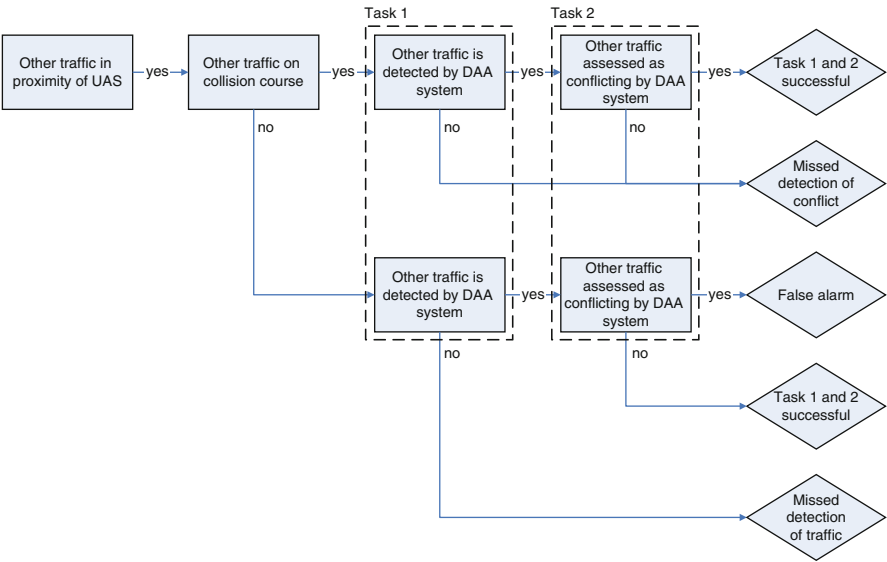


Fig. 80.1 An event tree including task 1 and task 2 of a DAA solution

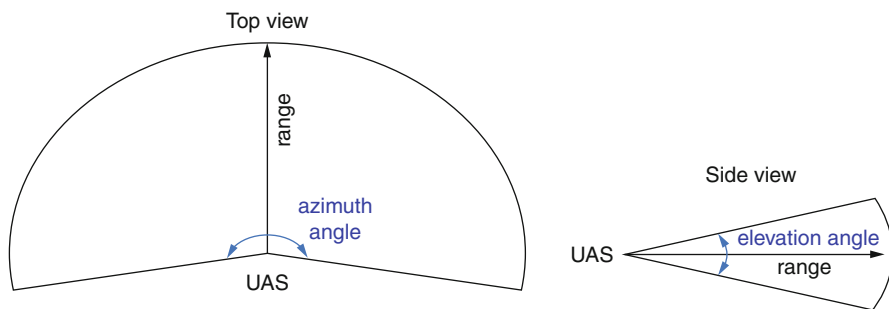


Fig. 80.2 Field of regard and range

of regard (the combination of azimuth and elevation) and the detection range; see Fig. 80.2.

This results in the following two, of five, generic requirements (GR):

- GR01: *The DAA system must provide a sufficient surveillance volume by having a sufficient field of regard.*
- GR02: *The DAA system must provide a sufficient surveillance volume by having a sufficient detection range.*

If these two requirements are met, the performance of the sensor suite is determined by its success rate (the ratio of detected traffic to total traffic in the surveillance volume). This ratio needs to be sufficiently high. An implicit resultant of this requirement is the need to detect both cooperative and noncooperative traffic. A failure of task 1 results in a missed detection of traffic, which can result in the missed detection of a conflict. The latter is of course unwanted since it can result in an actual collision. The failure to detect traffic that is not conflicting is also unwanted since one cannot depend on providence in designing a DAA solution. Summarizing, the third requirement is

- GR03: *The success rate of traffic detection must be sufficient.*

If task 1 is fulfilled, the sensor suite will detect (nearly) all traffic in its surveillance volume. However, not all traffic will be conflicting. The system needs to assess if traffic is conflicting and needs to include a classification of traffic in case the traffic mix includes balloons, gliders, and powered aircraft. The classification is needed to be able to comply with right-of-way rules. Erroneous assessments can be caused by inaccurate sensor information of the other traffic's range, bearing, altitude, and lateral, and vertical speed. Also incorrect assessment of traffic that is not conflicting is unwanted, since it leads to false alarms and can possibly lead to a maneuver that induces a conflict. Summarizing, the fourth requirement is

- GR04: *When traffic has been detected, the rate of correct assessments if a situation is a conflict or not must be sufficient.*

Since the function of a DAA solution is to provide collision avoidance and in some instances separation provision, the solution must be able to detect traffic conflicts. Any missed conflict can lead to a loss of separation or, worse, a collision. To emphasize this need the third and fourth requirement are combined into a fifth requirement:

GR05: *The DAA system missed conflict detection rate must be kept at an acceptable level.*

80.2.2 Requirements Under Development by European Organizations

A DAA solution has to fulfill the generic requirements as defined in the previous section. Complimentary to those requirements are those requirements that have to be met if a UAS is operated in nonsegregated airspace. For the analysis of those requirements, four European organizations drafting requirements are considered. The organizations considered are the European Aviation Safety Agency (EASA), EUROCONTROL, the European Organization for Civil Aviation Equipment (EUROCAE), and the Flight in Non-Segregated Airspace (FINAS) group of the North Atlantic Treaty Organization (NATO).

EASA develops the regulatory framework for civil UAS with a maximum take-off weight larger than 150 kg. In 2008, EASA stated the following on DAA:

Airworthiness certification is considered to address the intrinsic safety of the UAS. “Sense and Avoid” falls outside this area as its sole purpose is for anti-collision. The operating criteria on which it relies to adequately perform its function are dependent on the airspace being used and the aircraft flying into it. Such criteria should be defined by the authorities responsible for the safety regulation of air navigation services. (EASA 2008)

In Europe, the organization that usually dealt with harmonized safety regulation of air navigation services was EUROCONTROL. However, EASA will gradually take over this responsibility in the coming years.

EUROCONTROL distinguishes two types of air traffic: Operational Air Traffic (OAT) and General Air Traffic (GAT). GAT flights are all movements of civil aircraft, as well as all movements of State aircraft, when these movements are carried out in accordance with the procedures of ICAO. OAT flights are all flights which do not comply with the provisions stated for GAT and for which rules and procedures have been specified by appropriate national authorities. The EUROCONTROL UAV OAT Task Force has drafted Air Traffic Management (ATM) specifications for the use of military UAS as OAT outside segregated airspace (EUROCONTROL 2007). This set includes a subset of requirements that are directly or indirectly of influence on the sensor suite used to Detect and Avoid other traffic. These requirements are independent of the chosen solutions and are based on three main principles:

- UAS operations should not increase the risk to other airspace users.
- ATM procedures should mirror those applicable to manned aircraft.
- The provision of air traffic services to UAS should be transparent to ATC controllers.

It is considered likely that for the use of UAS as GAT in nonsegregated airspace, similar specifications will be developed. The EUROCONTROL Collision Avoidance Requirements for UAS (CAUSE) study confirm this (EUROCONTROL 2010).

According to EUROCONTROL a DAA solution should achieve an equivalent level of safety to a manned aircraft (EUROCONTROL 2007). Since the first discussions on certification of UAS, many attempts have been undertaken to quantify this level of safety for manned aviation. Clearly, when it is not possible to quantify all the elements related to detect and avoid, it is not possible to define an equivalent level of safety. The alternative is to establish a target level of safety (TLS) for UAS. A safety assessment is then needed to prove the TLS can be met by a particular UAS operation. UAS proponents have to work closely with the authorities (e.g., FAA and EASA) to determine what the TLS should be or how it should be defined. Although not easy, the task focus can initially be on a comparison of UAS with manned aircraft.

The NATO FINAS group has also drafted DAA requirements, which are restricted to the situation where separation provision responsibility rests with the designated UAS operator (NATO 2007). The FINAS requirements have been evaluated by the NATO NIAG Steering Group 134. The steering group's conclusion was that the FINAS requirements are incomplete, not coherent in level of detail and often without proper rationale (NATO 2010). Their recommendation toward NATO is to use the EUROCONTROL subset of DAA requirements.

EUROCAE is a European body in which industries work on commonly accepted standards for specific implementations. Generally, showing compliance with these standards can serve industry as certification evidence to convince authorities that their system is safe to use and also increases interoperability between products of different manufacturers. EUROCAE Working Group 73 is tasked to develop a requirement framework that will support civilian UAS airworthiness certification and operational approvals (Hawkes et al. 2008; Kallevig 2011). EUROCAE WG73 has expressed that it is unlikely that a single company producing UAS will make the step as a certification applicant for a UAS to fly in nonsegregated airspace. Focusing on DAA only, WG73 expects that huge efforts, beyond the budget of most industry, are needed to show compliance for requirements such as equivalent level of safety with manned aviation and compliance of DAA solutions with existing ACAS equipment. Instead, a standard would collect the common requirements of potential users, share data for establishing safety, performance and interoperability criteria, and validate the result. This should simplify the process for each applicant who can conform and for the approving authority. WG73 proposes a stepwise UAS integration process:

1. To develop initial standards offering significant additional flight capabilities in nonsegregated airspace compared to flight in segregated airspace, in a reasonable timeframe and with an affordable effort (i.e., UAS nominally flying under IFR with ATC providing separation from other airspace users at all times and UAS flight within Visual Line Of Sight (VLOS) of the DUO)

2. To benefit from the experience of initial standards to develop more complex steps to follow

As a consequence, the result will be a number of EUROCAE standards (e.g., Minimum Aviation System Performance Standards (MASPS) and Minimum Operational Performance Standards (MOPS)) that express system requirements for a specific system or solution. By complying with this standard, the industry is then better able to develop a DAA solution that becomes certifiable and accepted by aviation authorities. Hopefully, this approach will be better than trying to develop several independent and proprietary DAA solution. EUROCAE is currently drafting a framework for such UAS standards. At this stage, no concrete DAA requirements have been published by EUROCAE WG-73 in the public domain.

A continuing discussion is the separation minima; a DAA solution should accomplish if no ATC separation is provided. There are no prescribed ICAO separation minima for manned aircraft where responsibility for separation rested with the onboard pilot. Instead, according to EUROCONTROL it is only necessary that aircraft should not be operated in such proximity to other aircraft as to create a collision hazard (EUROCONTROL 2007). However, industry required something less vague. As a consequence, a practical minimum separation to be achieved by a UAS pilot-in-command is proposed. According to EUROCONTROL several organizations quote or imply that 500 ft is an appropriate and acceptable miss distance for UASs (EUROCONTROL 2007). In the USA, the FAA view of “well clear” (i.e., so as to not represent a collision hazard) is a minimum separation of 500 ft between aircraft. To a considerable degree, this figure is accepted by the Joint JAA/EUROCONTROL UAV Task Force as the basis for recommending work to identify minimum performance standards (MPS) for future DAA systems. Industry itself regards 500 ft as a minimum “worst-case separation” distance for DAA (EUROCONTROL 2007). Finally, in the context of maneuvering between aircraft to achieve safe separation, NATO defines 500 ft as “well clear” (NATO 2007). The use of 500 ft vertical separation is routine between manned aircraft and should therefore not cause undue concern to other airspace users; the application of 500 ft horizontal separation could generate a heightened sense of collision risk. An increase in horizontal separation to 0.5 nm would reduce this perception and also the collision risk and is therefore preferable.

According to EUROCONTROL, ATM regulations and procedures for UASs should mirror as closely as possible those applicable to manned aircraft. UAS-specific ATM procedures should therefore only be implemented where the absence of an onboard pilot – particularly in combination with loss of control data-link – generates a need for special arrangements. Otherwise, the provision of an air traffic service to a UAS should be transparent to the ATC controller and other airspace users. Due to the maturity of the requirements, they can be used in the assessment of DAA solutions and more specifically the sensor suite. A number of requirements are independent of the sensor suite of the DAA solution either because they are relevant for ATM only or for other operational aspects of the UAS. Some requirements are not deemed relevant for this study since they are part of the DAA solution but not dependent on the mix of sensors used; one

such requirement is the requirement to notify the DUO in case of separation violation. For this study only those requirements that are directly or indirectly of influence on the sensor suite are considered. These requirements are given below (EUROCONTROL 2007) (note that the numbering from EUROCONTROL is adopted here as well).

Specification UAV4 UAVs should comply with the right-of-way rules as they apply to other airspace users.

Specification UAV6 For VFR flight by UAVs, the UAV pilot-in-command should utilize available surveillance information to assist with separation provision and collision avoidance. In addition, technical assistance should be available to the pilot-in-command to enable him to maintain VMC and to detect and avoid conflicting traffic. An automatic system should provide collision avoidance in the event of failure of separation provision.

Specification UAV7 A UAV DAA system should enable a UAV pilot-in-command to perform those separation provision and collision avoidance functions normally undertaken by a pilot in a manned aircraft, and it should perform a collision avoidance function autonomously if separation provision has failed for whatever reason. The DAA system should achieve an equivalent level of safety to a manned aircraft.

Specification UAV9 Implementation of separation provision and collision avoidance functions in a DAA system should as far as is reasonably practicable be independent of each other. In execution, they should avoid compromising each other.

Specification UAV10 Within controlled airspace where separation is provided by ATC, the separation minima between UAVs operating IFR and other traffic in receipt of a separation service should be the same as for manned aircraft in the same class of airspace.

Specification UAV11 Where a UAV pilot-in-command is responsible for separation, he should, except for aerodrome operations, maintain a minimum distance of 0.5 NM horizontally or 500 ft vertically between his UAV and other airspace users, regardless of how the conflicting traffic was detected and irrespective of whether or not he was prompted by a DAA system.

Specification UAV12 Where a UAV system initiates collision avoidance autonomously, it should achieve miss distances similar to those designed into ACAS. The system should be compatible with ACAS.

80.3 Detect and Avoid Solutions

Detect and Avoid is the capability of a UAS to remain well clear from and avoid collision with other airborne traffic; therefore, it is a combination of self-separation and collision avoidance. In this research a DAA solution is defined as the combination of the sensor suite and the avoidance algorithms and the method of operation of the UAS in order to detect and avoid conflicting traffic. The method of operation is not further discussed; it is assumed a dedicated UAS operator will be in the loop. Many studies do, either briefly or more elaborately, consider operational

issues of UAS as part of their work. The issues are related to UAS integration in an operational environment and not on detect and avoid specifically. In the short term, it is not expected that the UAS will fly completely autonomous. Only “last ditch” collision avoidance could be made autonomous. Avoidance algorithms are not widely discussed since the initial focus seems to be on the detect task and not on the avoid task. This is understandable because in order to be able to avoid air traffic, it has to be detected first. There is no detailed information available in the public domain regarding the avoidance algorithms used. One European research project does state that they aim at standardized avoidance logic, and as consequence the algorithms will be in the public domain (MIDCAS 2010a). The focus is on the sensor suite, being a combination of sensors used. A reason for the need for more than one sensor would be the ability to detect both cooperative and noncooperative traffic.

80.3.1 Sensor Suite

The European research projects and the sensors studied are shown in Table 80.2. The background of the projects discussed in this section is publicly available (USICO 2004a,b; EDA 2007; Hutchings et al. 2007; Korn and Edinger 2008; Selier et al. 2008; MIDCAS 2010a,b).

All studies except OUTCAST combine EO/IR sensors with radar technology. Cooperative traffic can be detected better using a cooperative technology such as TCAS (note that the concept is named ACAS). TCAS is equipment compliant with the ACAS concept. In this chapter the term TCAS is used (When referring to the specific equipment TCAS II is meant). Other types of sensors that have not been studied by the above research projects are active and passive acoustic systems, active millimeter wave systems, and collision avoidance systems such

Table 80.2 Type of sensors studied by different European research projects (Verstraeten and Selier 2011)

| | TCAS | ADS-B | Radar | LIDAR LADAR | IR camera | Daylight (EO) camera | Passive millimeter wave |
|----------------|----------------|----------|-------------|----------------|-------------|----------------------------|-------------------------------|
| Type of sensor | <i>C</i> | <i>C</i> | <i>NC-A</i> | <i>NC-A</i> | <i>NC-P</i> | <i>NC-P</i> | <i>NC-P</i> |
| OUTCAST | ▪ | | | | ▪ | ▪ | |
| WASLA-HALE | | | ▪ | ▪ | ▪ | ▪ | |
| USICO | | | ▪ | ▪ | ▪ | ▪ | ▪ |
| ASTRAEA | | ▪ | ▪ | | ▪ | ▪ | |
| DAA study EDA | ▪ | | ▪ | ▪ ^a | ▪ | ▪ | |
| MIDCAS | ▪ ^b | ▪ | ▪ | | ▪ | ▪ | |

C cooperative, *NC* noncooperative, *A* active, *P* passive
^aOnly in medium-term solution
^bThe surveillance part of TCAS equipment (transponder interrogator) is considered as a candidate MIDCAS sensor (MIDCAS 2010a)

as FLARM (a commercially available active and cooperative traffic and collision warning system capable of giving traffic advisories) and TCAD (Traffic Collision Avoidance Device: a passive collision avoidance device capable of giving traffic advisories that is commercially available).

The most important ongoing projects with a focus on DAA technology are MIDCAS and ASTRAEA. The importance of these projects is due to the stakeholders involved and the budget available. MIDCAS is a project of the European Defense Agency (EDA) and 14 partners with a budget of 50 million euros. It started in 2009 and will finish in 2013. ASTRAEA is a research program of a UK consortium with a budget of 32 million pounds.

MIDCAS has as objective to demonstrate a DAA system for UAS able to fulfill the requirements for traffic separation and midair collision avoidance in nonsegregated airspace for both cooperative and noncooperative intruders. ASTRAEA aims to ensure that UAS operation is transparent to manned aviation. The behavior of the UAS in flight must be consistent with that expected by manned aviation. Within the ASTRAEA framework, there is a dedicated project focusing on collision avoidance systems. The project seeks to verify the merits of enabling technologies and system capabilities that could be used by UAS for collision avoidance.

80.3.2 Candidate Detect and Avoid Solutions

Five candidate solutions for DAA, representing a number of combinations of sensor technologies, are derived. The DAA solution should include a collision avoidance function at all times. A separation provision function is needed where this is not provided by ATC. For the five candidate solutions the following sensors are considered: TCAS, ADS-B, Radar, IR, and EO. Next, the five conceptual solutions are described, starting with a full noncooperative concept and a full cooperative concept:

1. *Full Noncooperative Concept* This concept combines Radar sensors with EO/IR. This concept is inspired by projects such as WASLA-HALE and USICO. Radar is capable of quickly scanning a large area, also in IMC, although it requires more energy to detect objects through clouds and rainy conditions (when more energy is absorbed) than in VMC. In general powerful Radar introduce more weight and require more power. Besides traffic detection Radar can also be used to detect terrain and weather phenomena. Although this is beyond the scope of DAA for air traffic, it may enhance UAS intrinsic safety. Once directed properly on the conflicting aircraft, EO/IR provides a highly accurate bearing toward the other aircraft and can be used to classify the object by visual inspection.
 - a. Advantages: requires no cooperation from the object to be sensed and avoided.
 - b. Disadvantages: Detection of objects highly depends on a single type of sensor (Radar) with limited bearing accuracy.
2. *Full Cooperative Concept* This concept combines two cooperative sensors: TCAS II and ADS-B. TCAS II is de facto system for collision avoidance and is mandated in Europe for larger aircraft with a maximum take-off weight greater

than 5,700 kg or authorized to carry more than 19 passengers. Beside detection TCAS II can also advise the pilot with an avoidance maneuver. A version of TCAS that is capable of automatically performing the resolution advisory maneuver has been certified (Airbus 2009). TCAS can therefore be considered the most mature collision avoidance system available. Detection and avoidance is based on information contained in the reply of an interrogated mode-S transponder in the other aircraft. If the conflicting aircraft is also equipped with TCAS II, the avoidance maneuvers of both aircraft are “negotiated” by the systems to maintain maximum safety. An ADS-B equipped aircraft automatically transmits the aircraft identity, position, altitude, and intent information (among others) to other airspace users. With this information other airspace users, but also Air Traffic Control, are able to construct a traffic picture. This way ADS-B can provide a basis for separation provision, which is especially useful when separation is not provided by ATC. The completeness of such a traffic picture depends on the population of nearby aircraft equipped with “ADS-B out,” which at this moment is not yet the case in Europe.

- Advantages: Once detected good tracks may be obtained of nearby traffic, and ADS-B may provide information useful for classification.
- Disadvantages: With this system it is not possible to detect noncooperative traffic.

The next three concepts mix both cooperative and noncooperative sensors. These concepts are described below:

3. *Mixed Concept 1* This concept combines one cooperative sensor TCAS II with the noncooperative sensor EO/IR. It is based on the work performed in the project OUTCAST that had the objective to investigate a DAA solution with equipment readily available on the market. The use of operation is thought to be as follows: detect other traffic with TCAS and estimate if there is a reasonable chance of collision course. If that is the case, use EO/IR to classify the intruder (for separation provision) and obtain a better bearing toward the intruder. With this information an evasive maneuver can be calculated.
 - Advantages: simple concept.
 - Disadvantages: limitations of TCAS sensing accuracies (bearing), detection of noncooperative traffic only by EO/IR, difficult to keep track of multiple intruders by EO/IR.
4. *Mixed Concept 2* This concept extends Mixed Concept 1 (TCAS and EO/IR) with Radar. It is based on the work performed in the DAA study by EDA. The use of operation is thought to be as follows: detect other traffic with TCAS and Radar and, estimate if there is a reasonable chance of collision course. If that is the case, use EO/IR to classify the intruder (for separation provision). With this information calculate an evasive maneuver and obtain a better bearing toward the intruder.
 - Advantages: In addition to Mixed Concept 1, detection of noncooperative traffic can primarily be done by Radar.
 - Disadvantages: Use of Radar adds more weight to the UAS and requires more power.

Table 80.3 Summary of five candidate detect and avoid solutions

| | 1. Full non-cooperative concept | 2. Full cooperative concept | 3. Mixed concept 1 | 4. Mixed concept 2 | 5. Mixed concept 3 |
|-------|---------------------------------|-----------------------------|--------------------|--------------------|--------------------|
| EO/IR | • | | • | • | • |
| Radar | • | | | • | • |
| TCAS | | • | • | • | |
| ADS-B | | • | | | • |

5. *Mixed Concept 3* This concept combines one cooperative sensor (ADS-B) and two noncooperative sensors (Radar and EO/IR) and only differs from Mixed Concept 2 in the sense that TCAS is replaced by ADS-B as the cooperative sensor. It is based on the work performed in the ASTRAEA project. The rationale for the replacement of TCAS is that through ADS-B more accurate information about air traffic can be obtained, compared to TCAS. In that case, for the collision avoidance function another solution must be in place (not necessarily performed by ADS-B only, but likely a combination of all sensors). ADS-B may provide information useful for classification.

- Advantages: Mix of sensors gives a high probability of detecting all nearby traffic.
- Disadvantages: ADS-B not mandated.

The five concepts are summarized in Table 80.3.

80.4 Assessment of Detect and Avoid Solutions

Tables 80.4 and 80.5 provide an assessment of the 5 DAA conceptual solutions (in columns) against the identified UAS requirements (in rows). The table is filled using the information available and NLR expert judgment. In Table 80.4 the situation is considered where ATC is responsible for separation provision and the DUO/UAS for collision avoidance. In Table 80.5 the situation is considered where the DUO/UAS is responsible for separation provision and collision avoidance. For each conceptual solution, the feasibility of compliance with the requirement is discussed; to what extent is the DAA conceptual solution compliant with the UAS requirement? Specific issues that are still in the way of compliance are indicated.

Using the full noncooperative concept with Radar and EO/IR sensors, it is harder to detect cooperative traffic than when using cooperative concepts. Using only a cooperative system is not feasible however, because there is always a need to detect noncooperative traffic. Even if transponders are mandated, transponder system failures are possible, resulting in de facto noncooperative traffic. The most feasible solution in the near future is combining TCAS, EO/IR, and Radar. Such a system is only possible in fairly large UAS where accommodation is not an issue. To reliably detect noncooperative traffic, high performance is required from sensors. Replacement of the cooperative surveillance functionality of TCAS by ADS-B (as in mixed concept III) is not foreseen in the near future. No practical DAA solutions are fully compliant with the requirement for independence between collision avoidance and separation assurance.

Table 80.4 Assessment of five DAA sensor suites. Case: ATC is responsible for separation

| 1. Full non-cooperative concept | | | | |
|--|---|---|--|---|
| 2. Full cooperative concept | | | | |
| 3. Mixed concept 1 | | | | |
| 4. Mixed concept 2 | | | | |
| 5. Mixed concept 3 | | | | |
| | Radar, EO/IR | TCAS, ADS-B | TCAS, EO/IR | ADS-B, EO/IR, Radar |
| Generic Requirements | | | | |
| GR01: field of regard (azimuth, elevation) | | | | |
| Note: assessed for noncooperative (NC) and cooperative (C) traffic separately. Cooperative traffic is assessed considering the means available to exchange state information | | | | |
| + = Large - = Small 0 = None | NC: +/- | NC: 0 | NC: - | NC: +/- |
| | C: 0 | C: + | C: + | C: + |
| | Both EO/IR and Radar suffer more from masking by aircraft structure when increased field of regard is required | TCAS and ADS-B have a full sphere field of view; reception does not suffer from masking by aircraft structure | EO/IR suffer from masking by aircraft structure; TCAS for cooperative traffic only | EO/IR and Radar suffers from masking by aircraft structure; compensated by TCAS for cooperative traffic |
| GR02: detection range | | | | |
| Note: assessed for noncooperative (NC) and cooperative (C) traffic separately. Cooperative traffic is assessed considering the means available to exchange state information | | | | |
| + = Long range - = Short range 0 = None | NC: +/- | NC: 0 | NC: - | NC: +/- |
| | C: +/- | C: + | C: +/- | C: + |
| | EO/IR detection range is limited and unable to detect airspace users through clouds; Radar has longer detection range | ADS-B detection range is high, and TCAS detection range is medium; both are hardly influenced by weather | TCAS detection range is medium and hardly influenced by weather | ADS-B detection range is high and hardly influenced by weather. For noncooperative traffic equal to concept 1 |

(continued)

Table 80.4 (continued)

| | 1. Full non-cooperative concept | 2. Full cooperative concept | 3. Mixed concept 1 | 4. Mixed concept 2 | 5. Mixed concept 3 |
|---|---|---|--|---|--|
| | Radar, EO/IR | TCAS, ADS-B | TCAS, EO/IR | TCAS, EO/IR, Radar | ADS-B, EO/IR, Radar |
| Generic Requirements | | | | | |
| GR03: success rate of traffic detection | +/- | - | +/- | + | + |
| + = High success rate | DAA potentially capable to detect all nearby traffic | Detection of all traffic is poor, as DAA is only capable to detect cooperative traffic | Both DAA sensors needed for detection of traffic (TCAS for distance EO/IR for accurate bearing) | DAA in potential capable to detect all nearby traffic | DAA in potential capable to detect all nearby traffic |
| - = Low success rate | capability to detect cooperative traffic is less than mixed concepts | | | | |
| GR04: rate of correctly assessed conflicts | | | | | |
| Note: the correctness of an assessment of a whether a situation is a conflict or not depends on accuracies of bearing/range estimations and tracking errors. For the rating it is assumed traffic has been detected | | | | | |
| + = High success rate | - | + | - | +/- | + |
| - = Low success rate | Radar has good range accuracy but poor bearing accuracy, partly compensated by EO/IR, depending on weather conditions | ADS-B provides accurate position of conflicting aircraft; hence tracking accuracy is high, improved by TCAS | TCAS has good range accuracy but poor bearing accuracy, partly compensated by EO/IR, depending on weather conditions | TCAS and Radar have good range accuracy but poor bearing accuracy, partly compensated by EO/IR, depending on weather conditions | ADS-B provides accurate position of conflicting aircraft; hence tracking accuracy is high, improved by Radar and EO/IR |

| | | | | | |
|---|--|--|---|--|---|
| GR05: missed conflict detection rate + Low missed conflict detection rate - High missed conflict detection rate | Note: this is a combination of GR03 (success rate of traffic detection) and GR04 (rate of correctly assessed conflicts) | | | | |
| | - | - | - | +/- | + |
| | Medium detection success rate, but low tracking performance | Poor detection success rate of all traffic | Medium detection success rate, but low tracking performance | High detection success rate and capable of accurate tracking | High detection success rate and well capable of accurate tracking |
| Eurocontrol Requirements | | | | | |
| UAV04: comply with the right-of-way rules | Derived requirement: bearing of intruders must be determined | | | | |
| | Feasible | Feasible | Feasible | Feasible | Feasible |
| UAV06: VFR flight: system to assist DUO in separation provision. An automatic system should provide collision avoidance in the event of failure of separation provision | Note: in the assessment it is assumed that SSR transponders are mandated, but may fail during flight, therefore collision avoidance with noncooperative traffic must be included | | | | |
| | Challenging The challenge is to detect all nearby traffic with noncooperative sensors only. The performance of the sensors must be very high to estimate distance, bearing, and altitude difference with the intruder aircraft | Not possible Detection of noncooperative traffic is not possible with cooperative sensors only | Challenging The challenge is to detect all nearby noncooperative traffic only by visual detection | Likely solution Cooperative traffic is handled by TCAS. Detection of remaining noncooperative traffic would be possible by combining Radar and EO/IR | Likely solution Cooperative traffic is handled by ADS-B as a TCAS replacement. Detection of remaining noncooperative traffic would be possible by combining Radar and EO/IR |

(continued)

Table 80.4 (continued)

| | 1. Full non-cooperative concept | 2. Full cooperative concept | 3. Mixed concept 1 | 4. Mixed concept 2 | 5. Mixed concept 3 |
|--|---|--|--|--|--|
| | Radar, EO/IR | TCAS, ADS-B | TCAS, EO/IR | TCAS, EO/IR, Radar | ADS-B, EO/IR, Radar |
| Eurocontrol Requirements | | | | | |
| UAV07: equivalent level of safety (for collision avoidance) with manned aviation | Note: quantification of equivalent level of safety (ELOS) is a challenge in itself, and there is no consensus that ELOS should be posed as a requirement for certification. Alternatively a target level of safety could be agreed upon. When ELOS is interpreted as having a type of sensor equivalent to the human eye, concept 2 fails, because it lacks an E/O sensor | | | | |
| | Collision avoidance risk containment using DAA to be shown | Collision avoidance risk containment using DAA to be shown but expected to be highly challenging | Collision avoidance risk containment using DAA to be shown | Collision avoidance risk containment using DAA to be shown | Collision avoidance risk containment using DAA to be shown |
| UAV09: independence of separation provision and collision avoidance as far as reasonably practicable | Compliance with requirement is implicitly met because in this case ATC is responsible for separation provision and the DUO for collision avoidance | | | | |
| UAV10: separation minima UAV in IFR equal to traffic with manned aircraft | Compliance with requirement is implicitly met because in this case ATC is responsible for separation provision | | | | |
| UAV11: separation minima 0.5 nm horizontal or 500 ft vertical | Compliance with requirement is implicitly met because in this case ATC is responsible for separation provision | | | | |
| UAV12: DAA compatible with ACAS (in terms of miss distances) | Compliance with this performance requirement does not only depend on the sensor performance but also on the UAS maneuverability. Focusing on the sensors only all concepts could comply with the requirement, provided the sensors have enough performance | | | | |

Table 80.5 (continued)

| 1. Full non-cooperative concept | | 2. Full cooperative concept | 3. Mixed concept 1 | 4. Mixed concept 2 | 5. Mixed concept 3 |
|---|---|--|--|---|---|
| Radar, EO/IR | | TCAS, ADS-B | TCAS, EO/IR | TCAS, EO/IR, Radar | ADS-B, EO/IR, Radar |
| Generic Requirements | | | | | |
| GR03: success rate of traffic detection + = High success rate — = Low success rate | + / — | — | + / — | + | + |
| | DAA potentially capable to detect all nearby traffic | Detection of all traffic is poor, as DAA is only capable to detect cooperative traffic | Both DAA sensors needed for detection of traffic (TCAS for distance, EO/IR for accurate bearing) | DAA in potential capable to detect all nearby traffic | DAA in potential capable to detect all nearby traffic |
| | cooperative traffic is less than mixed concepts | | | | |
| GR04: rate of correctly assessed conflicts (after traffic has been detected) + = High success rate — = Low success rate | | | | | |
| Note: the correctness of an assessment of a whether a situation is a conflict or not depends on accuracies of bearing/range estimations and tracking errors. For the rating it is assumed traffic has been detected | | | | | |
| — | — | + | — | + / — | + |
| Radar has good range accuracy but poor bearing accuracy, partly compensated by EO/IR, depending on weather conditions | Radar has good range accuracy but poor bearing accuracy, partly compensated by EO/IR, depending on weather conditions | ADS-B provides accurate position of conflicting aircraft; hence, tracking accuracy is high, improved by TCAS | TCAS has good range accuracy but poor bearing accuracy, partly compensated by EO/IR, depending on weather conditions | TCAS and Radar have good range accuracy but poor bearing accuracy, partly compensated by EO/IR, depending on weather conditions | ADS-B provides accurate position of conflicting aircraft; hence, tracking accuracy is high, improved by Radar and EO/IR |

| | | | | | |
|---|---|--|--|--|--|
| GR05: missed conflict detection rate | Note: this is a combination of GR03 (success rate of traffic detection) and GR04 (rate of correctly assessed conflicts) | | | | |
| + Low missed conflict detection rate | — | — | — | + / — | + |
| — High missed conflict detection rate | Medium detection success rate but low tracking performance | Poor detection success rate of all traffic | Medium detection success rate but low tracking performance | High detection success rate and capable of accurate tracking | High detection success rate and well capable of accurate tracking |
| Eurocontrol Requirements | | | | | |
| UAV04: comply with the right-of-way rules | Derived requirement: bearing of intruders must be determined | | | | |
| | Feasible | Feasible | Feasible | Feasible | Feasible |
| UAV06: VFR flight: system to assist DUO in separation provision. An automatic system should provide collision avoidance in the event of failure of separation provision | Challenging Detection only based on noncooperative sensors | Not feasible Detection of noncooperative traffic not possible with cooperative sensors only | Challenging Detection of noncooperative traffic based only on EO/IR | Likely solution Detection of noncooperative traffic possible with Radar and EO/IR | Likely solution Detection of noncooperative traffic possible with Radar and EO/IR |
| UAV07: equivalent level of safety with manned aviation | Note: quantification of equivalent level of safety (ELOS) is a challenge in itself, and there is no consensus that ELOS should be posed as a requirement for certification. Alternatively a target level of safety could be agreed upon. When ELOS is interpreted as having a type of sensor equivalent to the human eye, concept 2 fails, because it lacks an E/O sensor | | | | |
| | DAA risk containment to be shown | DAA risk containment to be shown but expected to be highly challenging | DAA risk containment to be shown | DAA risk containment to be shown | DAA risk containment to be shown |
| <i>(continued)</i> | | | | | |

Table 80.5 (continued)

| | | | | | |
|---|--|---|---|--|--|
| | 1. Full non-cooperative concept | 2. Full cooperative concept | 3. Mixed concept 1 | 4. Mixed concept 2 | 5. Mixed concept 3 |
| | Radar, EO/IR | TCAS, ADS-B | TCAS, EO/IR | TCAS, EO/IR, Radar | ADS-B, EO/IR, Radar |
| Eurocontrol Requirements | | | | | |
| UAV09: independence of separation provision and collision avoidance as far as reasonably practicable. In execution, they should avoid compromising each other | Requirement interpreted as sensors used for collision avoidance should not be used for separation provision. Sensor types are allocated to fulfill SA and CA functions independently | | | | |
| | Architectures possible in theory, but not practicable When two different radars are used (1 for SA function and 1 for CA function), EO/IR used to enhance each of them | Architectures possible in theory, but not practicable Not practicable, as noncooperative traffic cannot be detected. Assuming cooperative traffic only here: ADS-B could be used for SA function, and TCAS for CA functions, provided the UAV maneuverability is sufficient (i.e., compliant with ACAS II) | Not possible Both TCAS and EO/IR would be needed for CA function. This leaves no independent sensor for SA function | Possible, but without full compliance with requirement Full compliance with both TCAS and EO/IR would be needed for CA function. Radar could be used for the SA function. High sensor performance required (especially to detect difference in altitude). For best performance, all sensors would be needed for CA function, while ADS-B and Radar should also be used for the SA function | Possible, but without full compliance with requirement Full compliance with e.g., Radar and EO/IR used for CA function and ADS-B for SA function. High sensor performance required (especially to detect difference in altitude). For best performance, all sensors would be needed for CA function, while ADS-B and Radar should also be used for the SA function |

Not applicable for the case where DUO is both responsible for separation provision and collision avoidance

| | | | | |
|---|---------------------------|---------------------------|---------------------------|---------------------------|
| UAV10: separation minima UAV IFR same as for manned traffic | To be shown | To be shown | To be shown | To be shown |
| UAV11: separation minima 0.5 nm hor or 500 ft vert | Compatibility to be shown | Compatibility to be shown | Compatibility to be shown | Compatibility to be shown |
| UAV12: DAA compatible with ACAS (miss distances) | | | | |

80.5 Conclusions and Recommendations

Overall it is concluded that it is a great challenge to develop a *collision avoidance* solution for UAS with a satisfactory level of safety. It is even more difficult to develop a Detect and Avoid (DAA) system that is capable of *both collision avoidance and separation provision*. Many consider therefore collision avoidance as the first step in developing DAA solutions for UAS, restricting the operations to flights in controlled airspace with ATC providing separation at all times.

There is no consensus that the requirement for equivalent level of safety (ELOS) compared to manned aircraft is feasible to work with. The level of safety for manned aviation is difficult to adequately quantify. As an alternative quantified Target Levels of Safety (TLS) for UAS operations could be set.

There is general consensus that DAA solutions require multiple sensors to detect and avoid both cooperative and non cooperative air traffic. Therefore, in this study, five conceptual solutions combining different types of sensors are assessed against UAS. The conclusions are:

- The full noncooperative concept with Radar and electro-optical/infrared (EO/IR) sensors is flawed because cooperative traffic is harder to detect than when using cooperative concepts.
- The full cooperative concept seems not feasible because there is always a need to detect noncooperative traffic.
- DAA solutions combining a Traffic Collision Avoidance System (TCAS), EO/IR and Radar, seem to be the most feasible ones in the near future if accommodation is not an issue. However, high performance is required from sensors to reliably detect noncooperative traffic. Replacement of the cooperative surveillance functionality of TCAS by Automatic Dependent Surveillance-Broadcast (ADS-B) (as in mixed concept III) is not foreseen in the near future.
- No practical DAA solutions are fully compliant with the requirement for independence between collision avoidance and separation assurance.

The following recommendations are made:

- Existing passive and active sensors capable of detecting noncooperative traffic need to be further developed such that the performance is sufficient to reliably detect noncooperative traffic.
- DAA solutions for small UAS where accommodation is an issue need further study.
- It should be assessed if it is possible to develop a practical DAA solution that is fully compliant with the requirement for independence between collision avoidance and separation assurance.

Acknowledgments The NLR research underlying this chapter was sponsored by the FAA UAS Research Program, managed by the FAA Research and Technology Development Office, as part of a Cooperation Agreement between FAA and Civil Aviation Authorities (CAA) the Netherlands. This specific chapter was funded by the NLR Project “Knowledge for Development of a Safety

Policy for General Aviation (including UAS)” for the Dutch CAA. The support and guidance of Dr. Xiaogong Lee and Tong Vu (both FAA), Ron van de Leijgraaf (Dutch CAA), and Dr. ir. Lennaert Speijker (the project manager for the NLR research studies) was very much appreciated.

References

- Airbus, EASA certifies new “Autopilot/Flight Director” TCAS mode for A380, Airbus Press Release, 20 Aug 2009
- EASA, Opinion No 1/2008 of the European Aviation Safety Agency, Extension of the EASA system to the regulation of Air Traffic Management and Air Navigation Services (ATM/ATS), Attachment 1, Essential Requirements, Attachment 2, Regulatory Impact Assessment, 15 Apr 2008
- EDA, Technology demonstration study on sense and avoid technologies for long endurance unmanned aerial vehicles, Project overview, EDA contract 05-R&T-003, 18 Dec 2007
- EUROCONTROL, EUROCONTROL specification for the use of military unmanned vehicles as operational air traffic outside segregated airspace, released issue, 26 July 2007
- EUROCONTROL, Unmanned aircraft systems – ATM collision avoidance requirements, CND/CoE/CNS/09-156, Ed. 1.3., May 2010
- D. Hawkes, G. Mardiné, M. Allouche, Unmanned aircraft systems EUROCAE activities, EASA workshop, Paris, Feb 2008
- T. Hutchings, S. Jeffryes, S.J. Farmer, Architecting UAV Sense & Avoid systems, Institution of Engineering and Technology (IEEE) Conference on Autonomous Systems, London, 2007
- ICAO, *Rules of the Air, Annex 2*, 10th edn. (International Civil Aviation Organization, Montreal, 2005)
- T.B. Kallevig, EUROCAE WG-73 on unmanned aircraft systems, UVS International, 2011
- B. Korn, C. Edinger, UAS in civil airspace: demonstrating sense and avoid capabilities in flight trials, 27th Digital Avionics Systems Conference, St. Paul, Oct 2008
- MIDCAS, Minutes of first stakeholder workshop, D2.1.1-2, P-0057, May 2010a
- MIDCAS, MIDCAS workshop #2: towards the UAS traffic encounter CONOPS, afternoon session, Brussels, Oct 2010b
- NATO, Naval armaments group, Joint capability group on unmanned aerial vehicle systems operating in non-segregated airspace, PFP(NNAG-JC GUAV)WP(2006)0002-REV2, 13 June 2007
- NATO, Industrial Advisory Group SG-134, Sense and avoid, Main report, NIAG-D/(2010)00xx, March 2010
- M. Selier, M. Stuip, R.P.M. Verhoeven, OUTCAST NTP Final Report: Operations of UAVs – Transition to Civil Air Space and Traffic Environments, NLR Contract Report 2007–270, June 2008
- USICO, See and Avoid Technologies, D3.4/WP3400 Report, Apr 2004a
- USICO, See and Avoid Technologies, D3.5/WP3500 Report, Apr 2004b
- P. van Blyenburgh, in *UAS Unmanned Aircraft Systems: The Global Perspective*, ed. by P. van Blyenburgh (Blyenburgh, Paris, 2011), p. 151
- J.G. Verstraeten, M. Selier, Survey of European Sense and Avoid Technologies and Requirements, NLR Contract Report 2009–422, Sept 2011

Section XV

Networked UAVs and UAV Swarms

***Randal Beard and
Camille Alain Rabbath***

Kimon P. Valavanis and George J. Vachtsevanos

The unmanned aircraft military and civilian communities are already experimenting with UAV swarms flying in formation while cooperating in the execution of specific missions. Research in networked UAVs is also proceeding via simulation studies and limited experimentation. Networked UAVs and UAV Swarms addresses sensing, supervisory control, coordination and cooperation, computing, and communication issues that constitute the basic ingredients of safe and effective flight for multiple UAVs, emphasizing the “intelligence” requirements for safe flight and cooperative/supervisory control.

► **UAV Swarms: Models and Effective Interfaces** by Chapman and Mesbahi examines the modeling and design of effective control interfaces for human operators of UAV swarms. The swarm is modeled as a two-component hierarchical system consisting of the interaction dynamics among the UAVs in the swarm, referred to as the *network dynamics*, and the *UAV dynamics* itself. Human operators are assumed to be able to interface with the swarm via the network dynamics, which in turn has adopted a *leader–follower consensus model*. The system-theoretic and topological features of the network dynamics are then examined in order to design effective mechanisms for interfacing with the swarm. A performance metric is selected for reasoning about effective human swarm interaction. The role of topological features of the network is highlighted in the context of the chosen metric and is related through the effective resistance of the corresponding electrical

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

network. This is then followed by exploiting such topological features for designing a network rewiring protocol to maximize the metric. These topology design tools are applied to wind gust rejection in disturbed swarming scenarios, demonstrating the viability of topology-assisted design for improved swarm performance. A network-based model reduction is also proposed to form a lower-order model of the network which is easier for the human operators to conceptualize and manage. The reduction process involves a novel partitioning scheme, dubbed *leader partition*, in order to fuse “similar” states in the UAV network and to form a graph-theoretic method for model reduction. This model reduction technique is then applied to derive improved swarming performance in the presence of wind gusts.

► **Decentralized Multi-UAV Coalition Formation with Limited Communication Ranges** by Sujit, Manathara, Ghose, and Sousa discusses a communication protocol derived to search for potential coalition members over a dynamic UAV network and a strategy for a UAV coalition to track and destroy moving targets. The effects of communication ranges and delays as well as the scalability of the proposed schemes with change in number of targets and UAVs are studied. The results show that it is beneficial to search for potential coalition members over a wide diameter in the UAV network when the communication delay is less, while it is more advantageous to determine a coalition from among the immediate neighbors in the presence of significant communication delay.

► **Coordinated Convoy Protection Among Teams of Unmanned Aerial Vehicles** by Egerstedt, Rahmani, and Young investigates how to control and coordinate teams of UAVs to provide protection to convoys of ground vehicles. This is a rich yet canonical problem when coordinating multiple UAVs in that coordinated movements, task assignments, and resource balancing must all be performed for a successful completion of the mission. Time optimal paths for providing convoy protection to stationary ground vehicles are presented, and these algorithms are extended to moving ground vehicles. The assignment problems, associated with dispatching UAVs from the convoy to inspect and clear potential threats, are also discussed.

► **UAV Routing and Coordination in Stochastic, Dynamic Environments** by Enright, Frazzoli, Pavone, and Savla focuses on a technical approach that relies upon methods from queuing theory, combinatorial optimization, and stochastic geometry. The main advantage of this approach is its ability to provide analytical estimates of the UAV system performance on a given problem, thus, providing insight into how performance is affected by design and environmental parameters, such as the number of UAVs and the target distribution. This approach also provides provable guarantees on the system’s performance with respect to an ideal optimum. To illustrate this approach, a variety of scenarios are considered, ranging from the simplest case where one UAV moves along continuous paths and has unlimited sensing capabilities, to the case where the motion of the UAV is subject to curvature constraints, and finally to the case where the UAV has a finite sensor footprint. Finally, the problem of cooperative routing algorithms for multiple

UAVs is considered, within the same queuing-theoretical framework, and with a focus on control decentralization.

Collectively, this section introduces the reader to requirements of swarms of UAVs operating in unison to complete complex missions and also paves the way for future developments in unmanned–manned swarms.

Airlie Chapman and Mehran Mesbahi

Contents

| | | |
|------------|---|------|
| 82.1 | Introduction | 1988 |
| 82.2 | Model | 1992 |
| 82.2.1 | Two-Component Hierarchical Approach | 1992 |
| 82.2.2 | Network Topology | 1993 |
| 82.2.3 | Network Dynamics | 1995 |
| 82.3 | Effective Interfaces Using the Open Loop \mathcal{H}_2 Norm | 1997 |
| 82.3.1 | Common Signal Operator | 1999 |
| 82.3.2 | Independent Signal Operator | 2000 |
| 82.3.3 | Generic Signal Operator | 2001 |
| 82.4 | Topology Design | 2004 |
| 82.4.1 | Constraint Graph | 2004 |
| 82.4.2 | Network Design Protocol | 2005 |
| 82.4.3 | Topology Design for Wind-Gust Disturbance Rejection | 2007 |
| 82.5 | Model Reduction | 2012 |
| 82.5.1 | Model Truncation | 2012 |
| 82.5.2 | Model Reduction Using Equitable Distance Partitions | 2013 |
| 82.5.3 | Model Reduction for Wind-Gust Disturbance Rejection | 2016 |
| References | | 2017 |

Abstract

This chapter examines modeling and the design of effective control interfaces, for human operators, of unmanned aerial vehicle (UAV) swarms. The swarm is modeled as a two-component hierarchical system consisting of the interaction dynamics among the UAVs in the swarm, referred to as the *network dynamics*, and the *UAV dynamics* itself. Human operators are assumed to be able to interface with the swarm via the network dynamics, which in turn has adopted a *leader-follower consensus model*. The system-theoretic and topological features of the

A. Chapman (✉) • M. Mesbahi

Aeronautics and Astronautics, University of Washington, Seattle, WA, USA

e-mail: mesbahi@aa.washington.edu; airliec@u.washington.edu

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,
DOI 10.1007/978-90-481-9707-1_34,

© Springer Science+Business Media Dordrecht 2015

1987

network dynamics are then examined in order to design effective mechanisms for interfacing with the swarm. Specifically, the open loop \mathcal{H}_2 norm of the network is selected as a performance metric for reasoning about effective human-swarm interaction. The role of topological features of the network is highlighted in the context of this metric and is related through the effective resistance of the corresponding electrical network. This is then followed by exploiting such topological features for designing a network rewiring protocol to maximize the \mathcal{H}_2 norm. These topology design tools are applied to wind-gust rejection in disturbed swarming scenarios, demonstrating the viability of topology-assisted design for improved swarm performance. A network-based model reduction is also proposed to form a lower-order model of the network which is easier for the human operators to conceptualize and manage. The reduction process involves a novel partitioning scheme, dubbed *leader partition*, in order to fuse “similar” states in the UAV network and to form a graph-theoretic method for model reduction. This model reduction technique is then applied to derive improved swarming performance in the presence of wind gusts.

82.1 Introduction

Manageability of unmanned aerial vehicle (UAV) swarms presents new challenges at the intersection of interdisciplinary fields. The first challenge pertains to advances in designing small UAVs and micro aerial vehicles (MAVs) involved in complex mission scenarios (Mueller 2001; Pines and Bohorquez 2006; Shyy et al. 2007; Ulrich et al. 2010). The reduced physical scale of such vehicles limits their sensing and actuation capabilities, often requiring their cooperative operation for successful missions. The second challenge relates to the improved autonomy of individual and multiple UAVs, removing the need of a single operator for each UAV (Griffiths et al. 2007; Ren and Beard 2007; How et al. 2008; Michael et al. 2010; Mellinger and Kumar 2011).

In this chapter, components of the UAV swarm are abstracted away to reason about creating effective interfaces with a swarm of interacting UAVs. These interactions, forming a *UAV interaction network*, consist of inter-UAV wireless communications or relative sensor measurements. The UAV swarm involves a large number of dynamic vehicles, making it infeasible for human operators to interface directly with every UAV. An approach for creating effective interfaces is to have only a subset of the UAVs directly controlled by ground or airborne human operators. The remaining UAVs are then allowed to be indirectly controlled via the UAV interaction network, depicted in Fig. 82.1. This setup, although more advantageous than the scenario where each operator manage one UAV, results in a highly coupled high-order nonlinear control model, that proves to be difficult for human operators to manage.

The approach advocated in this chapter for creating effective human-swarm interaction is via a two-component hierarchical model, consisting of the vehicle dynamics building on the UAV interaction network dynamics. This in turn leads to a methodology for improving operators’ ability to manage the swarm, based



Fig. 82.1 A UAV swarming concept where only a subset of the UAVs are directly controlled by human operators. These operators influence the remaining UAVs through an interaction network between vehicles

on the topological features of the swarm interaction network. The main tool for investigating the effectiveness of the operator's interface with the swarm is the open loop \mathcal{H}_2 norm of the resulting networked system – used in this chapter as a measure of system manageability. In fact, when the network dynamics is diffusive, the open loop \mathcal{H}_2 norm provides a means of relating topological features of the network that promote manageability of the swarm. In situations where the UAV network is particularly poor for effective operator interfaces, an option is to redesign the network online – made possible by the wireless nature of the interactions. In particular, the open loop \mathcal{H}_2 norm for the network can be employed to dynamically rewire the interaction topology, improving the effectiveness of the operators' interface with the swarm.

This chapter focuses on UAV interaction networks that have adopted a diffusive protocol, specifically leader-follower dynamics based on the *consensus protocol*. An attraction of the consensus protocol is that a group of UAVs can manage, in a distributed way, to converge to agreement, for example, a common altitude or a formation. More generally, consensus provides a framework for simple but effective distributed information sharing and control for networked, multi-agent systems in settings such as multi-vehicle control, formation control, swarming, and distributed estimation; see, for example, Bullo et al. (2009) and Mesbahi and Egerstedt (2010). One of the popular adaptations of traditional consensus is *leader-follower consensus dynamics* (Tanner et al. 2004; Mesbahi and Egerstedt 2010), in which controllers (leaders), that do not conform to traditional consensus, can act as traditional agents to impact the network behavior by exploiting the other agents' (followers') consensus dynamics. The relationship between the network topology

and the effectiveness of the controllers' input in a leader-follower consensus system is explored in this chapter. Topological features of the network are then investigated that vary the controllers' impact, measured in terms of the open loop \mathcal{H}_2 norm of the network dynamics. Favorable network topologies are then designed, via a decentralized rewiring protocol, to exploit the aforementioned topological features.

An attraction of networked systems is that system performance is strongly coupled with the underlying network topology. For traditional consensus, system performance and its ties to network topology is a well-researched problem where the second smallest eigenvalue of the graph Laplacian is a judicious metric to quantify the convergence rate (Olfati-Saber 2006). Interest has also been shown for other network measures, for example, the largest eigenvalue of the graph Laplacian (Li and Cao 2009). However, these metrics prove less attractive in leader-follower consensus where convergence rates can vary dramatically depending on where the leaders are located within the network. For the leader-follower consensus, the selected performance metric in this chapter is the open loop \mathcal{H}_2 norm. The system-theoretic interpretation of this metric provides a tangible link to system performance in the presence of leaders. Further, the equivalent circuit representation of the network, described in Bollobás (1998) and Barooah and Hespanha (2006), provides the representation of the open loop \mathcal{H}_2 norm as the total *effective resistance* of the network.

The open loop \mathcal{H}_2 norm is studied when the manageability of the system is a concern. Increasing the open loop \mathcal{H}_2 norm tends to increase the receptiveness of the system to control. Work in this area has focused on allocating the weights on the interaction links between UAVs (Ghosh et al. 2008). This chapter examines the structure of the interaction topology that exhibits a favorable open loop \mathcal{H}_2 norm. Unfortunately, maximizing this metric via topological means is NP-hard (a generally considered intractable problem), and thus, a suboptimal protocol to improve this metric is required. The proposed approach is to perform edge trades among neighboring agents within the network that tend to increase the total effective resistance of the graph and consequently the open loop \mathcal{H}_2 norm.

Designing and adapting topologies to optimize for certain metrics has been addressed by several authors: Ghosh and Boyd (2006) aimed to maximize the second smallest eigenvalue of the graph Laplacian, Zelazo and Mesbahi (2011) examined the minimization of the network's \mathcal{H}_2 performance when noise is applied to the interaction links in the network, and Wan et al. (2007) considered maximizing the largest eigenvalue of the graph Laplacian. All aforementioned authors used centralized optimization techniques over *weighted* graphs. Kim and Mesbahi (2006) used fading functions to approximately represent the on/off linkage relationship when searching for the maximum second smallest eigenvalue of the graph Laplacian to increase the convergence properties of the network dynamics. Wu and Wang (2009) have approached the same problem using genetic algorithms. Intuitive methods for network reconfiguration have been designed to improve network resilience, for example, using thresholding methods to decide when to alter the topology (Tyson et al. 2010).

One of the challenges in reducing the size of a UAV is its increased susceptibility to external disturbances, such as wind gusts. An example is provided that examines

the wind-gust swarming scenario which involves a swarm of UAVs with the objective of maintaining a constant altitude aided by relative sensing between neighboring UAVs. An \mathcal{H}_2 controller is implemented by operators to reject the effect of wind-gust disturbance on the overall network. As the swarm is cooperative, favorable controller disturbance corrections can be inadvertently dampened via coordination between UAVs. This chapter presents techniques to manipulate the UAV coordination network topology to improve the swarm's resilience to wind gusts. A network topology-based model reduction method is further proposed for the network dynamics that takes advantage of symmetries or near-symmetries in the network, with the goal of reducing the order of the system, requiring lower-order controllers and reducing monitoring overhead.

The effect of wind gusts on aircraft is a well-researched area (Mutuel and Douglas 1997; Stevens and Lewis 2003). The single and collective response of micro aerial vehicle (MAV) swarms to wind gusts (Hall 2007), on the other hand, is an emerging research field. Shyy et al. (2008) have investigated a bio-inspired MAV design to combat environmental disturbances. A myriad of controllers have been proposed for such a task; for example, \mathcal{H}_2 and \mathcal{H}_∞ controllers have been proposed in Cheviron (2006) and Yang et al. (2006). One of the few papers that have studied a gust-exposed network-based UAV swarm is Meskin et al. (2010) who proposed a hybrid fault-detection and isolation approach involving thresholding between small tolerable and large disturbances in the system, *without* explicit attention to the coordination topology of the UAV swarm. The approach in this chapter aims instead to maintain the underlying consensus dynamics and favorably alter the coordination topology.

Another facet presented in this chapter is motivated by the observation that large-scale UAV swarms are difficult for humans to conceptualize and manage. A traditional system-theoretic solution for reducing the complexity of large-scale systems is the formulation of lower-order approximate models using model reduction techniques. For the UAV swarm, this leads to obscuring the UAV network topology in the reduced order model. A network-centric model reduction technique that produces an approximate lower-order model is proposed, preserving certain characteristics of the network topology.

Traditional methods of model reduction such as balanced realization, balanced residualization, and Hankel norm approximations (Dullerud and Paganini 2000) are purely system-based with minimal, if any, network interpretation. The significance of network symmetry and its role in controllability has recently been investigated (Martini et al. 2008; Rahmani et al. 2009) and provides a natural extension of system-based to network topology-based model reduction. Preliminary work in this direction was undertaken by Martini et al. (2008) using single-leader leader-follower consensus systems. Extending this to multi-leader scenarios is presented in the chapter.

This chapter is organized as follows. Section 8.2 contains the problem formulation and relevant background. The proposed modeling approach consists of a two-component hierarchical model, separating the role of the network dynamics from the vehicle dynamics. Consensus-based, leader-follower dynamics is described for the network dynamics. An analysis of the open loop \mathcal{H}_2 norm is then presented in

Sect. 82.4, and its relationship to system performance and the effective resistance of the network is established. Also in Sect. 82.4, a network rewiring method is proposed to design network topologies optimized for the open loop \mathcal{H}_2 norm and conducive to wind-gust disturbance rejection. Section 82.5 presents a network partitioning technique that can be used for fusing nodes to form a reduced order model but without losing the graph-theoretic interpretation of the dynamics. The chapter is concluded with a few remarks in Sect. 82.6.

82.2 Model

This section provides the models that will be used in this chapter, including abbreviated descriptions of graphs and the consensus protocol in both its traditional and leader-follower versions. The following notation is used in this chapter: $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ denote the one, Euclidean and infinity norms, respectively; $\text{tr}(\cdot)$ denotes the trace of a matrix; $|\cdot|$ denotes the cardinality of a set; $\mathbf{1}$ denotes the column vector of ones; and \otimes denotes the Kronecker product. The spectrum of a symmetric matrix $M \in \mathbb{R}^{n \times n}$ is denoted and ordered as $\lambda_1(M) \leq \lambda_2(M) \leq \dots \leq \lambda_n(M)$. For matrices $M, N \in \mathbb{R}^{n \times n}$, $N \preceq M$ denotes that $M - N$ is positive semi-definite.

82.2.1 Two-Component Hierarchical Approach

A common approach to modeling large-scale multi-agent systems is to decompose the system into a hierarchy of manageable components. This process decouples a multi-agent system into a network component which interfaces with the individual system agents. Many applications adopt such an approach, for example, to design inner- and outer-loop systems and to decouple longitudinal and lateral modes in aircraft dynamics (Schmidt 1998; Stevens and Lewis 2003).

In a two-component hierarchical structure, the network dynamics drives the vehicle dynamics and in turn is influenced by the human operators. A benefit of this approach is that if there exists a weak coupling between components, then complicated systems can be reduced to simpler decoupled systems, with only small performance degradation. Also, existing theory regarding one component can be applied to other models with only minimal changes to that component.

For the specific case of a UAV swarm, the swarm is decomposed into an inter-agent network dynamics component (*network dynamics*) and a closed loop vehicle dynamics component (*vehicle dynamics*). The vehicle dynamics encapsulates the individual characteristics of the platform such as a six degrees-of-freedom UAV dynamic model. The study of vehicle dynamics is a well-researched area, while the study of network dynamics is relatively new. Examples of network infrastructure through which the network dynamics act include communication links between agents, range sensors to measure interagent distance and relative-agent velocities, and cameras for interagent relative bearing measurements. A set of human operators may be considered as part of the network in this decomposition. The operators may

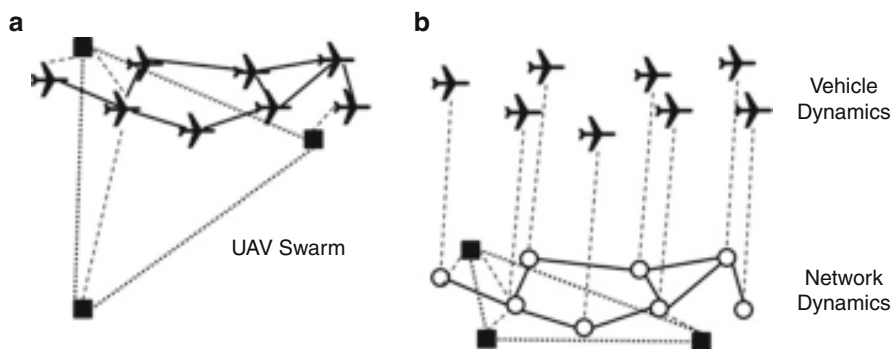


Fig. 82.2 (a) Original UAV swarm. The UAV swarms' communication links are marked with *solid lines*. The *squares* denote airborne and land-bound human operators, and the *dotted lines* represent their intercommunication links. The *dashed lines* denote the human operators, communication links to the swarm. (b) The two-component hierarchical design approach

communicate from ground stations and nearby aerial vehicle or directly control all or a subset of UAVs.

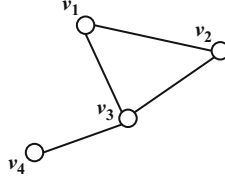
This hierarchical approach is graphically represented in Fig. 82.2. The operators' signal drives the network dynamics. The network dynamics is a representation of the interagent interaction that runs a simplified version of the UAV dynamics, such as a particle or unicycle model. The states of the network dynamics are then used for output tracking by the vehicle dynamics. Additional decoupling can be inserted into this hierarchy, such as a reduced order network component, explored in Sect. 82.5.

Care must be taken to place necessary constraints on the dynamics to account for coupling that has been removed. For example, if the network is based on Euclidean distances, the topology of the network must be maintained throughout the trajectory of the vehicle, or alternatively the network must be time varying, based on the states of the vehicle dynamics. Another case, addressed in Sect. 82.4, is that perturbations of the vehicle dynamics should have equivalent perturbations accounted for in the design of controllers for the network component.

This decoupling of the network dynamics serves to isolate the role of the interaction dynamics from the vehicle dynamics. The benefit is that system-theoretic features of the network can be formed and applied to many different vehicle models. The network can then be considered as a design tool in forming a more manageable UAV swarm. The role of the network will be the main focus of this chapter including system's attributes like the controllability Gramian, as well as design attributes like network topology, discussed in Sect. 82.4.

82.2.2 Network Topology

To represent the network topology, the communication points in the network are denoted as nodes, and edges are communication links between points.



$$\mathcal{A}(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \Delta(\mathcal{G}) = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathcal{L}(\mathcal{G}) = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Fig. 82.3 A sample network graph $\mathcal{G} = (V, E)$ where $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_3, v_4\}\}$. The adjacency matrix $\mathcal{A}(\mathcal{G})$, degree matrix $\Delta(\mathcal{G})$, and Laplacian matrix $\mathcal{L}(\mathcal{G})$ of the graph are provided

For UAV swarms, the nodes represent UAVs, and the edges correspond to inter-UAVs links, such as wireless communications or relative sensors. It is assumed that all communication links are bidirectional. These nodes and edges can therefore be considered as forming an undirected graph.

Abstractly, an undirected graph $\mathcal{G} = (V, E)$ is defined by a node set V with cardinality n , the number of nodes in the graph, and an edge set E comprised of pairs of nodes, where nodes v_i and v_j are adjacent if $\{v_i, v_j\} \in E \subseteq [V]^2$, the set of two-element subsets of V . One special family of undirected graphs are *tree graphs*, denoted by the set \mathcal{T} , where all two-node pairs are connected by exactly one simple path, that is, a connected graph without cycles. A *spanning tree* of a graph is a tree subgraph that connects all vertices in the graph.

The *neighborhood set* $\mathcal{N}(v_i)$ is composed of the set of nodes adjacent to v_i . The scalar $d(v_i, v_j)$ is the minimum path length, induced by the graph \mathcal{G} , between nodes v_i and v_j . The degree ∂_i of node v_i is the number of its adjacent nodes. The *degree matrix* $\Delta(\mathcal{G}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with ∂_i at entry (i, i) .

There are many matrix representations of a graph; the two most popular are the adjacency matrix and graph Laplacian matrix that each fully characterize the graph (Bollobás 1998; Diestel 2000). The *adjacency matrix* is a $n \times n$ symmetric matrix with $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ when $\{v_i, v_j\} \in E$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The *graph Laplacian matrix* is defined as $L(\mathcal{G}) = \Delta(\mathcal{G}) - \mathcal{A}(\mathcal{G}) \in \mathbb{R}^{n \times n}$. Figure 82.3 shows an example graph with its corresponding graph matrices.

The graph Laplacian matrix plays an important role in the dynamics of the network. An important feature of this matrix is that it is a (symmetric) positive semi-definite matrix. The spectrum is ordered as $0 = \lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \dots \leq \lambda_n(\mathcal{G})$, where, for brevity, $\lambda_i(\mathcal{G})$ is used instead of $\lambda_i(L(\mathcal{G}))$. The spectrum of the graph Laplacian matrix displays many features of the network. For example, the number of zero eigenvalues is equal to the number of connected components. Other features that couple the network topology and dynamics are explored in the following subsection.

82.2.3 Network Dynamics

A common objective for UAV swarms is to reach agreement on one or more of their states. For example, agreement on UAV position achieves rendezvous, or, if agreeing on a virtual position, formation flight can be acquired with known position offsets from the virtual position. Velocity agreement is another attractive property for formation flight. For distributed surveillance, bearing agreement is often desired.

If agreement is required over a network of UAVs without all-to-all communications, a distributed approach is necessary. A protocol which is particularly adept at distributed agreement is the consensus protocol, which is detailed below, and runs within the network dynamics of the UAV swarm.

82.2.3.1 Consensus Dynamics

Consider $x_i(t) \in \mathbb{R}$ to be the i -th node's state at time t on which agreement is required for all nodes. The continuous-time consensus dynamics is defined over the graph $\mathcal{G} = (V, E)$ as

$$\dot{x}_i(t) = \sum_{\{v_i, v_j\} \in E} (x_j(t) - x_i(t)). \quad (82.1)$$

Thus, to update the i -th node's state, only the relative state of node i 's neighbor's state is required. In a compact form with $x(t) \in \mathbb{R}^n$, the collective dynamics is represented as

$$\dot{x}(t) = -L(\mathcal{G})x(t),$$

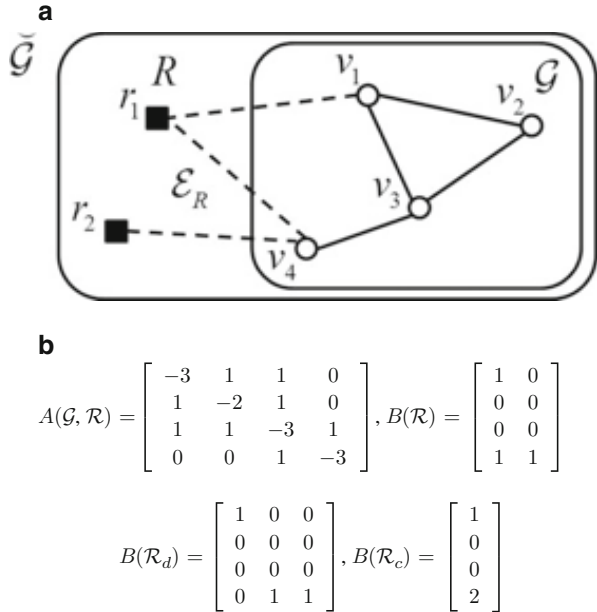
with $L(\mathcal{G})$ being the graph Laplacian matrix of the underlying interaction topology, described in the previous subsection. For a connected graph \mathcal{G} , the network dynamics will converge to an agreement on the state, that is, $x_1(t) = x_2(t) = \dots = x_n(t) = \alpha$, for some constant α , for all initial conditions (Olfati-Saber et al. 2007). Further, the slowest convergence of the dynamics is determined by $\lambda_2(\mathcal{G})$ which is a measure of graph connectivity.

82.2.3.2 Leader-Follower Consensus Dynamics

One of the advantages of the consensus dynamics is that the human operators can seamlessly integrate into the dynamics by directly applying an appropriately designed signal to the interface link. As human operators do not follow (82.1), the operator-influenced consensus is dubbed *leader-follower consensus*. The operators are considered as leaders, and the UAVs as followers.

To form the leader-follower consensus dynamics, the network graph $\mathcal{G} = (V, E)$ is extended to incorporate operator nodes into the graph. This is accomplished by considering the operator pair $\mathcal{R} = (R, \mathcal{E}_R)$, where R is the r element operator node set and $\mathcal{E}_R \subseteq R \times V$ is the set of edges used by the operators to inject signals into the network. It is assumed that for an operator $r_j \in R$, the same signal $u_j(t) \in \mathbb{R}$ is delivered along every edge adjacent to it. A new graph $\check{\mathcal{G}} = (\check{V}, \check{E})$ is thus formed with $\check{V} = V \cup R$ and $\check{E} = E \cup \mathcal{E}_R$. Any communication links between human

Fig. 82.4 (a) The network nodes V with coordination edges E form a graph \mathcal{G} . The human operator in the set R is connected to the rest of the network via the communication links corresponding to the edge set \mathcal{E}_R . The complete operator-network node graph containing all agents $\tilde{V} = V \cup R$ and edges $\tilde{E} = E \cup \mathcal{E}_R$ is denoted as $\tilde{\mathcal{G}}$. (b) The operators r_1 and r_2 communicate to UAVs v_1 and v_4 , leading to an altered graph Laplacian matrix $A(\mathcal{G}, \mathcal{R})$ and the input matrix $B(\mathcal{R})$ of model (82.3). The control matrices relating to the special cases of distinct and common control



operators are removed. This assumption is justified as the operators are assumed to be working cooperatively and do not require coupled dynamics to the other operators' states. Figure 82.4 provides a graphical representation of this notation and setup.

The resulting leader-follower consensus system now assumes the form

$$\dot{x}_i(t) = \sum_{\{v_i, v_j\} \in E} (x_j(t) - x_i(t)) + \sum_{\{v_i, r_j\} \in \mathcal{E}_R} (u_j(t) - x_i(t)) \quad (82.2)$$

with the full dynamics

$$\dot{x}(t) = A(\mathcal{G}, \mathcal{R})x(t) + B(\mathcal{R})u(t), \quad (82.3)$$

where $B(\mathcal{R}) \in \mathbb{R}^{n \times r}$ with $[B(\mathcal{R})]_{ij} = 1$ when $\{r_j, v_i\} \in \mathcal{E}_R$ and $[B(\mathcal{R})]_{ij} = 0$ otherwise, and

$$A(\mathcal{G}, \mathcal{R}) = -(L(\mathcal{G}) + D(\mathcal{R})) \in \mathbb{R}^{n \times n}, \quad (82.4)$$

where $D(\mathcal{R}) \in \mathbb{R}^{n \times n}$ with $[D(\mathcal{R})]_{ii} = \delta_i^r$, where δ_i^r is the number of operators adjacent to v_i , and $[D(\mathcal{R})]_{ij} = 0$ otherwise. The scalar δ_j^v is defined as the number of network nodes adjacent to r_j . There are two special operator interfaces examined in this chapter: one in which there is exactly one operator for each edge \mathcal{E}_R , and so an independent (or *distinct*) control signal is delivered through each edge, denoted with the operator pair \mathcal{R}_d , and one where there exists only one operator, so a *common*

signal is delivered through each edge of \mathcal{E}_R , denoted with pair \mathcal{R}_c . A sample model and its system matrices as well as the special cases of \mathcal{R}_d and \mathcal{R}_c are depicted in Fig. 82.4. The set $\pi(\mathcal{E}_R)$ is composed of all nodes $v_i \in V$ such that $\{r_j, v_i\} \in \mathcal{E}_R$. In other words, $\pi(\mathcal{E}_R)$ is simply the set of agents that directly connect to an operator

The matrix $A(\mathcal{G}, \mathcal{R})$ in (82.4) is the *Dirichlet matrix*, or grounded Laplacian (Barooah and Hespanha 2006; Salsa 2008). The spectrum of $A(\mathcal{G}, \mathcal{R})$ relates closely to the spectrum of $L(\mathcal{G})$. In this way, the topology of the underlying graph is related to the dynamics of model (82.3). An auxiliary observation about the Dirichlet matrix, to be used subsequently, is the following.

Proposition 1 (Chapman and Mesbahi 2010) *The matrix $A(\mathcal{G}, \mathcal{R})$ of model (82.3) is negative definite and so invertible, if the original graph is connected.*

In the following discussion, only connected graphs are considered as each component of a disconnected graph can be analyzed separately.

82.3 Effective Interfaces Using the Open Loop \mathcal{H}_2 Norm

The two-component hierarchical model approach separates factors that contribute to the formation of manageable UAV swarms. The first is the familiar vehicle dynamics controllability, where the hardware and software design of individual UAVs affects the vehicles' receptiveness to operator commands. The second is network dynamics controllability which is unique to multi-vehicle platforms. Network dynamics controllability is a function of the network protocol and the underlying interagent connections, which can be represented by a graph as described in the previous section. The following will be focusing on network-theoretic aspects to controllability. The results can be used to select effective interfaces (attachment points) for the operators to interact with the UAV swarm.

The main tool of analysis is the open loop \mathcal{H}_2 norm of the system. For a linear system, with inputs, outputs and internal states, $u(t)$, $y(t)$ and $x(t)$, respectively, of the form

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t),\end{aligned}$$

the open loop \mathcal{H}_2 norm is defined in terms of the controllability Gramian,

$$P_T(A, B) := \int_0^T e^{A\tau} B B^T e^{A^T \tau} d\tau,$$

for the system. Specifically, the open loop \mathcal{H}_2 norm is

$$\|G(s)\|_2 = \sqrt{\text{tr}(C P_\infty(A, B) C^T)}, \quad (82.5)$$

where the state-space realization is $G(s) = C(sI - A)^{-1}B$. The open loop \mathcal{H}_2 norm serves as an attractive metric of the manageability of a dynamic system. This is evident from its relationship to the controllability Gramian, as well as the relationship to the state-space realization describing the mapping of inputs to outputs, that is, $y(s) = G(s)u(s)$. Another relationship is to the energy amplification of the system. The energy of the states at the system output, from a unit impulse input $u(t)$ when $x(0) = 0$, is

$$\int_0^\infty y(t)^T y(t) dt = \|G(s)\|_2^2.$$

From these observations, in general, inputs perturb the outputs more effectively as $\|G(s)\|_2$ increases. In other words, systems with a larger open loop \mathcal{H}_2 norm tend to be more manageable.

For the leader-follower consensus model introduced in Sect. 8.2.3.2, the operators are assumed to be able to monitor all of the UAVs in the swarm. This implies that $C = I$, and the open loop \mathcal{H}_2 norm reduces to

$$\|G(s)\|_2 = \sqrt{\text{tr}(P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R})))}.$$

If $A(\mathcal{G}, \mathcal{R})$ is negative definite, then $\|G(s)\|_2$ is bounded. From Proposition 1, this requirement is met when \mathcal{G} is connected. For a specific model, the open loop \mathcal{H}_2 norm is henceforth denoted as $\|G_{\mathcal{G}, \mathcal{R}}(s)\|_2$, as it is solely dependent on \mathcal{G} and \mathcal{R} . Similarly, $P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R}))$ is denoted as $P(\mathcal{G}, \mathcal{R})$. For UAV swarming applications, the open loop \mathcal{H}_2 norm quantifies the network's amplification of the operators' inputs.

Directly, for connected graph \mathcal{G} , from the definition of the controllability Gramian,

$$\begin{aligned} \|G_{\mathcal{G}, \mathcal{R}}(s)\|_2^2 &= \text{tr} \left(\int_0^\infty e^{A(\mathcal{G}, \mathcal{R})\tau} B(\mathcal{R}) B(\mathcal{R})^T e^{A(\mathcal{G}, \mathcal{R})^T \tau} d\tau \right) \\ &= \text{tr} \left(\int_0^\infty B(\mathcal{R}) B(\mathcal{R})^T e^{A(\mathcal{G}, \mathcal{R})^T \tau} e^{A(\mathcal{G}, \mathcal{R})\tau} d\tau \right) \\ &= \text{tr} \left(B(\mathcal{R}) B(\mathcal{R})^T \int_0^\infty e^{2A(\mathcal{G}, \mathcal{R})\tau} d\tau \right) \\ &= -\frac{1}{2} \text{tr} (M(\mathcal{R}) A(\mathcal{G}, \mathcal{R})^{-1}), \end{aligned} \tag{82.6}$$

where $M(\mathcal{R}) = B(\mathcal{R}) B(\mathcal{R})^T$.

82.3.1 Common Signal Operator

Equation (82.6) provides network-theoretic properties of the open loop \mathcal{H}_2 norm for the special case introduced in Sect. 82.2.3.2, where the same signal is delivered along every edge in \mathcal{E}_R with operator pair \mathcal{R}_c . Here, either only one operator applies a signal or all operators apply the same signal. The following proposition states that the open loop \mathcal{H}_2 norm only depends on the number of communication links between the operators and the UAV nodes, $|\mathcal{E}_R|$.

Proposition 2 *For a connected graph \mathcal{G} where all operators apply the same signal,*

$$\|G_{\mathcal{G}, \mathcal{R}_c}(s)\|_2^2 = \frac{1}{2} |\mathcal{E}_R|.$$

Proof. The eigenvalue $\lambda_1(\mathcal{G}) = 0$ is associated with the left eigenvector $\mathbf{1}$ and $B(\mathcal{R}_c) \in \mathbb{R}^{n \times 1}$ where $B(\mathcal{R}_c)_i = [D(\mathcal{R}_c)]_{ii}$, for $i = 1, \dots, n$. Then,

$$\begin{aligned} -\mathbf{1}^T A(\mathcal{G}, \mathcal{R}_c) &= \mathbf{1}^T (L(\mathcal{G}) + D(\mathcal{R}_c)) \\ &= \mathbf{1}^T D(\mathcal{R}_c) \\ &= B(\mathcal{R}_c)^T. \end{aligned}$$

From Proposition (82.6),

$$\begin{aligned} \|G_{\mathcal{G}, \mathcal{R}_c}(s)\|_2^2 &= -\frac{1}{2} \text{tr}(B(\mathcal{R}_c)B(\mathcal{R}_c)^T A(\mathcal{G}, \mathcal{R}_c)^{-1}) \\ &= \frac{1}{2} \text{tr}(B(\mathcal{R}_c)\mathbf{1}^T) \\ &= \frac{1}{2} |\mathcal{E}_R|. \end{aligned}$$

■

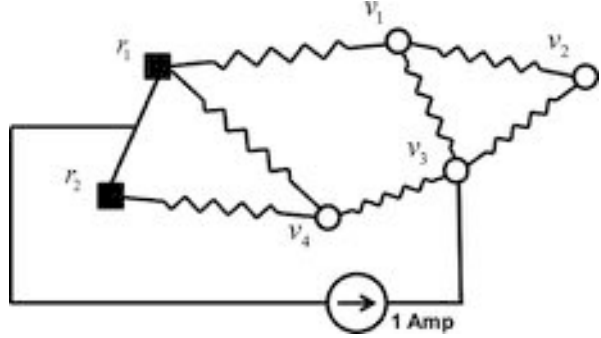
In some UAV swarming scenarios, an operator may have a group of UAVs to control directly. Proposition 2 provides the best selection of UAVs from the swarm based on the network topology.

Corollary 3 *For a graph $\check{\mathcal{G}}$ and model (82.3) with one operator,*

$$\|G_{\check{\mathcal{G}}, \mathcal{R}}(s)\|_2^2 = \frac{1}{2} \delta(r_1),$$

where $\delta(r_1)$ is the degree of the operator in the graph $\check{\mathcal{G}}$.

Fig. 82.5 Equivalent electrical network of the graph in Fig. 82.4. The potential difference $V_{v_3} - V_{\mathcal{R}}$ is the effective resistance between v_3 and common resistor node $\{r_1, r_2\}$



Proof. The statement of the corollary follows directly from Proposition 2. ■

The implication of Corollary 3 is that the operator should select the UAV with the largest number of neighboring UAVs to maximize the open loop \mathcal{H}_2 norm of the system, independent of other network features.

82.3.2 Independent Signal Operator

The matrix $-A(\mathcal{G}, \mathcal{R})^{-1}$ in (82.6) is not unfamiliar to graph theorists and describes features of an equivalent electrical network interpretation of the graph (Bollobás 1998; Baroah and Hespanha 2006). In this interpretation, the edges \hat{E} of $\hat{\mathcal{G}}$ are replaced with one-ohm resistors and the nodes \hat{V} by connection points between resistors. In addition, all connection points corresponding to the node set R are electrically shorted. The effective resistance between two connection points in an electrical network is defined as the potential drop between the two points when a one-Amp current source is connected across the two points. This interpretation is that the i -th diagonal element of $-A(\mathcal{G}, \mathcal{R})^{-1}$ is the effective resistance $E_{\text{eff}}(v_i)$ between the common shorted external agents R and node v_i (Bollobás 1998). An example of the equivalent electrical network of the graph in Fig. 82.4 is displayed in Fig. 82.5.

With the graph-based machinery of the effective resistance, the open loop \mathcal{H}_2 norm can be reformulated for another special operator pair. Specifically, the following proposition presents an equivalent relation for the open loop \mathcal{H}_2 norm for the case where there is an independent (or distinct) control signal delivered along each edge of \mathcal{E}_R , described in Sect. 82.2.3.2 and denoted with the operator pair \mathcal{R}_d .

Proposition 4 *For a connected graph \mathcal{G} , if each operator has exactly one edge, and each edge in \mathcal{R}_d supplies an independent signal, then*

$$\|G_{\mathcal{G}, \mathcal{R}_d}(s)\|_2^2 = \frac{1}{2} \sum_{v_i \in \pi(\mathcal{E}_R)} E_{\text{eff}}(v_i). \quad (82.7)$$

Proof. The matrix $M(\mathcal{R}_d)$ is a diagonal matrix with $[M(\mathcal{R}_d)]_{ii} = 1$ if $v_i \in \pi(\mathcal{E}_R)$ and $[M(\mathcal{R}_d)]_{ii} = 0$ otherwise. Therefore,

$$[M(\mathcal{R}_d) A(\mathcal{G}, \mathcal{R}_d)^{-1}]_{ii} = \begin{cases} [A(\mathcal{G}, \mathcal{R}_d)^{-1}]_{ii} & \text{if } v_i \in \pi(\mathcal{E}_R) \\ 0 & \text{otherwise.} \end{cases}$$

From (82.6) and the effective resistance interpretation of $[A(\mathcal{G}, \mathcal{R}_d)^{-1}]_{ii}$, the statement of the lemma now follows. ■

From Proposition 4, the selection of directly controlled UAVs to maximize the open loop \mathcal{H}_2 norm will, in general, involve selecting UAVs with a large number of neighbors and a large separation between UAVs. The intuition is that the placement of resistors in series as opposed to parallel increases resistance; thus, maximizing the path length between UAVs effectively introduces more resistors in series and less in parallel, thus generating a larger effective resistance sum.

82.3.3 Generic Signal Operator

The open loop \mathcal{H}_2 norm of the special cases with constant and distinct operator pairs, \mathcal{R}_c and \mathcal{R}_d , respectively, can be used to bound the open loop \mathcal{H}_2 norm of a generic \mathcal{R} . The approach to relate the special operator cases involves fusing all the leaders in \mathcal{R} and causing a common control signal to be sent along all edges in \mathcal{E}_R , thus forming a constant operator pair $\widetilde{\mathcal{R}}_c$. Similarly, designating a distinct leader for each \mathcal{E}_R in \mathcal{R} forms a distinct operator pair $\widetilde{\mathcal{R}}_d$.

Additionally, scaling is required for the comparable operator pairs $\widetilde{\mathcal{R}}_c$ and $\widetilde{\mathcal{R}}_d$ such that a control signal applied to the edges in \mathcal{E}_R of $\widetilde{\mathcal{R}}_c$ maps to the same signal along the edges of \mathcal{R} and similarly for \mathcal{R} to $\widetilde{\mathcal{R}}_d$. The special operator pair $\widetilde{\mathcal{R}}_c$ is defined such that a control u_c of system $\widetilde{\mathcal{R}}_c$ maps to a control u of system \mathcal{R} so that there exists a transformation matrix H_1 such that $B(\widetilde{\mathcal{R}}_c)u_c = B(\mathcal{R})u$ and $\|u_c\|_2 = 1$ implies $\|u\|_2 = 1$ when $H_1 u_c = u$. Similarly, there exists a transformation matrix H_2 such that $B(\mathcal{R})u = B(\widetilde{\mathcal{R}}_d)u_d$ and so that $\|u\|_2 = 1$ implies $\|u_d\|_2 = 1$ when $H_2 u = u_d$. This ensures that $B(\mathcal{R}) = B(\widetilde{\mathcal{R}}_d)H_2$, $B(\widetilde{\mathcal{R}}_c) = B(\mathcal{R})H_1$, where $H_1 = \frac{1}{\sqrt{|\mathcal{R}|}} \mathbf{1} \in \mathbb{R}^{|\mathcal{R}| \times 1}$ and $H_2 \in \mathbb{R}^{|\mathcal{R}_d| \times |\mathcal{R}|}$. Arbitrarily ordering the edges in \mathcal{E}_R , then each edge $k = 1, \dots, |\mathcal{E}_R|$ corresponds to some $(r_j, v_i) \in \mathcal{E}_R$. The transformation matrix H_2 is defined as $[H_2]_{kj} = 1/\sqrt{\delta_j^v}$ for each edge k and zero otherwise. Therefore, $[B(\widetilde{\mathcal{R}}_c)]_i = \frac{\delta_i^r}{\sqrt{|\mathcal{R}|}}$ if $(r_j, v_i) \in \mathcal{E}_R$, and zero otherwise and $[B(\widetilde{\mathcal{R}}_d)]_{jk} = \sqrt{\delta_j^v}$ for each edge k , and zero otherwise.

When \mathcal{R} is a constant operator pair, $H_1 = I$ and $\mathcal{R} = \widetilde{\mathcal{R}}_c$. Similarly, when \mathcal{R} is a distinct operator pair, $H_2 = I$ and $\mathcal{R} = \widetilde{\mathcal{R}}_d$.

Proposition 5 For a graph \mathcal{G} with operator pair \mathcal{R} and operator pairs $\widetilde{\mathcal{R}}_c$ and $\widetilde{\mathcal{R}}_d$ defined above,

$$P(\mathcal{G}, \widetilde{\mathcal{R}}_c) \leq P(\mathcal{G}, \mathcal{R}) \leq P(\mathcal{G}, \widetilde{\mathcal{R}}_d).$$

Proof. Let $b_1, \dots, b_{|R|}$ be the columns of the matrix $B(\mathcal{R})$, then $B(\mathcal{R}_c) = \sum_{i=1}^{|R|} b_i$, and $B(\widetilde{\mathcal{R}}_c) = \frac{1}{\sqrt{|R|}} \sum_{i=1}^{|R|} b_i$. Hence, using the fact that $\|x\|_1 \leq \sqrt{p} \|x\|_2$ for all vectors $x \in \mathbb{R}^p$, for $y \in \mathbb{R}^n$

$$\begin{aligned} \|B(\widetilde{\mathcal{R}}_c)^T y\|_2^2 &= \frac{1}{|R|} \left(\sum_{i=1}^{|R|} b_i y \right)^2 \\ &\leq \frac{1}{|R|} \left(\sum_{i=1}^{|R|} |b_i y| \right)^2 \\ &\leq \frac{1}{|R|} \sum_{i=1}^{|R|} (b_i y)^2 \\ &= \frac{1}{|R|} \|B(\mathcal{R})^T y\|_1^2 \\ &\leq \|B(\mathcal{R})^T y\|_2^2, \end{aligned}$$

and $B(\widetilde{\mathcal{R}}_c)B(\widetilde{\mathcal{R}}_c)^T \leq B(\mathcal{R})B(\mathcal{R})^T$. Similarly, $B(\mathcal{R})B(\mathcal{R})^T \leq B(\widetilde{\mathcal{R}}_d)B(\widetilde{\mathcal{R}}_d)^T$. Noting that $A(\mathcal{G}, \mathcal{R}) = A(\mathcal{G}, \widetilde{\mathcal{R}}_c) = A(\mathcal{G}, \widetilde{\mathcal{R}}_d)$, the controllability gramian consequently has the relationship,

$$\begin{aligned} P(\mathcal{G}, \widetilde{\mathcal{R}}_c) &= \int_0^\infty e^{A(\mathcal{G}, \mathcal{R})\tau} B(\widetilde{\mathcal{R}}_c)B(\widetilde{\mathcal{R}}_c)^T e^{A(\mathcal{G}, \mathcal{R})^T \tau} d\tau \\ &\leq \int_0^\infty e^{A(\mathcal{G}, \mathcal{R})\tau} B(\mathcal{R})B(\mathcal{R})^T e^{A(\mathcal{G}, \mathcal{R})^T \tau} d\tau \\ &= P(\mathcal{G}, \mathcal{R}), \end{aligned}$$

and similarly, $P(\mathcal{G}, \mathcal{R}) \leq P(\mathcal{G}, \widetilde{\mathcal{R}}_d)$. ■

The groundwork is now in place to present bounds for the open loop \mathcal{H}_2 norm for a generic operator pair \mathcal{R} .

Lemma 6 For a graph \mathcal{G} and operator pair \mathcal{R} ,

$$\frac{1}{2|R|} |\mathcal{E}_R| \leq \|G_{\mathcal{G}, \mathcal{R}}(s)\|_2^2 \leq \frac{1}{2} \sum_{v_i \in \pi(\mathcal{E}_R)} \alpha_i E_{\text{eff}}(v_i),$$

where

$$\alpha_i = \sum_{(r_j, v_i) \in \mathcal{E}_R} \delta_j^v,$$

that is, α_i is the sum of the non-operator degrees for each operator attached to v_i .

Proof. From Proposition 5,

$$\left\| G_{\mathcal{G}, \widetilde{\mathcal{R}}_c}(s) \right\|_2^2 \leq \left\| G_{\mathcal{G}, \mathcal{R}}(s) \right\|_2^2 \leq \left\| G_{\mathcal{G}, \widetilde{\mathcal{R}}_d}(s) \right\|_2^2,$$

and

$$\begin{aligned} \text{tr}(P(\mathcal{G}, \widetilde{\mathcal{R}}_c)) &= \text{tr}(B(\widetilde{\mathcal{R}}_c)^T A(\mathcal{G}, \mathcal{R})^{-1} B(\widetilde{\mathcal{R}}_c)) \\ &= \frac{1}{|R|} \text{tr}(B(\mathcal{R}_c)^T A(\mathcal{G}, \mathcal{R})^{-1} B(\mathcal{R}_c)) \\ &= \frac{1}{|R|} \text{tr}(P(\mathcal{G}, \mathcal{R}_c)). \end{aligned}$$

The matrix $M(\widetilde{\mathcal{R}}_d)$ is diagonal with $[M(\widetilde{\mathcal{R}}_d)]_{ii} = \alpha_i$ if $(r_j, v_i) \in \mathcal{E}_R$ and $[M(\widetilde{\mathcal{R}}_d)]_{ii} = 0$ otherwise, and so

$$[M(\widetilde{\mathcal{R}}_d)A(\mathcal{G}, \widetilde{\mathcal{R}}_d)^{-1}]_{ii} = \begin{cases} \alpha_i [A(\mathcal{G}, \mathcal{R}_d)^{-1}]_{ii} & \text{if } v_i \in \pi(\mathcal{E}_R) \\ 0 & \text{otherwise.} \end{cases}$$

Hence,

$$\begin{aligned} \text{tr}(P(\mathcal{G}, \widetilde{\mathcal{R}}_d)) &= \frac{1}{2} \sum_{v_i \in \pi(\mathcal{E}_R)} [M(\widetilde{\mathcal{R}}_d)A(\mathcal{G}, \widetilde{\mathcal{R}}_d)^{-1}]_{ii} \\ &= \frac{1}{2} \sum_{v_i \in \pi(\mathcal{E}_R)} \alpha_i [A(\mathcal{G}, \mathcal{R}_d)^{-1}]_{ii}. \end{aligned}$$

Applying Proposition 2 and the proof of Proposition 4, the proof of the lemma follows. ■

Examining Lemma 6, the lower bound of the open loop \mathcal{H}_2 norm of the network dynamics increases as the ratio of operator edges to operators increases. The upper bound increases with the number of UAVs directly controlled by the operators. The upper bound also increases with the separation of the operators as per the effective resistance discussion for Proposition 4.

An in-depth analysis of $\left\| G_{\mathcal{G}, \widetilde{\mathcal{R}}_d}(s) \right\|_2^2$ was undertaken for a specialized class of graphs, namely, trees \mathcal{T} (Chapman and Mesbahi 2013). To apply some of these

results to more generalized connected graphs, consider any spanning tree \mathcal{T}_s of a connected graph \mathcal{G} . In terms of the electrical resistance analogy, the resistor network corresponding to \mathcal{T}_s is formed by removing resistors from the resistor network corresponding to \mathcal{G} . Rayleigh's monotonicity law states that if the edge resistance in an electrical network is decreased, then the effective resistance between any two agents in the network can only decrease (Bollobás 1998). Consequently, $\left\|G_{\mathcal{G}, \widetilde{\mathcal{R}}_d}(s)\right\|_2^2 \leq \left\|G_{\mathcal{T}_s, \widetilde{\mathcal{R}}_d}(s)\right\|_2^2$, that is, the open loop \mathcal{H}_2 norm of the system with the coordination graph \mathcal{G} , is bounded above by the open loop \mathcal{H}_2 norm system with the coordination graph \mathcal{T}_s corresponding to any of the spanning trees of \mathcal{G} .

82.4 Topology Design

As a group of networked UAVs require only nonphysical interconnections for their coordinated behavior, they have the advantage that their inter-vehicle coordination graph can be *rewired*. This observation leads to an online method for improving network manageability, namely, via a judicious topology design. The goal is to design network topologies that exhibit good control authority. The metric for manageability used in the following analysis is the open loop \mathcal{H}_2 norm of the network dynamics, described in Sect. 82.3.

82.4.1 Constraint Graph

Before continuing, it is important to explore constraints on the underlying graph topology that are prescribed by the swarming application. For all UAVs in the swarm to reach a common consensus, there must be some path connecting each and every UAV along some subset of graph edges, that is, the graph must remain connected for any topology redesign. The nodes within the graph represent the UAVs. As no loss or gain of UAVs is assumed, the number of nodes n remains fixed. An edge $\{v_i, v_j\} \in E$ in the graph \mathcal{G} indicates that UAV v_i is sensing or has communicated the relative information of UAV v_j and vice versa; thus, the total sensing or communication load on the network is $2|E|$, and so fixing the number of edges equates to maintaining this load. Individual UAVs may also have limited sensing or communication capabilities. From a graph-theoretic perspective, this is represented as an upper bound on a node's degree δ_i . Finally, the ability of UAV v_i to perform accurate relative sensing or low-cost communication with UAV v_j is limited by the Euclidean separation of the two UAVs. A *Euclidean-based constraint graph* $\check{\mathcal{G}}_c = (\check{V}, \check{E}_c)$ can be formed where edges exist between UAVs when they lie in each other's sensing range. For topology design, only a subgraph of this constraint graph may be selected.

Figure 82.6a depicts a Euclidean-based constraint graph on 20 nodes. The graph in Fig. 82.6b satisfies the connected Euclidean subgraph constraint with the added global and local load sensing constraints that $|E| \leq 51$ and $\delta_i \leq 7$ for all $v_i \in V$.

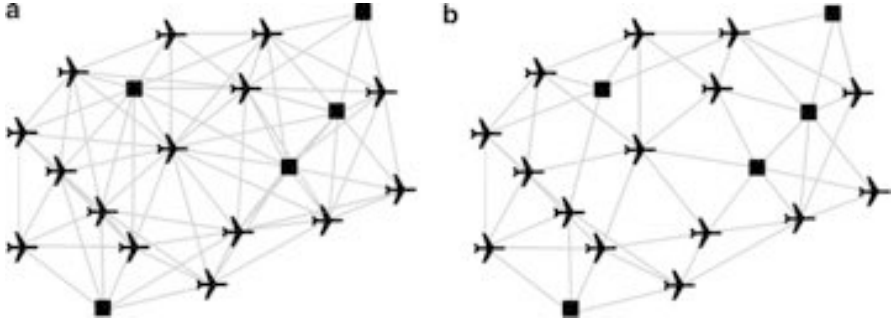


Fig. 82.6 Agent graph with operator agents (*squares*). (a) A Euclidean constraint graph $\check{\mathcal{G}}_c = (V, \check{E}_c)$. (b) A connected subgraph of (a) meeting the global and local edge constraints, $|E| \leq 51$ and $\delta_i \leq 7$ for all $v_i \in V$

82.4.2 Network Design Protocol

This section introduces a network design protocol to improve the open loop \mathcal{H}_2 norm of the network. The proposed protocol involves decentralized edge trading (rewiring) between neighboring nodes in the network. The protocol runs over the spanning trees of \mathcal{G} with the objective of increasing $\|G_{\mathcal{G}, \mathcal{R}}(s)\|_2^2$ via increasing $|\mathcal{E}_R|$ and $E_{\text{eff}}(v_i)$ for all $v_i \in \pi(\mathcal{E}_R)$ as illustrated through Lemma 6.

The protocol involves edge trades between neighboring agents executed concurrently and/or in a random agent order while maintaining a connected tree at each iteration. The general procedure is to randomly select a spanning tree \mathcal{T}_s of \mathcal{G} and apply Protocol 1 for some finite number of edge trades and then repeat with a new spanning tree. Within the protocol, an edge removal and subsequent addition is denoted by the notation, \setminus for set minus and \cup for set addition, and the set $\mathcal{I}(v_i)$ contains all nodes that are neighbors of v_i and lie on the shortest path between v_i and any $r_j \in R$. Formally,

$$\mathcal{I}(v_i) := \{v \in \mathcal{N}(v_i) | \exists r \in R, d(v, r) < d(v_i, r)\}.$$

In this direction, the special set of nodes that lie on any of the shortest paths between nodes in \mathcal{R} is defined as the *main path nodes*, that is, those nodes that are neighbors with an operator or with $|\mathcal{I}(v_i)| > 1$. An example of the described notation appears in Fig. 82.7.

The set $\mathcal{I}(v_i)$ can be established locally with a broadcast by the human operator-controlled UAVs, informing their neighbors that they are being directly controlled. This signal can be rebroadcast through neighbors to all UAVs in the swarm, and in doing so, all UAVs are aware of the local directions of the controlled vehicles and more specifically $\mathcal{I}(v_i)$ – their neighbors that are closer to the controlled vehicles set $\pi(\mathcal{E}_R)$. The protocol can be executed on any arbitrary node $v_i \in V$ and requires

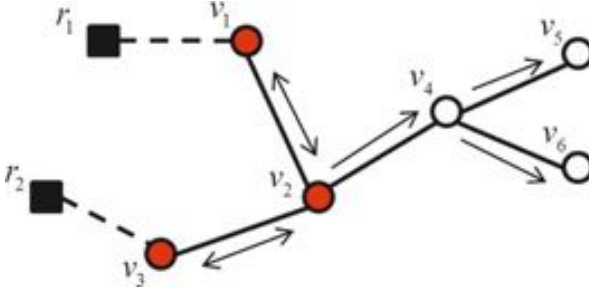


Fig. 82.7 Example of protocol notation. The operators are marked with *squares* and the network nodes with *circles*. Arrows on the edges indicate presence of a directed path away from the operators. The set $\mathcal{I}(\cdot)$ is defined for each node as $\mathcal{I}(v_1) = \{r_1, v_2\}$, $\mathcal{I}(v_3) = \{r_2, v_2\}$, $\mathcal{I}(v_2) = \{v_1, v_3\}$, $\mathcal{I}(v_4) = \{v_2\}$, and $\mathcal{I}(v_5) = \mathcal{I}(v_6) = \{v_4\}$. The main path nodes $\{v_1, v_2, v_3\}$ are marked with *solid (red) circles*

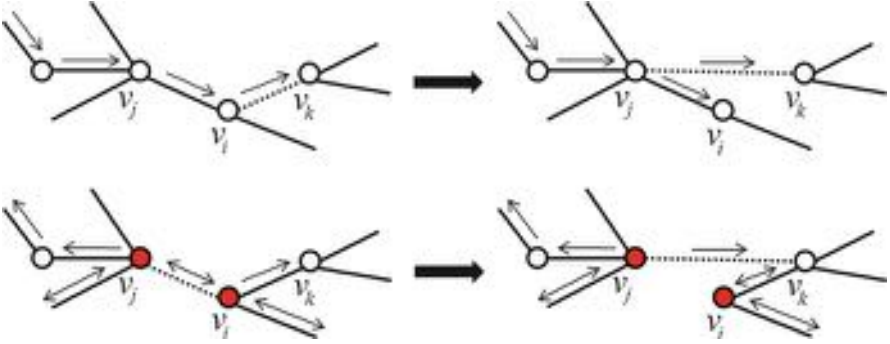


Fig. 82.8 (1a)–(1b) Edge swap of Protocol 1 when $|\mathcal{I}(v_i)| = 1$ (condition 1). (2a)–(2b) Edge swap of Protocol 1 when $|\mathcal{I}(v_i)| > 1$ (condition 2). The main path nodes are marked with *solid (red) circles*, and *dotted edges* indicate edges that are removed and added in the protocol

the knowledge of $\mathcal{I}(v_i)$ and $\mathcal{N}(v_i)$ (with the neighborhood defined over $\check{\mathcal{G}}$); hence, the protocol is distributed and decentralized – operating on only local information.

The protocol involves two conditions on the spanning tree: the first pushes nodes toward the main path nodes and the other increases the main path node set. The purpose of both conditions is to increase the degree of operators in R thus increasing $|\mathcal{E}_R|$ and in turn to increase the lower bound in Lemma 6. The second condition's purpose is to increase the effective resistance for nodes in set π (\mathcal{E}_R) and, in doing so, increase the upper bound in Lemma 6. The conditions of the protocol are demonstrated pictorially in Fig. 82.8.

The following lemmas appear in Chapman and Mesbahi (2010), and readers are referred to this paper for the corresponding proofs.

Protocol 1: Increasing open loop \mathcal{H}_2 edge swap

```

foreach Node  $v_i \in V$  do
  if  $\exists v_j, v_k \in \check{V}$  s.t.  $v_j \in \mathcal{I}(v_i), v_k \in \mathcal{N}(v_i) \setminus \mathcal{I}(v_i), \{v_j, v_k\} \notin E$  and  $\{v_j, v_k\} \in \check{E}$  then
     $E \rightarrow E \cup \{v_j, v_k\};$ 
    if  $|\mathcal{I}(v_i)| = 1$  then
       $E \rightarrow E \setminus \{v_i, v_k\};$ 
    else
       $E \rightarrow E \setminus \{v_i, v_j\};$ 
    end
  end
end

```

Lemma 7 Under conditions of Protocol 1, $|\mathcal{E}_R|$ only increases and the graph remains connected.

Lemma 8 For a tree \mathcal{T} , under the second conditions of Protocol 1, $E_{\text{eff}}(v_i)$ for $v_i \in \pi(\mathcal{E}_R)$, monotonically increases.

Lemma 7 has the effect of compressing the network about the main path agents. On the other hand, Lemma 8 adds agents to the main path of \mathcal{T} and in doing so elongates the main path. From a network standpoint, the path between operators is increased. While from an equivalent resistor circuit interpretation, parallel resistors are replaced by series resistors, increasing effective resistance for $v_i \in \pi(\mathcal{E}_R)$. As Protocol 1 iterates over the node set V but involves edges swaps with nodes in set R , then the number of edges attached to nodes in R can only increase.

82.4.3 Topology Design for Wind-Gust Disturbance Rejection

Unpredictable environmental perturbation such as wind gusts is considered one of the main factors that risk a UAV program failure (Drezner et al. 1999). Fulghum (1998) commented that wind could be particularly onerous in cities where buildings produce microbursts.

It has previously been demonstrated that operators applying a *grounded signal* (all operators apply a common constant signal) are able to control the mean consensus value of a wind-gust disturbed system (Chapman and Mesbahi 2013). The following example investigates improving the disturbance rejection for a wind-gust disturbed system via topology design.

82.4.3.1 Wind-Gust Perturbed Model

A vertical wind-gust w_g is not a white noise random signal but has a power spectral density given in Dryden form as (Stevens and Lewis 2003),

$$\Phi_w(\omega) = 2L\sigma^2 \frac{1 + 3L^2\omega^2}{(1 + L^2\omega^2)^2}, \quad (82.8)$$

with frequency in rad/s ω , turbulence intensity σ , and turbulence scale length divided by true airspeed L .

The power spectral density (82.8) can be factored (Stevens and Lewis 2003) as,

$$\Phi_w(s) = H_w(s)H_w(-s),$$

where

$$H_w(s) = \sigma \sqrt{\frac{6}{L}} \frac{s + 1/L\sqrt{3}}{s^2 + 2s/L + 1/L^2}.$$

Now a realization of $H_w(s)$ corresponding to w , a white noise input with zero mean and unit variance W , is given by

$$\begin{aligned} \dot{z}(t) &= \begin{bmatrix} 0 & 1 \\ -\frac{1}{L^2} & -\frac{2}{L} \end{bmatrix} z(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t), \\ &= A_w z(t) + B_w w(t), \end{aligned} \quad (82.9)$$

where $z(t) \in \mathbb{R}^2$ denotes the internal states of the vertical wind gust and

$$\begin{aligned} w_g(t) &= \gamma \begin{bmatrix} \frac{1}{L\sqrt{3}} & 1 \end{bmatrix} z(t) \\ &= C_w z(t), \end{aligned} \quad (82.10)$$

where $\gamma = \sigma \sqrt{6/L}$.

The vertical wind model dynamics can now be incorporated into the UAV network dynamics.

For small UAVs and MAVs, it is common that a wind gust has a first-order effect on the vehicle dynamics (Shyy et al. 2007). This means that a wind-gust disturbance to the vehicle dynamics can be replaced with a disturbance on the network dynamics. Due to the close proximity of UAV swarms, the same vertical gust $w_g(t) \in \mathbb{R}$ is assumed to act upon all UAVs in the swarm. The network dynamics is running consensus on a time-varying altitude state $x(t)$, for example, to maintain a common formation height for surveillance and communication benefits. Subsequently, the network dynamics running a perturbed particle model assumes the form

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \begin{bmatrix} A(\mathcal{G}, \mathcal{R})x(t) + B(\mathcal{R})u(t) \\ \dot{z}(t) \end{bmatrix} + \begin{bmatrix} H \\ 0 \end{bmatrix} w_g(t) \\ &= \begin{bmatrix} A(\mathcal{G}, \mathcal{R})x(t) \\ A_w z(t) + B_w w(t) \end{bmatrix} + \begin{bmatrix} B(\mathcal{R}) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} H \\ 0 \end{bmatrix} C_w z(t) \\ &= \begin{bmatrix} A(\mathcal{G}, \mathcal{R})x(t) \\ A_w z(t) + B_w w(t) \end{bmatrix} + \begin{bmatrix} B(\mathcal{R}) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} A_z \\ 0 \end{bmatrix} z(t) \\ &= \begin{bmatrix} A(\mathcal{G}, \mathcal{R}) & HC_w \\ 0 & A_w \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B(\mathcal{R}) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ B_w \end{bmatrix} w(t), \end{aligned} \quad (82.11)$$

where $H = [h_1, \dots, h_n]^T$ and $h_i \in \mathbb{R}$ is a scaling factor specific to UAV i 's gust-vehicle interaction.

It is assumed the operators can measure the altitude of all vehicles with some white measurement noise $v(t) \in \mathbb{R}^n$ with correlation matrix $V \in \mathbb{R}^{n \times n}$. Restating (82.11) in compact form,

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \mathcal{A}_D \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \mathcal{B}_D u(t) + G_D w(t) \\ y(t) &= \mathcal{C}_D \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + v(t), \end{aligned} \quad (82.12)$$

where \mathcal{A}_D , \mathcal{B}_D , and G_D are apparent from (82.11) and $y(t) \in \mathbb{R}^n$ is the operators' observations. As the operators can observe all the followers' states, but not the internal dynamics of the wind gust, then it follows that $\mathcal{C}_D = [I_{n \times n} \ 0_{n \times 2}]$.

82.4.3.2 \mathcal{H}_2 Controller Design

An \mathcal{H}_2 controller will now be designed for wind-gust rejection for the network dynamics. A consequence of Proposition 1 is that, if the underlying network is connected, $A(\mathcal{G}, \mathcal{R})$ has only negative eigenvalues. Examining A_w , $\lambda_i(A_w) = -1/L$ for $i = 1, 2$ and as $L > 0$, it follows that A_w has only negative eigenvalues. Therefore, the system is stabilizable and detectable, so a stabilizing compensator can be formed.

Hence, given a performance measure

$$J = \lim_{t \rightarrow \infty} \mathbb{E} \{x(t)^T Q x(t) + u(t)^T R u(t)\},$$

a compensator that minimizes J can be designed with the form

$$\begin{aligned} \dot{\hat{x}}(t) &= (\mathcal{A}_D - \mathcal{B}_D K_c - K_f \mathcal{C}_D) \hat{x}(t) + K_f y(t) \\ u(t) &= K_c \hat{x}(t), \end{aligned}$$

where $\hat{x}(t)$ is an estimate of $[x(t)^T z(t)^T]^T$, $K_c \in \mathbb{R}^{r \times n}$ is the compensator gain, and $K_f \in \mathbb{R}^{n \times n}$ is the filter gain. The compensator and filter gains satisfy $K_c = R^{-1} \mathcal{B}_D^T P$ and $K_f = S \mathcal{C}_D^T V^{-1}$, and P and S are the solutions of the algebraic Riccati equations,

$$0 = \mathcal{A}_D^T P + P \mathcal{A}_D + Q - P \mathcal{B}_D R^{-1} \mathcal{B}_D^T P$$

and

$$0 = \mathcal{A}_D S + S \mathcal{A}_D^T + G_D W G_D^T - S \mathcal{C}_D^T V^{-1} \mathcal{C}_D S.$$

The corresponding optimal performance cost is then

$$J^* = \text{tr}(P K_f V K_f^T) + \text{tr}(S Q). \quad (82.13)$$

Fig. 82.9 Original network with controlled UAVs marked with squares

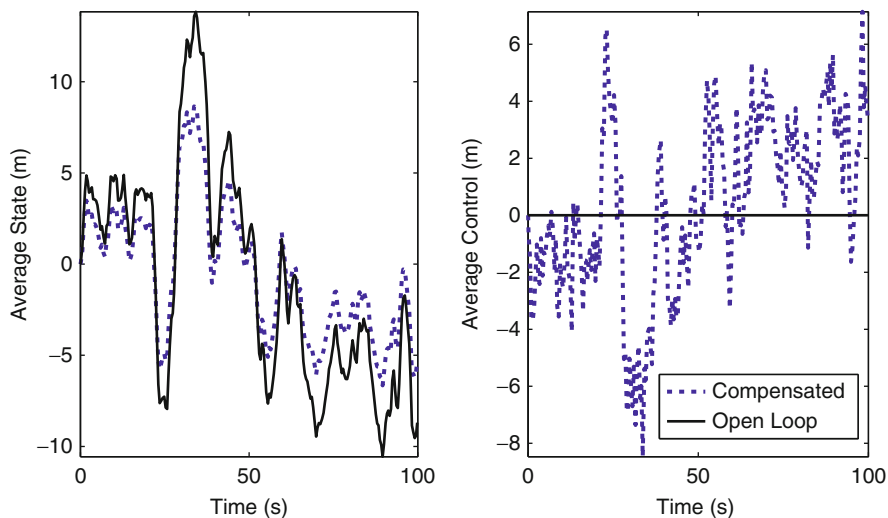
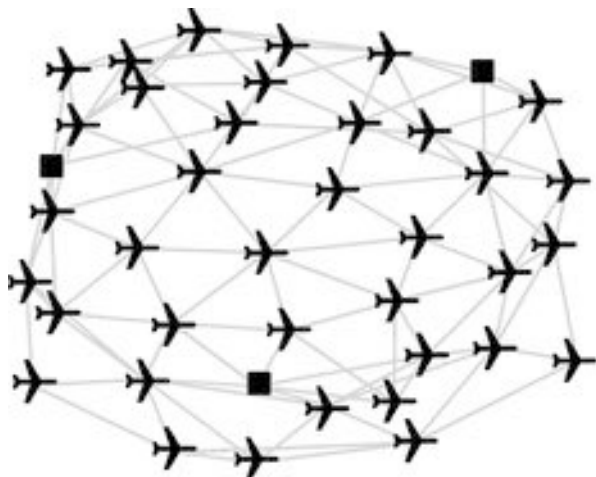


Fig. 82.10 Average state and control over time for the compensated (closed loop \mathcal{H}_2 controller) and grounded constant (open loop) control for the wind-perturbed network in Fig. 82.9

For the swarming model (82.12), the stabilizing compensator was applied to a 40-UAV swarm exposed to a vertical wind gust with 3 UAVs directly controlled. The constant parameters of the system have $Q = I_{37 \times 37}$, $R = 10I_{3 \times 3}$, $V = 0.1I_{37 \times 37}$, $H = \mathbf{1}$, $W = 1$, $L = 3.49$, $\sigma = 10$, and initial altitude set as $x(0) = 0$. The network topology is depicted in Fig. 82.9. The performance with respect to the average state and control to maintain a desired altitude ($x(t) = 0$) is compared to the grounded signal control in Fig. 82.10.

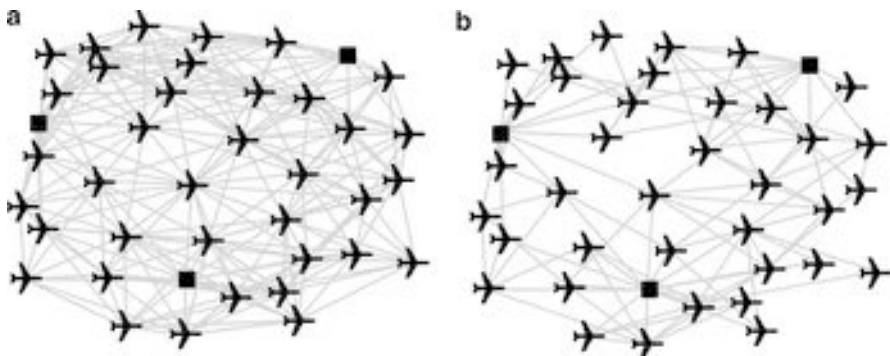


Fig. 82.11 (a) Edge constraint graph and (b) resultant leader-follower consensus network after applying Protocol 1 to 50 spanning trees of the graph in Fig. 82.9. The controlled UAVs are marked with *squares*

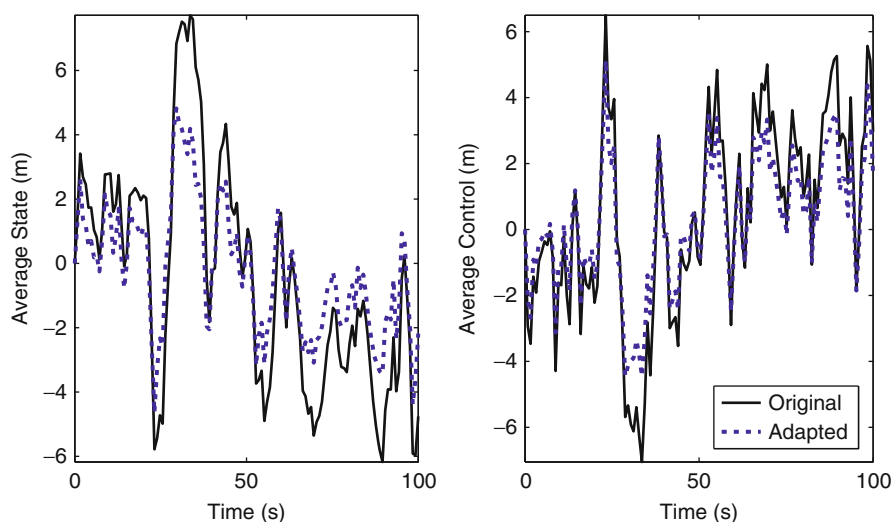


Fig. 82.12 Average state and control over time for the wind perturbed system corresponding to Fig. 82.9 before and after applying Protocol 1

Protocol 1 is applied to the network topology to improve the effectiveness of the controller. The protocol is restricted to only edges within the constraint graph displayed in Fig. 82.11a and limited to a maximum of 8 neighbors for each noncontrolled UAV. Running the protocol on the graph in Fig. 82.9 over 50 spanning trees, involving 289 edge trades, the final graph displayed in Fig. 82.11b is produced. The resultant leader-follower consensus model has a $\|G_{\mathcal{G}, \mathcal{R}}(s)\|_2^2 = 11.3$ compared to the original system with 4.6. The resultant graph and its response to wind gusts are displayed in Figs. 82.11 and 82.12. The optimal performance cost J^* as defined in (82.13) decreased from 14,815 to 3,265.

82.5 Model Reduction

The \mathcal{H}_2 design framework produces controllers of order at least equal to that of the underlying plant and often higher due to the inclusions of dynamic weights. The corresponding control laws may be too involved with regard to practical implementation and simpler designs are preferred. For this purpose, one can either reduce the order of the plant model prior to controller design or reduce the controller in the final stage, or both. The following section proposes an a priori model reduction method using model truncation but in such a way that the reduced order system corresponds to a new network.

82.5.1 Model Truncation

Model truncation involves partitioning the state vector x into $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ where x_2 is the vector of $n-k$ states which will be removed, rearranging the system model such that

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du.$$

A k -th order truncation subsequently produces the approximate model

$$\dot{x}_1 = A_{11}x_1 + B_1u$$

$$y = C_1x_1 + Du,$$

with transfer function $G_a(s) = C_1 (sI - A_{11})^{-1} B_1 + D$. A common error metric is the error between new full model system $G(s)$ and truncated model $G_a(s)$ with respect to the Hankel norm and infinity norm, that is, $\|G(s) - G_a(s)\|_H$ and $\|G(s) - G_a(s)\|_\infty$, respectively. The Hankel and infinity norm of any stable transfer function $E(s)$ is defined as $\|E(s)\|_H := \sqrt{\max_i |\lambda_i(PP_{obs})|}$ and $\|E(s)\|_\infty := \max_w \bar{\sigma}(G(jw))$, where P and P_{obs} are the controllability and observability Gramian of $E(s)$, respectively, and $\bar{\sigma}(\cdot)$ denotes the singular value of a matrix. The optimal k -th order truncation $G_a^H(s)$ with respect to the Hankel norm is

$$\|G(s) - G_a^H(s)\|_H = \sigma_{k+1}, \quad (82.14)$$

where $\sigma_k = \sqrt{\lambda_k(PP_{obs})}$. Another common form of truncation is balance truncation and balanced realization $G_a^B(s)$ which boasts the bound

$$\|G(s) - G_a^B(s)\|_\infty \leq 2(\sigma_{k+1} + \dots + \sigma_n). \quad (82.15)$$

These methods and their corresponding performance guarantees will be used as benchmarks for the following network-based truncation method. For descriptions of these and forms of truncation, see Dullerud and Paganini (2000).

82.5.2 Model Reduction Using Equitable Distance Partitions

The traditional methods of truncation lose characteristics of the underlying network topology. The following proposes a graph-based form of model truncation. The approach is to fuse nodes, reducing the model order, while still maintaining much of the network topology; see Fig. 82.13.

A partition of a graph is the grouping of nodes into subsets called cells. The i -th cell is denoted C_i . An *equitable partition* is one in which each node in C_j has the same number of neighbors in C_i for all i, j . A *distance partition* is one in which each cell is composed of all nodes the same distance from some node v_i . A distance partition over \mathcal{G} is defined with respect to the operators. If each operator is the sole member of a cell, a cell with more than one member exists, and the partition is equitable, then it has been shown that the graph is uncontrollable (Rahmani et al. 2009), and as such there exists a nonempty uncontrollable subspace which can be truncated from the model without introducing any model error. Further it was shown that symmetry within the graph topology with respect to the leader agents induces an equitable partition (Rahmani et al. 2009).

This framework can be extended to generate a graph-based reduced model by fusing together the states of all agents within a cell. Specifically the reduced model's states are

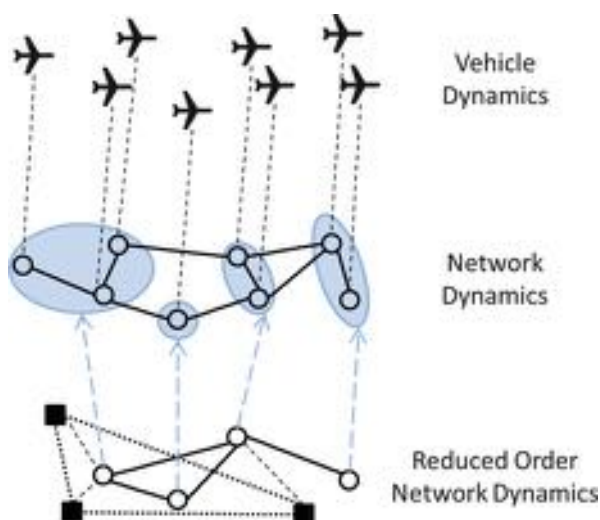


Fig. 82.13 A reduced order network dynamics, full network dynamics, and vehicle dynamics. The operators perceive a more manageable reduced order network dynamics formed by fusing nodes into meta-nodes

$$z_j = \frac{1}{|C_j|} \sum_{v_i \in C_j} x_i,$$

for all j , that is, the average state value within the cell. For the case where the partition is an *equitable distance partition* with respect to some r_i , then the same partition will be generated if it was calculated with respect to any r_j . As mentioned above, this model reduction has exactly the same input-output characteristics as the full state system, that is, no model error is introduced. This method for model reduction was introduced for a single leader in Martini et al. (2008). As most leader-follower consensus systems do not have an equitable distance partition, the partition depends on the selection of the leader r_i that defines it. To remove this dependence and provide equal contribution from all operators, the distance partition is modified to a new form dubbed an *operator partition*. This new partition is formally defined as follows.

Definition 9 An operator partition is composed of cells such that all nodes are initially designated to operator cells \overline{C}_j , where nodes who share the same closest operator are grouped together, that is, $\overline{C}_j = \{v_i : r_j = \operatorname{argmin}_k d(r_k, v_i)\}$. If nodes are equidistant between a set of operators, then membership in one of the corresponding operator cells is randomly chosen. Each leader cell \overline{C}_j is then distance partitioned with respect to their corresponding operator node r_j to form operator partition cells.

The proposed model reduction procedure involves fusing the states of all agents within each of the operator partition cells. In the case where the network contains an equitable distance partition, then combinations of the operator partition cells form the distance partition cells. Consequently, the operator partition model reduction would not introduce modeling errors. The strength of this model reduction process is that it exploits the structural components of the graph which are close to being symmetric, introducing only small modeling errors corresponding to ignored asymmetries.

The reduced order model has system matrices

$$A_r = STA(\mathcal{G}, \mathcal{R}) T^T S \in \mathbb{R}^{n_r \times n_r}$$

$$B_r = STB(\mathcal{R}) \in \mathbb{R}^{n_r \times r}$$

$$C_r = T^T S \in \mathbb{R}^{n \times n_r},$$

where $S \in \mathbb{R}^{n_r \times n_r}$, n_r are the number of cells in the operator partition, $[S]_{ii} = \frac{1}{\sqrt{|C_i|}}$, $T \in \mathbb{R}^{n_r \times n}$ and $[T]_{ij} = 1$ if $v_i \in C_j$ and $[T]_{ij} = 0$ otherwise.

Graphically, $\overline{A} = TA(\mathcal{G}, \mathcal{R}) T^T$ and $\overline{B} = TB(\mathcal{R})$ correspond to the state matrix of the reduced order graph $\overline{\mathcal{G}} = (\overline{V}, \overline{E})$. This can be formed by fusing the nodes of $\check{\mathcal{G}}$, forming *meta-nodes*, corresponding to the operator partition and reweighting the edges accordingly. Therefore, each node in $\overline{\mathcal{G}}$ corresponds to a fused cell, and the

weight on edge $(v_i, v_j) \in \bar{E}$, \bar{w}_{ij} , is equal to the number of edges between nodes in C_i and in C_j in $\check{\mathcal{G}}$.

The interpretation of A_r and B_r is a graph with self-loops and can be formed by reweighting the edges of $\bar{\mathcal{G}}$ from \bar{w}_{ij} to w_{ij} such that

$$w_{ij} = \begin{cases} \frac{1}{\sqrt{|C_i||C_j|}} \bar{w}_{ij} & v_i, v_j \notin R \\ \frac{1}{\sqrt{|C_i|}} \bar{w}_{ij} & \text{otherwise.} \end{cases}$$

For the self-loops, one has

$$\begin{aligned} w_{ii} &= \frac{1}{|C_i|} \sum_{v_j \neq v_i} \bar{w}_{ij} - \sum_{v_j \neq v_i} w_{ij} \\ &= \frac{1}{|C_i|} \sum_{v_j \neq v_i} \bar{w}_{ij} - \frac{1}{\sqrt{|C_i|}} \left(\sum_{v_j \notin R \cup \{v_i\}} \frac{1}{\sqrt{|C_j|}} \bar{w}_{ij} + \sum_{v_j \in R} \bar{w}_{ij} \right) \end{aligned}$$

Among the benefits of this type of reduction is a simple graphical approach to model reduction, compared to traditionally more numerically intensive model

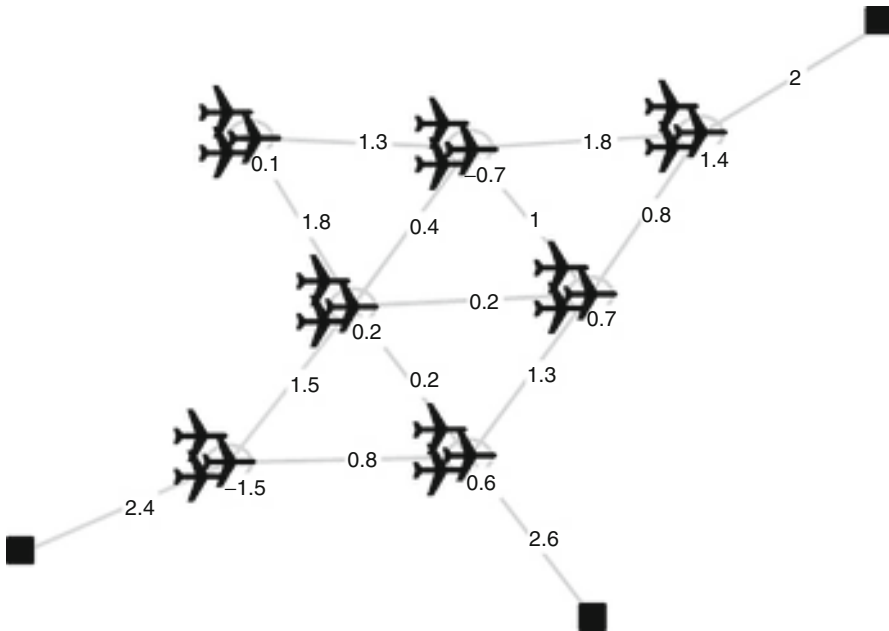


Fig. 82.14 Proposed model reduction graph $\bar{\mathcal{G}}$ for network in Fig. 82.9. The reduced weighted graph corresponding to A_r and B_r with operator modes (*squares*). Edges from a meta-node to itself (a self-loop) are denoted by a *circle* around the meta-node

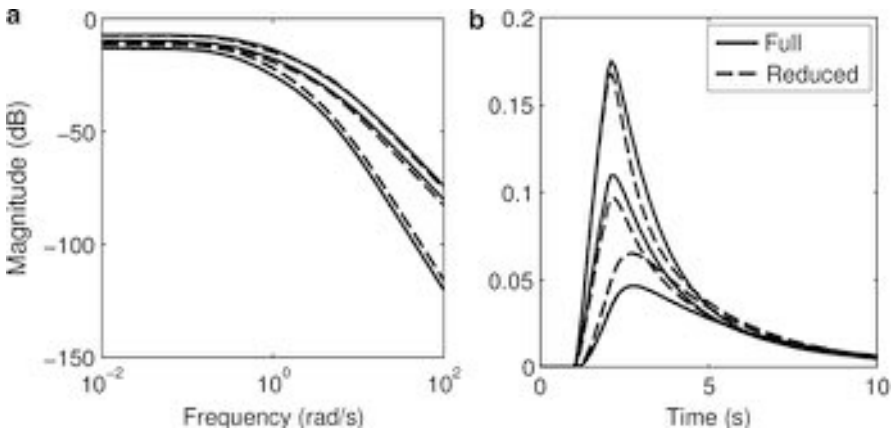


Fig. 82.15 (a) Singular values and (b) impulse response, for each of the three operators for the seventh-order operator partition model and the full state model (37 state) for a specific node

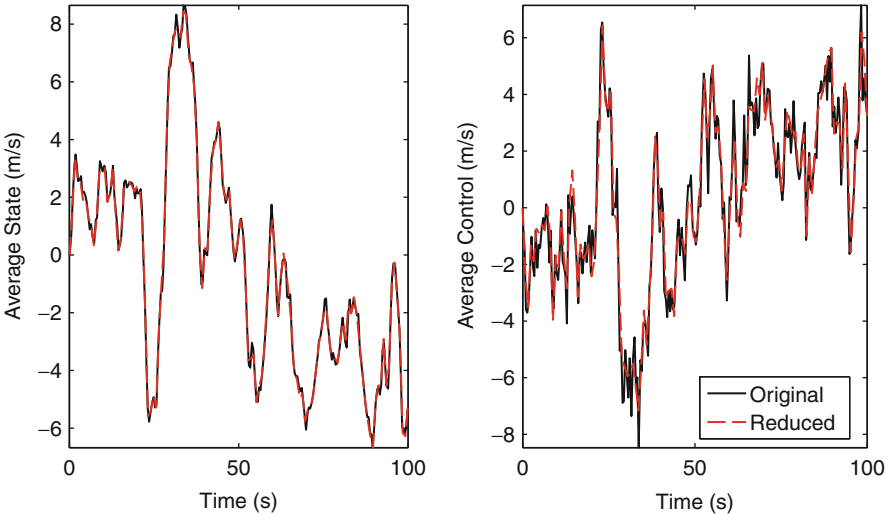


Fig. 82.16 Average state and control over time for the wind-perturbed system corresponding to Fig. 82.9 with original H_2 controller and controller partition reduced model (Fig. 82.11)

reduction techniques, for example, balanced residualization, as well as the potential to apply graph-theoretic analysis on the reduced model.

82.5.3 Model Reduction for Wind-Gust Disturbance Rejection

Revisiting the wind-gust example of Sect. 82.4.3, the proposed graph-based model reduction method can be performed on the original graph in Fig. 82.9 to produce a

lower-order more manageable graph for the operators to manipulate. A lower-order \mathcal{H}_2 controller can then be constructed based on this reduced model.

The operator partition model reduction method, when applied to the original network, reduces the order of the system from 37 to 7. The reduced graph corresponding to A_r and B_r is displayed in Fig. 82.14. The singular value and impulse response for a selected agent with respect to each of the controls are compared to the full state and reduced state systems in Fig. 82.15. Applying the norm metrics, then $\|G(s) - G_a(s)\|_H = 0.45$ and $\|G(s) - G_a(s)\|_\infty = 2.57$ compared to the benchmarks (82.14) and (82.15) with $\|G(s) - G_a^H(s)\|_H = 0.03$ and $\|G(s) - G_a^B(s)\|_H \leq 0.11$. The reduced model was used to produce an \mathcal{H}_2 controller, and its performance was compared to the full state \mathcal{H}_2 controller in Fig. 82.16.

Conclusion

This chapter presents network-theoretic approaches for the design of efficient models, interfaces, and topologies of the network for UAV swarming. A UAV swarm can be decomposed into a two-component hierarchical structure where the network dynamics drives the vehicle dynamics and in turn is influenced by the human operators. This two-component hierarchical model of the UAV swarm allows topological elements and algorithms to be employed to manage the swarm's performance and dynamics. The open loop \mathcal{H}_2 norm was analyzed for the network dynamics as a metric to quantify its performance. With this metric, the number of neighbors of the controlled UAVs and spacing of these UAVs were highlighted as characteristics of manageable swarms. An adaptive rewiring protocol based on the spanning trees of the graph was described for the use in the online design of networks for swarming with human operators. The protocol is decentralized across the UAVs in the swarm, increases the open loop \mathcal{H}_2 norm, and consequently produces effective interfaces for the human operators. Network properties pertaining to symmetry and node partitioning provided a network-centric method for model reduction. The method is a simple graphical approach involving fusing UAV nodes to form reduced order meta-nodes. The result is a lower-order model that corresponds to a manageable network for the operators to manipulate. The topology rewiring protocol and model reduction method were applied to UAV swarms perturbed by wind gusts with improved operator manageability and wind-gust rejection, to demonstrate the versatility of this work.

References

- P. Barooah, J.P. Hespanha, Graph effective resistance and distributed control: spectral properties and applications, in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, 2006, pp. 3479–3485
- B. Bollobás, *Modern Graph Theory* (Springer, New York, 1998)
- F. Bullo, J. Cortes, S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms* (Princeton University Press, Princeton, 2009)

- A. Chapman, M. Mesbahi, Semi-autonomous networks: network resilience and adaptive trees, in *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, no. 2, 2010, pp. 7473–7478
- A. Chapman, M. Mesbahi, Semi-autonomous consensus: network measures and adaptive trees. *IEEE Trans. on Autom. Control* **58**(1), 19–31 (2013)
- T. Cheviron, Robust control of an autonomous reduced scale helicopter in presence of wind gusts, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, 2006, pp. 1–22
- R. Diestel, *Graph Theory* (Springer, Berlin, 2000)
- J.A. Drezner, R.S. Leonard, G. Sommer, *Innovative Management in the DARPA High Altitude Endurance Unmanned Aerial Vehicle Program: Phase II Experience* (RAND, Washington, DC, 1999)
- G.E. Dullerud, F. Paganini, *A Course in Robust Control Theory – A Convex Approach* (Springer, New York, 2000)
- D.D. Fulghum, Miniature air vehicles fly into army's future. *Aviat. Week Space Technol.* **147**, 37–38 (1998)
- A. Ghosh, S. Boyd, Growing well-connected graphs, in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, 2006, pp. 6605–6611
- A. Ghosh, S. Boyd, A. Saberi, Minimizing effective resistance of a graph. *SIAM Rev.* **50**(1), 37–66 (2008)
- S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, R. Beard, *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, ed. by K.P. Valavanis (Springer, Dordrecht, 2007)
- J. Hall, Lateral control and observation of a micro aerial vehicle, in *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, 2007, pp. 8–11
- J.P. How, B. Bethke, A. Frank, D. Dale, J. Vian, Real-time indoor autonomous vehicle test environment. *IEEE Control Syst. Mag.* **28**(2), 51–64 (2008)
- Y. Kim, M. Mesbahi, On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE Trans. on Autom. Control* **51**(1), 116–120 (2006)
- X. Li, L. Cao, Largest Laplacian eigenvalue predicts the emergence of costly punishment in the evolutionary ultimatum game on networks. *Phys. Rev. E* **80**(6 Pt 2), 066101 (2009)
- S. Martini, M. Egerstedt, A. Bicchi, Controllability decompositions of networked systems through quotient graphs, in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, 2008, pp. 5244–5249.
- D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, in *International Conferences on Robotics and Automation*, Shanghai, 2011, pp. 2520–2525
- M. Mesbahi, M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks* (Princeton University Press, Princeton, 2010)
- N. Meskin, K. Khorasani, C.A. Rabbath, A hybrid fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances. *IEEE Trans. Control Syst. Technol.* **18**(6), 1422–1429 (2010)
- N. Michael, D. Mellinger, Q. Lindsey, V. Kumar, The GRASP multiple micro-UAV testbed. *IEEE Robot. Autom. Mag.* **17**(3), 56–65 (2010)
- T.J. Mueller (ed.), *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (AIAA, Danvers, 2001)
- L. Mutuel, R. Douglas, Controller design for low speed flight in turbulence, in *AIAA Guidance, Navigation, and Control Conference*, New Orleans, 1997, pp. 1111–1121
- R. Olfati-Saber, Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans. Auto. Control* **51**(3), 401–420 (2006)
- R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **95**(1), 215–233 (2007)
- D.J. Pines, F. Bohorquez, Challenges facing future micro-air-vehicle development. *J. Aircr.* **43**(2), 290–305 (2006)
- A. Rahmani, M. Ji, M. Mesbahi, M. Egerstedt, Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM J. Control Optim.* **48**(1), 162–186 (2009)

- W. Ren, R.W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control* (Springer, New York, 2007)
- S. Salsa, *Partial Differential Equations in Action: From Modelling to Theory*. (Springer, Milan, 2008)
- L.V. Schmidt, *Introduction to Aircraft Flight Dynamics* (AIAA, Reston, 1998)
- W. Shyy, Y. Lian, J. Tang, H. Liu, P. Trizila, B. Stanford, L. Bernal, C. Cesnik, P. Friedmann, P. Ifju, Computational aerodynamics of low Reynolds number plunging, pitching and flexible wings for MAV applications, in *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, vol. 24, no. 4, 2008, pp. 1–33
- W. Shyy, Y. Lian, J. Tang, D. Viieru, H. Liu, *Aerodynamics of Low Reynold Number Flyers* (Cambridge University Press, Cambridge, 2007)
- B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation* (Wiley, Chichester, 2003)
- H.G. Tanner, G.J. Pappas, V. Kumar, Leader-to-formation stability. *IEEE Trans. Robot. Autom.* **20**(3), 443–455 (2004)
- G. Tyson, A.T. Lindsay, S. Simpson, D. Hutchison, Improving wireless sensor network resilience with the INTERSECTION framework, in *Proceedings of the 2nd International Conference on Mobile Lightweight Wireless Systems, Critical Information Infrastructure Protection*, Barcelona, 2010
- E.R. Ulrich, D.J. Pines, J.S. Humbert, Pitch and heave control of robotic Samara micro air vehicles. *J. Aircr.* **47**(4), 1290–1299 (2010)
- Y. Wan, S. Roy, A. Saberi, Network design problems for controlling virus spread, in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, 2007, pp. 3925–3932
- Z. Wu, R. Wang, The consensus in multi-agent system with speed-optimized network. *Int. J. Modern Phys. B* **23**(10), 2339–2348 (2009)
- C.-D. Yang, W.-H. Liu, P.-W. Chang, H.-J. Weng, Decoupling control for hovering flight vehicle with parameter uncertainties, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, no. 1, 2006, pp. 1–13.
- D. Zelazo, M. Mesbahi, Edge agreement: graph-theoretic performance bounds and passivity analysis. *IEEE Trans. Autom. Control* **56**(3), 544–555 (2011)

Decentralized Multi-UAV Coalition Formation with Limited Communication Ranges

83

P. B. Sujit, J. G. Manathara, Debasish Ghose, and J. B. de Sousa

Contents

| | | |
|--------|--|------|
| 83.1 | Introduction | 2022 |
| 83.2 | Problem Formulation | 2024 |
| 83.2.1 | Problem Scenario and Approach | 2024 |
| 83.2.2 | Objective and Constraints | 2025 |
| 83.3 | Coalition Formation | 2029 |
| 83.3.1 | Polynomial Time Coalition Formation Algorithm (PTCFA) | 2031 |
| 83.3.2 | Target Prosecution Maneuvers | 2032 |
| 83.3.3 | Resource Removal | 2033 |
| 83.4 | Finding Potential Coalition Members Over a Dynamic Network | 2034 |
| 83.4.1 | Communication Protocol | 2034 |
| 83.5 | Simulation Results | 2037 |
| 83.5.1 | Example Scenario | 2037 |
| 83.5.2 | Effect of Varying Number of UAVs and Targets | 2041 |
| 83.5.3 | Effect of Varying Communication Range | 2043 |
| 83.5.4 | Effect of Hop Delay and Max-Hops | 2044 |
| 83.6 | Conclusions | 2046 |
| | References | 2047 |

Abstract

A team of unmanned aerial vehicles (UAVs) with limited communication and sensing ranges, communication delays, and finite resources are deployed to search and destroy targets in a given region. The targets can be static or moving.

P.B. Sujit (✉) • J.B. de Sousa

Department of Electrical and Computer Engineering, University of Porto, Porto, Portugal
e-mail: sujit@fe.up.pt; jtasso@fe.up.pt

J.G. Manathara • D. Ghose

Guidance, Control, and Decision Systems Laboratory (GCDL), Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India
e-mail: joel@aero.iisc.ernet.in; dghose@aero.iisc.ernet.in

To detect the targets, the UAVs perform a search task. When a target is detected, a coalition of UAVs, that has sufficient combined resources to destroy the target, must be formed. This coalition is to be determined, in a decentralized manner, by the UAVs communicating over a time and space varying communication network formed by UAVs in motion. In this chapter, a communication protocol to search for potential coalition members over a dynamic UAV network and a strategy for a UAV coalition to track and destroy moving targets is developed. The proposed mechanisms are validated through simulations. The effects of communication ranges and delays as well as the scalability of the proposed schemes with change in number of targets and UAVs are studied. The results show that it is beneficial to search for potential coalition members over a wide diameter in the UAV network when the communication delay is less, while it is more advantageous to determine a coalition from among the immediate neighbors in the presence of significant communication delay.

83.1 Introduction

Nowadays, several applications which are inaccessible and dangerous to human life are performed using unmanned aerial vehicles (UAVs). The capability of UAVs to deploy weapons, perform sensing operations, and act as communication relays has pushed the envelope of UAV applications covering missions like surveillance, search and rescue, border patrol, fire monitoring, target search and destroy, pipeline monitoring, communication relay, data mule, etc. Typically, these missions can be performed by a single UAV, in which case, the missions take more time to complete and the system is not robust to malfunctions. To enhance the mission performance in terms of robustness and mission completion time, deployment of multiple UAVs is desirable. The team of multiple UAVs must coordinate with each other to reap the benefits of a team. Therefore, there is a need to develop efficient cooperative algorithms to maximize the UAV team's performance.

Depending on the type of application, the coordination mechanism between the UAV team members varies. For example, in formation control problems, the coordination between UAVs is in maintaining constant distance and angle with respect to the neighbors. For surveillance problems, the cooperation between UAVs is to jointly cover a region while ensuring minimal area overlap. Yet another type of coordination is required in distributing tasks between multiple UAVs. The problem of task allocation between UAVs is combinatorial by nature. There are several types of task allocations depending on the task requirements and UAV capabilities. Gerkey and Mataric (2004) categorized multiple robot task allocation problems as single robot–single tasks (SR-ST), multiple robot–single tasks (MR-ST), single robot–multiple tasks (SR-MT), and multiple robot–multiple tasks (MR-MT). In this chapter, a MR-MT task allocation problem is considered where several UAVs need to search and destroy targets in a region.

In a task allocation problem, if the complete information about the targets (their locations and resources) and the UAV are known a priori, then the task allocation

problem can be solved using standard optimization techniques such as linear programming (LP), integer programming, and mixed-integer linear programming (MILP). However, this is rarely the case in a realistic mission scenario where the UAVs have limited sensing and communication ranges, delays in communication, kinematic constraints, and limited resources that deplete with use. In most cases, the target locations may not be known a priori. Moreover, the targets may be in motion and may require multiple resources to be destroyed. Determining task allocation solutions considering all the realistic constraints is highly complex and challenging. Hence, certain assumptions are made to obtain solutions under a given scenario. For example, several solutions have been developed with assumptions like all UAVs can communicate with each other, UAV resources are unlimited, and a single UAV alone can destroy a target. Nygard et al. (2001) developed a network flow optimization model that uses a linear programming method to allocate UAVs to targets. This approach assumes global communication between UAVs. The network flow model has been further extensively studied by Schumacher et al. (2003a,b) for wide area search munitions with variable path lengths and assignment with timing constraints. Chandler et al. (2003) explored various techniques like iterative network flow, auctions, LP, and MILP methods for multiple UAVs in an environment where the planning and allocation tasks are strongly coupled. Alighanbari et al. (2003) developed a task allocation algorithm for multiple UAVs using MILP to minimize the mission completion time while taking loitering time constraints into account. Schumacher et al. (2003a) developed a task allocation approach taking the arrival time constraints into account. They solve a single MILP problem that provides the optimal solution. Further, Shima and Schumacher (2009) considered the possibility of performing several tasks on the target before it can be destroyed completely. They developed a task allocation scheme using genetic algorithms for allocating tasks to multiple UAVs. On similar lines, Kingston and Schumacher (2005) develop a MILP-based solution to assign multiple UAVs to track and prosecute moving targets. Turra et al. (2004) developed a task allocation algorithm based on area assignment to consider moving targets.

When the constraint of limited communication range with delays is considered, several of the task allocation schemes, described above, fail to produce efficient results. The effect of communication delays on task allocation using an iterative network flow model is studied in Mitchell et al. (2003). Dionne and Rabbath (2007) developed an algorithm for decentralized task allocation in the presence of intermittent and asynchronous communication links, where the task allocation algorithm is formulated as a linear programming problem based on the information received from other agents. Lemaire et al. (2004) developed a task allocation scheme for multiple UAVs based on a contract net protocol taking limited communication constraints into account. To improve mission efficiency using auctions with limited communication, Sujit and Beard (2007) developed a sequential auction task allocation scheme where a single agent can be assigned to multiple targets.

There exists only a few works in the literature that consider the need to form a sub-team of UAVs to destroy a target. This sub-team is called a coalition, and the problem of determining a coalition is the coalition formation problem which is

NP-Hard (Shehory 1998). Kingston and Schumacher (2005) addressed this problem with the assumption of global communication. Vig and Adams (2006) developed a coalition scheme – RACHNA – where the tasks act as agents and perform the function of an auctioneer for gathering bids and thus determining the required coalition. Parker and Tang (2006) present a coalition formation scheme where a coalition leader robot broadcasts the existence of a task and other robots reply by providing their availability. The leader robot evaluates all possible coalitions and sends an accept decision to the robots that it considers suitable to be part of coalition. In the above approaches, the robots do not face a situation where the communication network can be broken during the coalition formation process since the robots can stop and form a static network until the coalition formation process is completed.

In a complete decentralized setup, a set of UAVs communicate with each other to form a coalition to engage a target. Due to limited communication ranges and constant motion, some of the UAVs may get disconnected from the network during the coalition formation process. If a disconnected member happens to be part of the determined coalition, then it cannot be informed about its requirement in the coalition. Hence, the coalition becomes invalid, and a new coalition has to be formed. Thus, determining a coalition in dynamic environments is very challenging. Scerri et al. (2005) determined potential team members using an algorithm that takes advantage of the “small world” property of large networks. Vanzin and Barber (2006) proposed a mechanism of finding potential members for a coalition based on the TCP protocol. However, both the approaches do not consider the possibility of communication delays and moving targets.

From the description of above articles, taking all the realistic constraints on UAVs and targets for a search and destroy mission into account is a complex task to accomplish. In this chapter, a novel mechanism to determine coalitions for moving targets taking all these constraints into account is developed. The only assumption is that the UAVs do not collide with each other while performing the mission. The chapter is organized as follows: In Sect. 83.2, the coalition formation problem is described. Section 83.3 gives a low complexity algorithm for coalition formation. The communication protocol employed in determining potential coalition members over a dynamic network is described in Sect. 83.4. The proposed algorithm is evaluated using Monte Carlo simulations in Sect. 83.5. Section 83.6 concludes this chapter with a brief summary.

83.2 Problem Formulation

83.2.1 Problem Scenario and Approach

Consider a search and destroy mission using N heterogeneous UAVs. The UAVs can carry n types of resources in limited numbers. Some of these resources are consumable, that is, the resources deplete with use, while others like sensors do not deplete. Each UAV is identified by its unique token number U_i , $i = 1, \dots, N$

assigned prior to the start of the mission and the resource capability vector of U_i is denoted by R^{U_i} of the form

$$R^{U_i} = (R_1^{U_i}, \dots, R_n^{U_i}), \quad i = 1, \dots, N$$

where $R_p^{U_i}$, $p = 1, \dots, n$ represents the number of type- p resources held by UAV U_i . Assume that there are M targets (some static and some in motion) whose positions and the resource requirements are unknown a priori. The target resource requirement vector is represented as

$$R^{T_j} = (R_1^{T_j}, \dots, R_m^{T_j}), \quad j = 1, \dots, M$$

where $R_q^{T_j}$, $q = 1, \dots, m$ represents the quantity of type- q resources required to destroy the target T_j and, without loss of generality, it can be assumed that $m \leq n$.

The UAVs have limited sensor range (r_s) and communication range (r_c). Typically, the communication ranges are higher than the sensing ranges; hence, it is assumed that $r_c > 2r_s$. This assumption ensures that multiple UAVs within the communication ranges of each other do not form multiple coalitions for the same target. The agents carry out a search task to detect targets. It is assumed that the agents use a heading search strategy where the UAV moves along a given heading until it crosses the boundary of the region of interest. At the boundary, it turns back into the operational region with some other heading direction. This process is continued until a target is found. When U_i detects a target T_j , as shown in Fig. 83.1, it assumes the role of a coalition leader and broadcasts the target resource requirement vector (R^{T_j}). It is assumed that the coalition leader can ascertain the type of resources required to destroy the target. The UAVs in the search region, that are within the communication range of U_i , possessing at least one of the required resources to destroy the target T_j will respond to U_i with their cost and resource capabilities (agent U_2 , U_5 , and U_6 in Fig. 83.1). Often, the resources of the immediate neighbors is not sufficient, in which case, the leader's broadcast must be passed on to the neighbors of the neighbors of U_i . The message passing between the UAVs occurs with delays. Thus, the coalition leader has to find potential members for its coalition over a dynamic ad hoc network subject to communication delays. Details on a robust mechanism to determine the coalition partners is described in Sect. 83.4.

83.2.2 Objective and Constraints

Assume that the UAV team of N agents takes τ time units to destroy all the targets. The mission time τ depends on the time taken to detect the targets, which in turn depends on the performance of the search strategy, and the time taken to

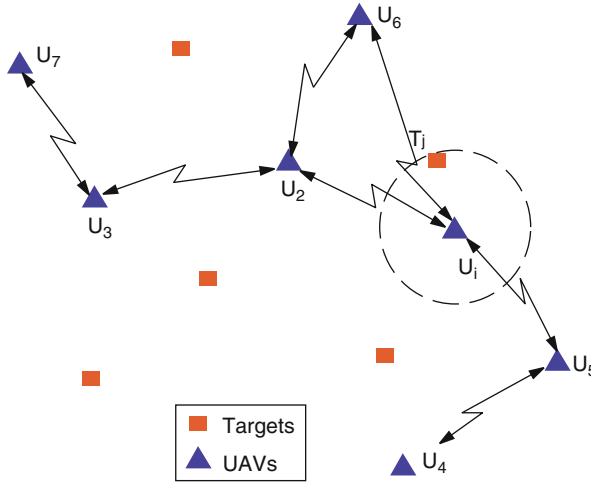


Fig. 83.1 The UAV U_i detects a target T_j and initiates a coalition formation process through a network formed by UAVs U_i and U_2 to U_7

form coalitions and destroy each detected target. The objective is to accomplish the mission in minimum time. That is, the global objective is

$$\min \tau \quad (83.1)$$

where the minimization is over all possible search strategies, prosecution sequences, and coalitions. Here, τ is bounded above by the sum of τ_s and $\sum_{j=1}^M \tau_c^{T_j}$, where τ_s is the search time and $\tau_c^{T_j}$ represents the time taken by the coalition to destroy the target T_j once it is detected. The mission time τ is not equal to but bounded by the above sum as some of the targets may be destroyed simultaneously. It is to be noted that the search time, τ_s , implicitly depends on the coalition formation strategies. This is because the choice of coalition will decide the UAVs that are free to carry out the search task. For example, for a given target, out of two coalitions with the same prosecution time, the coalition with smaller size will lead to more UAVs being available for search, which may ultimately lead to a reduction in the total search time. Also, note that the prosecution time, $\tau_c^{T_j}$, in turn, depends on the search strategy as the UAVs available for coalition formation, once T_j is detected, depends on the number of UAVs available within the communication range which, in turn, depends on the search deployment strategy. Thus, the global cost is a function of intertwined search and coalition formation strategies. The global objective can be achieved from Eq. (83.1) by (a) minimizing the time taken by the coalitions to destroy each target, that is,

$$\min \tau_c^{T_j}, \quad j = 1, \dots, M \quad (83.2)$$

where the minimization is over all possible coalitions, and (b) minimizing the size of coalitions formed to destroy it. Thus, the task of a coalition leader is to form a coalition such that (i) the target is destroyed in minimum time and (ii) the coalition contains minimum number of UAVs. The objective (i) (minimizing the time-to-attack the target) is a desirable quantity in achieving the objective given in Eq. (83.2), and the objective (ii) (which is to minimize coalition size) ensures that the UAVs distribute their search effort so that the targets are detected quickly and the mission is accomplished faster.

The coalition leader which forms a coalition to destroy a target is the UAV which detected the target. The role of the coalition leader U_i that detected target T_j is to form a coalition $\mathcal{C}(U_i, T_j)$, with combined resources $R^{\mathcal{C}(U_i, T_j)} \geq R^{T_j}$, to destroy T_j taking into account the objectives (i) and (ii). Let the set of all UAVs, the potential coalition members, that responded to the coalition formation request of U_i to destroy target T_j be denoted by $\mathcal{PC}(U_i, T_j)$. Let $\mathcal{D}(U_i, T_j) = \{D_U^{T_j} : U \in \mathcal{PC}(T_j, U_i)\}$ represent the set of earliest times of arrival for the UAVs in the set of potential coalition members, and $D_U^{T_j}$ denotes the earliest time of arrival (ETA) of the UAV U at the location of target T_j using Dubins curves (Dubins 1957). The ETA is used as the cost metric. To determine a coalition, the coalition leader solves the following optimization problem:

$$\arg \min_{\mathcal{C}'(U_i, T_j) \subseteq \mathcal{PC}(U_i, T_j)} \left(\max_{U \in \mathcal{C}'(U_i, T_j)} D_U^{T_j} \right) \quad (83.3)$$

$$\text{subject to } \left(\sum_{U \in \mathcal{C}'(U_i, T_j)} R_p^U \right) \geq R_p^{T_j}, \quad \text{for all } p = 1, \dots, m \quad (83.4)$$

The set $\mathcal{C}(U_i, T_j)$, the solution to the above optimization problem, represents the smallest coalition which satisfies Eq. (83.4). That is, if $\mathcal{C}''(U_i, T_j) \subseteq \mathcal{PC}(U_i, T_j)$ is another set which is a solution to Eqs. (83.3) and (83.4), then $|\mathcal{C}(U_i, T_j)| \leq |\mathcal{C}''(U_i, T_j)|$. The set obtained as a solution to the above optimization problem satisfies the objectives (i) and (ii). It is to be noted that the optimum value of the objective function in Eq. (83.3) is the earliest time at which all the members of the coalition, $\mathcal{C}(U_i, T_j)$, can arrive at the target T_j , which is equal to $\tau_c^{T_j}$, the time taken by the coalition to destroy T_j . Therefore, solving the optimization problem in Eqs. (83.3) and (83.4) for every target will minimize the objective function given in Eq. (83.2). The minimum time (objective (i)) for all the UAVs in a coalition to arrive at the target is determined by that UAV which has the highest ETA to the target. Since both the smallest coalition and the coalition that can destroy the target in minimum time needs to be determined simultaneously, the objective function is coupled. This coupling makes the optimization problem nontrivial and unsuitable for solution using standard techniques. In this chapter, previously developed coalition formation algorithms are used to determine coalitions that

are close to optimal and consume low overhead (Manathara et al. 2011). For completeness, a brief description of the algorithms is provided in Sect. 83.3.

The coalition leader forms the coalition in a decentralized manner which has significant advantages over centralized solution concepts. The coalition formation algorithms determine a set of agents that need to attack the target. The coalition members arrive at the target using Dubin's shortest path and deploy their resources at the target location. When all the agents in the target coalition deploy their resources, the target is assumed to be destroyed. However, in the case of moving targets, the target needs to be tracked and the target location information needs to broadcast to other agents in the coalition. Once the target is in the sensing range of an agent, the agent deploys the resource and tracks it until it is within the sensing range of the next coalition member. The attacking and tracking of the target with handshaking mechanism is described in Sect. 83.3.

The UAVs are subjected to kinematic constraints preventing instantaneous course changes. Assume that the autopilots of the UAVs hold a constant altitude and ground speed. The kinematics of the i th UAV is therefore modeled using first order kinematics as

$$\begin{aligned}\dot{x}_i &= v_i \cos \psi_i, \\ \dot{y}_i &= v_i \sin \psi_i, \\ \dot{\psi}_i &= k(\psi_i^d - \psi_i), \quad i = 1, \dots, N\end{aligned}\tag{83.5}$$

where x_i and y_i are the UAV location, ψ_i is its heading, ψ_i^d is the desired heading, v_i is the ground speed, and k is autopilot gain. The heading rate to be constrained within bounds. That is,

$$-\omega_{\max} \leq \dot{\psi}_i \leq \omega_{\max}, \quad i = 1, \dots, N\tag{83.6}$$

Some of the targets are assumed may be moving and can change their heading randomly, but this change is bounded by a maximum turn rate. The kinematic model of the target T_j is

$$\begin{aligned}\dot{x}_j^T &= v_j^T \cos \psi_j^T \\ \dot{y}_j^T &= v_j^T \sin \psi_j^T \\ \dot{\psi}_j^T &= \beta \dot{\psi}_{\max}^T, j = 1, \dots, M\end{aligned}\tag{83.7}$$

where v_j^T is the velocity of the target T_j , ψ_j^T is its heading, $\dot{\psi}_{\max}^T$ is the maximum turn rate, and β takes values ± 1 randomly for intervals of random durations as time evolves. The velocities of all the targets are assumed to be less than the velocities of any of the UAVs. After a target is attacked by a UAV, if it is not completely destroyed, then the target velocity is reduced by a random factor. This reduction

is to simulate the fact that after the resources are deployed by a UAV, the target capability reduces. Hence, the new target velocity after prosecution by a UAV is given by \hat{v}_j^T which is strictly less than v_j^T .

83.3 Coalition Formation

Once a UAV detects a target, it acts depending on its resource state and the resource requirement of the target. Figure 83.2 shows two different states (S_1 and S_2) and how the UAV makes a coalition. The sequence of actions carried out during these states are described below.

Coalition Leader Has All the Required Capabilities (S1) In this case, UAV U_i detects target T_j that requires R^{T_j} resources and $R_p^{U_i} \geq R_p^{T_j}$ for all $p = 1, \dots, m$. If U_i is not already a part of another coalition, then U_i would attack target T_j without requesting a coalition with other UAVs. This is a special case of coalition formation where only one UAV comprises of the coalition. The UAV determines the route to travel using Dubins curves (Dubins 1957) and then proceeds toward the target as shown by the state S_1 in Fig. 83.2.

Coalition Leader Has Partial Resources or No resources (S2) U_i assumes the role of coalition leader and broadcasts the information about the target, that is, its location and required resources, to the other UAVs. The UAVs that have at least one of the types of required resources will send their cost to arrive at the target (ETA) and their available resource vector. The coalition leader considers all the responses and determines whether a coalition can be formed. If a coalition cannot be formed, then it sends a “discard coalition” broadcast as shown in Fig. 83.2; otherwise, it will form a coalition and broadcast the coalition information. The rejected agents will discard their potential interest in the coalition and continues to perform their search task. During this process, the potential members who announced their availability for coalition will be in a wait state to receive information from the coalition leader. The complete sequence of actions that happen during the coalition formation for a target by the coalition leader as well as by other UAVs is shown in Fig. 83.2.

The coalition leader has to solve the optimization problem described by Eqs. (83.3) and (83.4) to determine the coalition members. Previously, in Manathara et al. (2011), two coalition formation algorithms were developed: (i) polynomial time coalition formation algorithm (PTCFA – which is suboptimal) and (ii) optimal coalition formation algorithm (OCFA). Both the algorithms use a two-stage mechanism to produce solutions that have low computational complexity. In Manathara et al. (2011), it is shown that PTCFA performs similar to OCFA with lower computational complexity. In the simulations presented later, PTCFA was used and hence PTCFA is described briefly below.

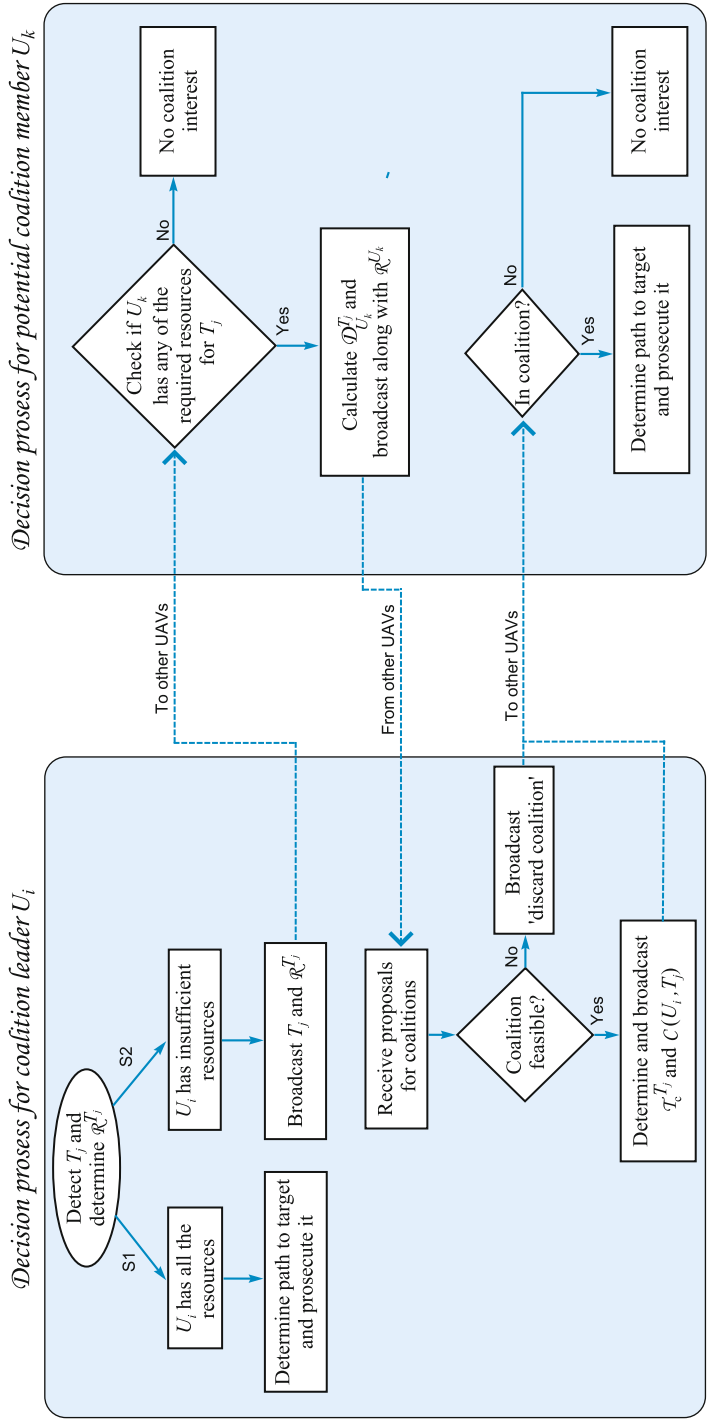


Fig. 83.2 Sequence of events during coalition formation

83.3.1 Polynomial Time Coalition Formation Algorithm (PTCFA)

Determining the smallest coalition that would successfully destroy the target in minimum time can be accomplished in two stages. In the first stage, a set of UAVs with required total resources that can achieve the minimum time requirement is determined. Then the obtained set of UAVs are pruned to achieve the minimum member coalition in the second stage. The process to achieve this task is given in Algorithms 1 and 2.

Assume that UAV U_i is the coalition leader and detected a target T_j that requires R^{T_j} resources. Algorithm 1 begins by initializing the coalition set to empty and the coalition resources to zero (line 1). First, the responses of the potential coalition members are sorted in the ascending order of their ETA to target (line 3). The algorithm takes one UAV (U_k) at a time (line 5) from the ordered list, includes U_k in the coalition $\mathcal{C}(U_i, T_j)$ (line 6), and updates the coalition resource vector $R^{\mathcal{C}(U_i, T_j)}$ (line 7) and the coalition time $\tau_c^{T_j}$ (line 8). Then it checks whether the target resource constraint is met (line 9). If the constraint is not met, then the process of including the next agent and its resources and verifying the resource constraint continues until the target resource constraint is met or the UAVs in the sorted potential coalition member set is exhausted in which case a feasible coalition is not formed (line 18). Once the resource constraint is met, the algorithm returns the feasible coalition $\mathcal{C}(U_i, T_j)$ and the coalition time $\tau_c^{T_j}$ (line 16).

Algorithm 1 First stage of the PTCFA

Input: Potential coalition members $\mathcal{PC}(U_i, T_j)$ and their ETAs $\mathcal{D}(U_i, T_j)$

Output: Coalition $\mathcal{C}(U_i, T_j)$ and coalition time $\tau_c^{T_j}$

```

1:  $\mathcal{C}(U_i, T_j) = \emptyset$  and  $R^{\mathcal{C}(U_i, T_j)} = \mathbf{0}$ 
2: Stage 1:
3:  $[\mathcal{D}_{sorted}, \mathcal{PC}_{sorted}] = \text{sortAscending}(\mathcal{D}(U_i, T_j), \mathcal{PC}(U_i, T_j))$ 
4: for  $l = 1$  to  $|\mathcal{PC}(U_i, T_j)|$  do
5:    $U_k \leftarrow \mathcal{PC}_{sorted}(l)$ 
6:    $\mathcal{C}(U_i, T_j) \leftarrow \text{append } U_k$ 
7:    $R^{\mathcal{C}(U_i, T_j)} \leftarrow R^{\mathcal{C}(U_i, T_j)} + R^{U_k}$ 
8:    $\tau_c^{T_j} \leftarrow \mathcal{D}_{sorted}(l)$ 
9:   if  $R_p^{\mathcal{C}(U_i, T_j)} \geq R_p^{T_j}$ , for all  $p$  then
10:     break
11:   else
12:     continue
13:   end if
14: end for
15: if  $R_p^{\mathcal{C}(U_i, T_j)} \geq R_p^{T_j}$ , for all  $p$  then
16:   return  $\mathcal{C}(U_i, T_j)$  and  $\tau_c^{T_j}$ 
17: else
18:   return No feasible coalitions
19: end if
```

Algorithm 2 Second stage of the PTCFA

Input: Minimum time coalition $\mathcal{C}(U_i, T_j)$ from Algorithm 1

Output: Pruned coalition $\mathcal{C}(U_i, T_j)$

```

1: Stage 2:
2: for  $l = 1 : |\mathcal{C}(U_i, T_j)|$  do
3:    $U_k \leftarrow \mathcal{C}(U_i, T_j)(l)$ 
4:    $\bar{R}^{C(U_i, T_j)} \leftarrow R^{C(U_i, T_j)} - R^{U_k}$ 
5:   if  $\bar{R}^{C(U_i, T_j)} \geq R_p^{T_j}$ , for all  $p$  then
6:      $\mathcal{C}(U_i, T_j) \leftarrow \mathcal{C}(U_i, T_j) \setminus U_k$ 
7:      $R^{C(U_i, T_j)} \leftarrow \bar{R}^{C(U_i, T_j)}$ 
8:   end if
9: end for
10: return  $\mathcal{C}(U_i, T_j)$ 

```

Algorithm 1 will always ensure a coalition with minimum time to target. This is proved in Manathara et al. (2011). Once the minimum time coalition is formed by Algorithm 1, then those members in the minimum time coalition whose resources are not required to form a coalition with smaller size must be pruned. This process is carried out by using Algorithm 2. The algorithm checks for each UAV $U_k \in \mathcal{C}(U_i, T_j)$ in the coalition formed by Algorithm 1, whether the resources of U_k are required for the coalition or not by removing its resources from $R^{C(U_i, T_j)}$ (lines 4 and 5). If the resources of U_k are not required, then the UAV U_k is removed from $\mathcal{C}(U_i, T_j)$ (line 7) and its resources are deducted from $R^{C(U_i, T_j)}$ (line 7). This process is carried out for all the UAVs $U_k \in \mathcal{C}(U_i, T_j)$. The Algorithms 1 and 2 together form the two-stage polynomial time coalition formation algorithm (PTCFA).

83.3.2 Target Prosecution Maneuvers

The environment contains static and moving targets. The procedure involved in prosecuting a static target is different from that of a maneuvering target.

83.3.2.1 Procedure to Prosecute a Static Target

In this case, the coalition strikes the target based on their ETA. The UAV with the smallest ETA attacks the target first and then continues to perform other tasks. After that, the UAV with the second lowest ETA will attack. This process continues until the last UAV in the coalition strikes the target and destroys it completely. The sequence in which the coalition members arrive at the target location is based on the ascending order of their ETAs to the target.

83.3.2.2 Procedure to Prosecute a Maneuvering Target

Unlike the static target, destroying a moving target with random maneuvers is difficult and requires cooperation between the coalition members. Since the target

moves in random directions, an attack and track mechanism is used. Once the target is attacked, depending on the resources deployed, it is either completely destroyed or its velocity is reduced by a random factor which represents a slower moving target after suffering some casualty. The process involved in destroying a maneuvering target is described below.

Assume that a coalition consisting of $N' \leq N$ is formed. In this case, the coalition leader continues to track the target until it directly communicates with the first UAV of the formed coalition. The coalition leader updates the information of the target to the approaching coalition member. Based on this information, the coalition member updates its heading continuously until it detects the target. Once the coalition member detects the target, the coalition leader relinquishes its role to track and monitor the target and switches itself to a search task. This process of handshaking is required because the coalition member does not know the target location accurately because of the target's random heading behavior.

Once the coalition member detects the target, it tracks the target and strikes. When resources are deployed on the target, its velocity is reduced randomly. After deploying the resources, the coalition members continue to track the target. When the next coalition member is within the communication range, the coalition member currently tracking the target updates the information of the target so that the arriving member can update its direction. When the second agent detects the target, the first relinquishes its tracking role to perform other tasks. This process continues until all the coalition members strike the target, finally destroying it.

83.3.3 Resource Removal

When a UAV attacks a target, it deploys some of its resources. The quantity of resources deployed depends on the quantity of resources required by the target at that instant in time. At any time t , if the required target resources are R^{T_j} and the p th resource is $R_p^{T_j}$, then the new resources required by the target after UAV resource deployment is $R_p^{T_j} = \max(R_p^{T_j} - R_p^{U_i}, 0)$. That is, the UAVs use a greedy approach, where the UAVs deploy all the available required resources when they prosecute the target. There is no exchange of the resources between the agents.

Although the coalition formation process developed above is simple, when implemented over a dynamic network, like the one consider in this chapter, highly nontrivial situations arise which need to be addressed effectively. For example, an intermediate UAV can fail to communicate a decision to a required UAV or by the time the coalition decision is declared, a potential coalition member can be out of communication range. To address such situations, there is a need to design behaviors that will either enable the agent to make its decision smoothly whenever such events occur, or avoid these situations altogether. The second strategy is adopted here and explained in detail below.

83.4 Finding Potential Coalition Members Over a Dynamic Network

The UAVs need to form coalitions over a time-varying network, and the success of forming a feasible coalition depends heavily on the communication protocol employed. The coalition formation process, as described before, involves the following communications (refer Fig. 83.2): (i) broadcast of the coalition leader seeking the participation of other UAVs in the coalition to be formed to destroy a target that it detected, (ii) message from potential coalition members about their willingness to participate in that coalition, and (iii) broadcast from coalition leader about the success/failure of the coalition formation process. When the UAVs have limited communication ranges, these communications have to be done over a time-varying network. This problem can be mitigated by finding a subnetwork, that is invariant for the time period of the coalition formation process, over which the communications happen.

83.4.1 Communication Protocol

In the network, a UAV will receive information from another UAV (say, a coalition leader) through intermediate UAVs called relay nodes. For example, in Fig. 83.1, the UAV U_7 receives information about target T_j from U_i through the information path $U_i \rightarrow U_2 \rightarrow U_3 \rightarrow U_7$, where the UAVs U_2 and U_3 act as the relay nodes rebroadcasting the message from U_i . This method of communication between wireless nodes is referred to in the literature as “flooding” (Obraczka et al. 2001; Williams and Camp 2002). To mimic a real-world wireless network, a delay is associated with each message hop – transmission of message from one UAV to another – called the hop delay denoted by Δ_{hop} . The hop delay includes the queuing delay, the message processing time at a relay node, and the propagation delay, which are the usual delays associated with a standard wireless network. The wireless communication standard – the IEEE 802.11 MAC (medium access control) protocol – handles broadcast packets by sending it to all the neighbors as soon as the carrier is sensed free (IEEE-SA Standards Board 2007). However, it does not use any collision detection to ensure the successful delivery of the broadcast packets. There are some schemes (Alagar et al. 1995; Tang and Gerla 2001; Chen and Huang 2005) that ensure reliable MAC designs for broadcast transmission but add overhead to the MAC layer. The UAVs are assumed to use one such protocol and the hop delay will include the associated overheads. The value of Δ_{hop} can be considered a constant taking the value of the estimated maximum possible delay in the network (Lamport 1984). For the above example, the UAV U_7 would receive the message from U_i within $3\Delta_{\text{hop}}$ seconds as the message from U_i has to make 3 hops to reach U_7 .

A message broadcast by a UAV flows through the network as it is rebroadcast by relay nodes. To prevent the message from floating around in the network indefinitely, thus reducing the bandwidth of the network, the notion of time-to-live (TTL) is used for a message – a notion used in packet switching (Clark 2003) in the Internet

Protocol. The TTL of a message is considered to be the maximum number of hops that it is allowed to make before it is abandoned (not rebroadcast) by the nodes in the network and specify it in terms of maximum number of hops (H_{\max}) a message can make in a network. The TTL concept is implemented as follows. Each message has in its body a hop counter, H , which is initially set to H_{\max} . Whenever a relay UAV rebroadcasts the message, H is reset to $H - 1$. If $H = 0$, then the message is not rebroadcast by a relay UAV and thus gets abandoned.

A UAV that acts as a relay node keeps a log of every message that it rebroadcasts. If it receives a message that is already in the log, it will not be rebroadcast. This is to prevent clogging the network with messages that are already broadcast. For example, in Fig. 83.1, U_2 gets message from U_i which it broadcasts. It will also receive the same message as rebroadcast by U_6 later. However, this will be ignored by U_2 as it has already broadcast that message. The log of a broadcast message is kept until the TTL of the message.

To determine a coalition, the coalition leader requires the following information from the potential members: (a) earliest time of arrival (ETA) at the target and (b) resources available for target prosecution. However, since some of the targets are nonstationary and the network has propagation delays, there is an ambiguity about the procedure for calculation of ETA. In order to determine ETA in a consistent manner, inspiration is drawn from the way contracts are carried out for tasks in real life. An excellent example of this can be the task of “call for papers” for a conference. The conference committee will broadcast the task with the submission deadline, the deadline for acceptance/rejection notification, and the final submission deadline. Thus, there exists deadlines for every part of the process to accomplish the task. Similarly, the coalition leader uses intermediate deadlines to receive the information and to form coalitions. The coalition leader will broadcast the availability of the target along with the deadline for proposal submission, the deadline to receive decisions, and the deadline to destroy the target. The coalition leader will also broadcast the exact time at which the coalition members are required to start the execution maneuver. The time at which a coalition member is expected to start its prosecution maneuver is called the go-ahead time, denoted by $\tau_G^{T_j}$. A potential coalition member uses this information to estimate its ETA to the target.

To determine coalition over a dynamic network, a set of agents whose communication graph exists for the coalition formation process period must be found. This is achieved as follows. A UAV agrees to be a relay node for a message it received only if it is going to be in the communication range of the UAV from which it received the message for the entire coalition formation period. Every message contains the information on the latest time by which the coalition formation process should end. For a coalition formation process for target T_j , this time is the same as the go-ahead time $\tau_G^{T_j}$. All the UAVs are assumed to have synchronized clocks. A potential coalition member shows interest in a coalition only if it remains in the communication range of the UAV from which it received the message until $\tau_G^{T_j}$. This behavior ensures a subnetwork that is invariant over the period of coalition formation.

The go-ahead time is determined by the coalition leader as a time which is greater than the time required for the entire coalition formation process. If H_{\max} is the maximum number of allowed hops for a message with a hop delay of Δ_{hop} associated with each hop, then the coalition request, the response to this request, and the message on coalition membership together take at most $3H_{\max}\Delta_{\text{hop}}$ seconds to propagate over the network. Each potential coalition member is given a time window of Δ_w seconds before they respond to a coalition participation request from a coalition leader. This will give the potential coalition members time to consolidate the requests and respond to the appropriate coalition leader (coalition leader with highest token number). The coalition leader takes a time of Δ_c seconds to form a feasible coalition. Thus, the total coalition process is upper bounded by $3H_{\max}\Delta_{\text{hop}} + \Delta_w + \Delta_c$. The coalition leader, therefore, can choose the go-ahead time, $\tau_G^{T_j}$, as $3H_{\max}\Delta_{\text{hop}} + \Delta_w + \Delta_c$.

From the above discussion, it is clear that the time for proposal submission by potential coalition members is $2H_{\max}\Delta_{\text{hop}} + \Delta_w$ seconds after the dispatch of the coalition participation request. A potential coalition member U_k , if it has any resource required to prosecute the target, estimates its position at time $\tau_G^{T_j}$ called its go-ahead location, denoted by $G_{U_k}^{T_j}$. It then broadcasts its ETA to target from $G_{U_k}^{T_j}$ and its resource capability vector. The coalition leader determines a feasible coalition using PTCFA and the message is broadcast. The UAVs in coalition reach their respective go-ahead locations ($G_{U_k}^{T_j}$) and take a Dubins path to arrive at target T_j . On the other hand, the rejected UAVs continue their previous task. If a coalition leader cannot form a feasible coalition for a target, it waits for a specified time (a value of 5 s is used in simulations) before a coalition formation request is broadcast again.

Coalition leader U_i broadcasts the request in the form

$$\langle U_i, T_j, \tau_G^{T_j}, H \rangle$$

where $\tau_G^{T_j}$ is the go-ahead time, T_j represents the location of the target, its heading and speed at $\tau_G^{T_j}$, and H is the hop counter which is set to H_{\max} by the coalition leader. The presence of H_{\max} in the message will help the potential coalition member to set the initial value of the hop counter H for the proposal message that it is sending back to a coalition leader. The information on $\tau_G^{T_j}$, apart from being useful for a potential coalition member to calculate its ETA to target in a consistent way, enables the intermediate UAVs to decide whether to be relay nodes and potential coalition members and tells the relay nodes when to delete the message from the log to save space.

A typical response of a potential coalition member, U_k , is of the form

$$\langle U_k, U_i, R^{U_k}, \tau_{U_k}^{T_j}, \tau_G^{T_j}, H \rangle$$

where R^{U_k} is the resource vector of U_k , $\tau_{U_k}^{T_j}$ is its ETA to target, and H is the hop counter for the message. Similar to the case of broadcast from the coalition leader, the $\tau_G^{T_j}$ information will help a receiving UAV to decide whether to be a relay node.

The decision process and the communication protocol during a coalition formation process, within a group of UAVs with limited communication ranges executing a search and destroy task, is given in Fig. 83.3. Although, the coalition leader, the relay nodes, and the potential coalition members are shown separately in the figure for clarity, every UAV, including the coalition leader, acts as both a relay node as well as a potential coalition member.

83.5 Simulation Results

In this section, the performance of the coalition formation algorithm for UAVs performing a search and destroy mission is studied via simulations. Initially, an example is presented to show how the coalition formation is carried out and then Monte Carlo simulation results are presented. Through Monte Carlo simulations, the effects of the network protocol parameters like maximum number of allowed hops (H_{\max}) and hop delay Δ_{hop} on the performance of the search and destroy mission are studied in detail.

For all the simulations presented, the UAVs are assumed to have a maximum turn rate of 20 deg/s. The following behavior for a maneuvering target is considered in simulations: it has an initial velocity, the magnitude of which is less than that of all UAVs. It moves forward for a random period of time bounded above by 5 s and then takes a turn randomly either to left or right with a turn radius of 100 m for a random time interval in the range of 0–3 s. This process is repeated several times till the target gets destroyed or till the end of the simulation. After being hit by a UAV, the velocity of a target reduces by a random amount and it does a turn maneuver either to left or right for a random time duration which is less than 3 s. A UAV that is tracking a maneuvering target communicates with the other UAVs, at every 1 s interval, to provide the information of the updated position of the target.

83.5.1 Example Scenario

Consider a scenario with five UAVs and two moving targets in a bounded region of $2,000 \times 2,000$ m as shown in Fig. 83.4a. The target resource requirements are $R^{T_1} = (1, 2, 1)$ and $R^{T_2} = (2, 1, 3)$. The targets have constant speeds of 13.06 and 17.2 m/s with initial headings of 335° and 240° , respectively. The resource capabilities of the UAVs are $R^{U_1} = (1, 0, 1)$, $R^{U_2} = (2, 1, 1)$, $R^{U_3} = (1, 2, 0)$, $R^{U_4} = (1, 2, 1)$, and $R^{U_5} = (0, 1, 1)$. The UAVs have constant speeds of 50 m/s. Figure 83.4a shows the initial placement of the UAVs and the targets. The circles around the UAVs represent the sensor ranges of the vehicles, and the lines

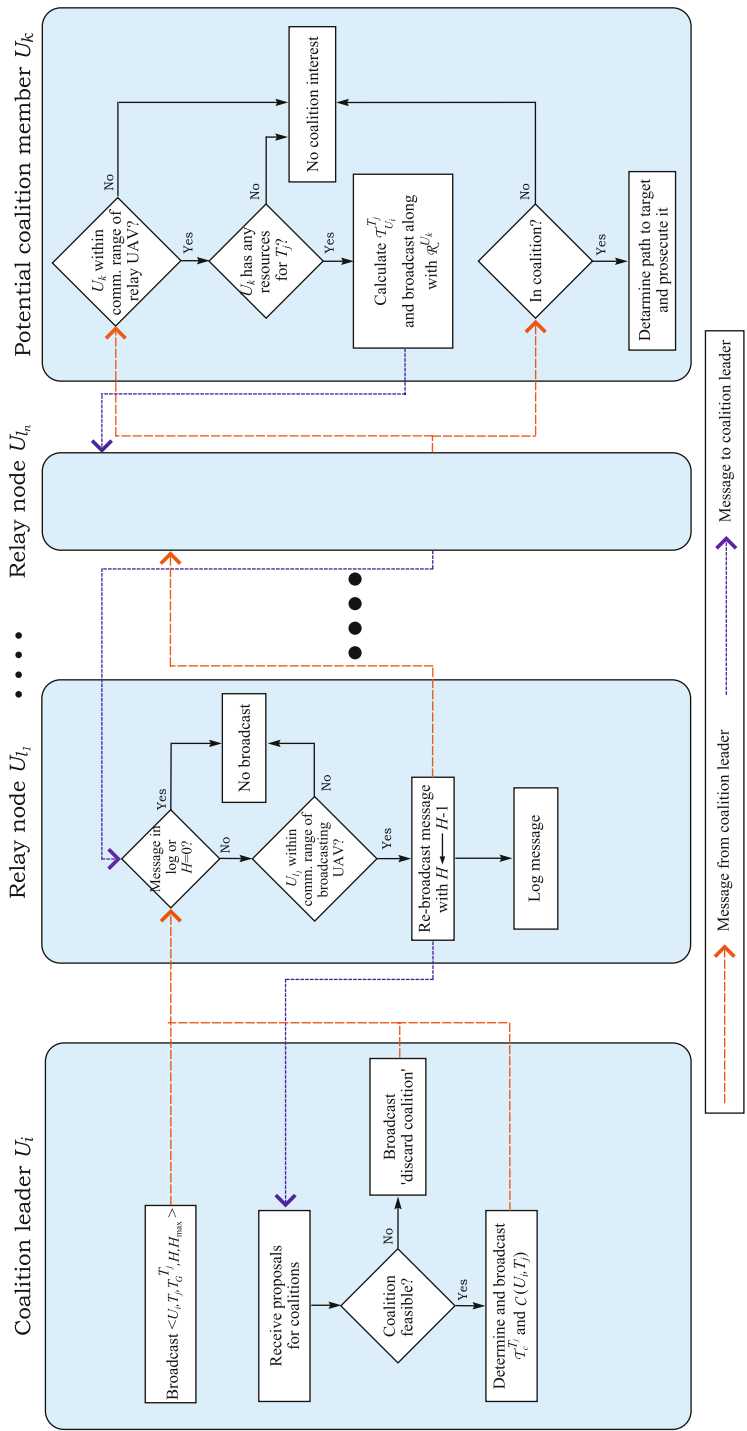


Fig. 83.3 Decision process and communication protocol for coalition formation of UAVs with limited communication ranges

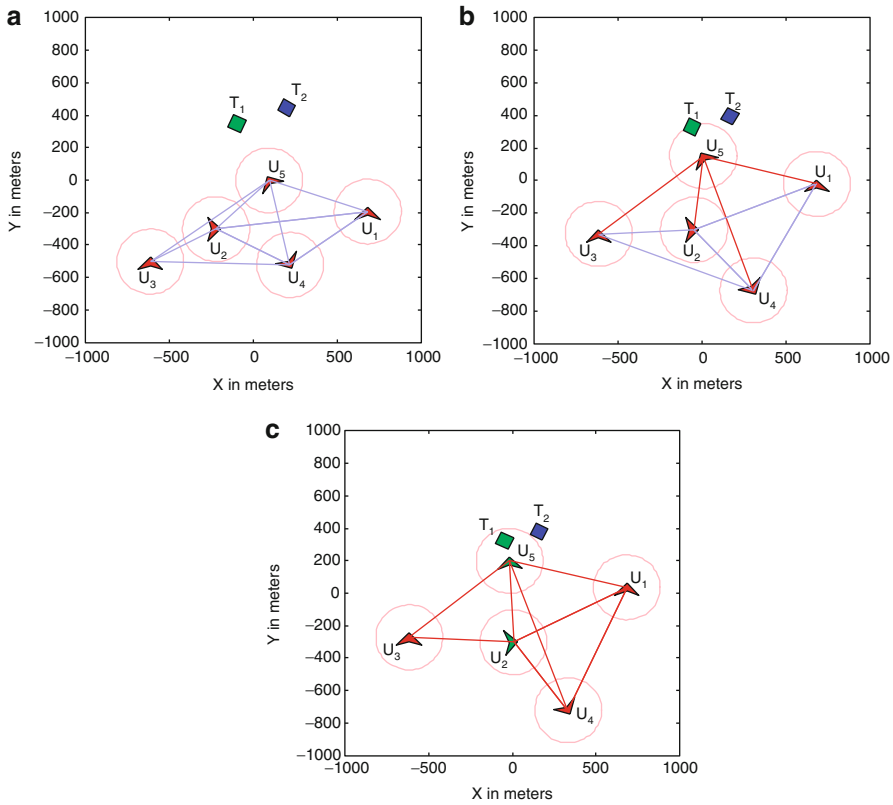


Fig. 83.4 (a) Initial positions of the UAVs and the targets. (b) UAV U_5 detects target T_1 at time $t = 3.6$ s and broadcasts the proposal. (c) A coalition consisting of U_5 and U_2 is formed to destroy target T_1 at time $t = 4.7$ s

connecting the vehicles denote the communication links between them. For this example, the sensor range is 200 m and the communication range is 1,000 m for all UAVs. The maximum number of allowed hops for a message is 2 and $\Delta_{\text{hop}} = 0.1$ s, $\Delta_w = 2\Delta_{\text{hop}}$, and $\Delta_c = 0.3$ s.

The sequence of events during the simulation is as follows. At time $t = 3.6$ s, U_5 detects T_1 (as shown in Fig. 83.4b), and since it does not have enough resources, it broadcasts the proposal for coalition. The UAV U_5 sets the go-ahead time as $\tau_G^{T_1} = 4.7$ s. Figure 83.4b shows all the UAVs are in the communication range of U_5 . The coalition leader U_5 determines a coalition $\mathcal{C}(U_5, T_1) = \{U_5, U_2\}$ at $t = 4.1$ s and sends the decision at $t = 4.2$ s. However, the target execution maneuver starts at $t = 4.7$ s, which is the same as $\tau_G^{T_1}$, as shown in Fig. 83.4c. The agent U_5 will direct its motion toward T_1 and periodically sends the position information of T_1 to U_2 , for aiding U_2 to close in on T_1 .

As the mission progresses, at $t = 5.3$ s, U_5 detects T_2 and sends out a coalition request. Since U_5 and U_2 have not yet completed their task of executing T_1 , they

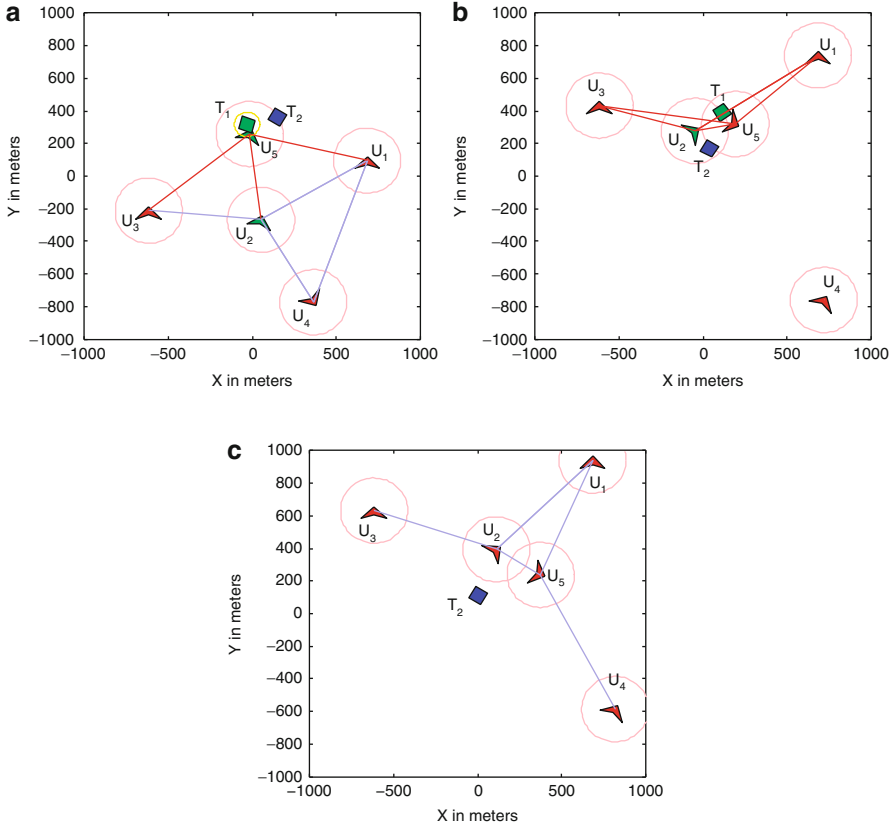


Fig. 83.5 (a) U_5 deploys resources on T_1 and continues to track it. U_5 also detects T_2 and tries to form an unsuccessful coalition due to lack of sufficient resources. (b) U_2 detects T_1 in the sensor range and U_5 releases itself from tracking T_1 . (c) U_2 deploys its resources on T_1 and destroys it completely

cannot yet be the part of the coalition formed to destroy T_2 . The combined resources of the UAVs U_1 , U_3 , and U_4 are $(3, 4, 2)$ and the required resource to destroy T_2 is $R^{T_2} = (2, 1, 3)$. As the combined quantity of the third type of resource of the UAVs available for coalition is less than the target requirement, a coalition cannot be formed. At time $t = 6$ s, U_5 deploys its resources on T_1 (as shown in Fig. 83.5a) and continues to track T_1 until T_1 is in the sensor range of U_2 , as shown in Fig. 83.5b. At time $t = 21.4$ s, U_2 deploys its resources on T_1 to destroy the target completely as shown in Fig. 83.5c.

By the time T_1 is destroyed, T_2 has moved away from the sensor range of all the agents as seen in Fig. 83.5c and the agents continue to search for T_2 . At time $t = 59.6$ s, U_4 detects T_2 and initiates a coalition formation proposal as shown in Fig. 83.6a. The coalition leader U_4 determines a coalition $\mathcal{C}(U_4, T_2) = \{U_4, U_1, U_2\}$ at time $t = 60.7$ s (as shown in Fig. 83.6b). The agent U_4 attacks T_2 and deploys its

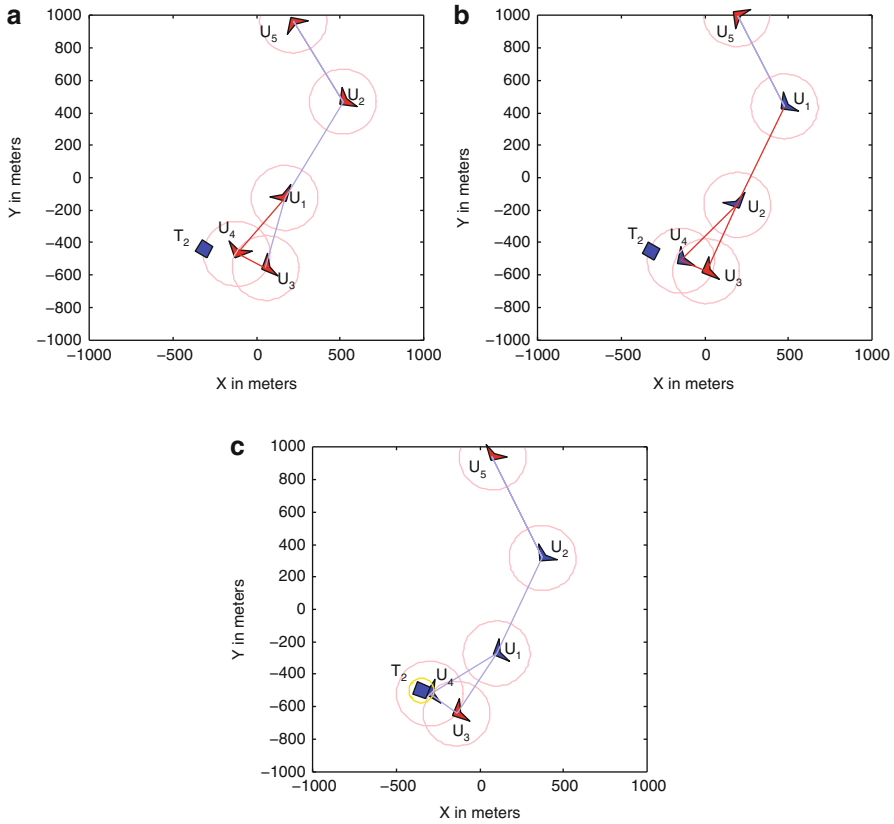


Fig. 83.6 (a) U_4 detects T_2 at $t = 59.6$ s. (b) U_4 forms a coalition $\{U_4, U_1, U_2\}$ for T_2 at time $t = 60.7$ s. (c) U_4 attacks T_2 at $t = 65.3$ s and continues to track it

resources at $t = 65.3$ s (as shown in Fig. 83.6c) and continues to track T_2 . Once T_2 is in U_1 's sensor range, U_4 releases the tracking tasks and U_1 strikes T_2 at $t = 70.5$ s (as shown in Fig. 83.7a). After U_1 's strike, U_1 tracks the target until U_2 detects the target in its sensor range as shown in Fig. 83.7b. Then U_2 tracks the target and strikes it to accomplish the mission as shown in Fig. 83.7c at time $t = 81.3$ s.

The above sequence of events for the sample scenario shows how the agents determine coalitions and the process involved in tracking randomly maneuvering targets and destroying them.

83.5.2 Effect of Varying Number of UAVs and Targets

The performance of the coalition formation algorithm with increase in the number of UAVs used for the search and destroy mission is studied in this section.

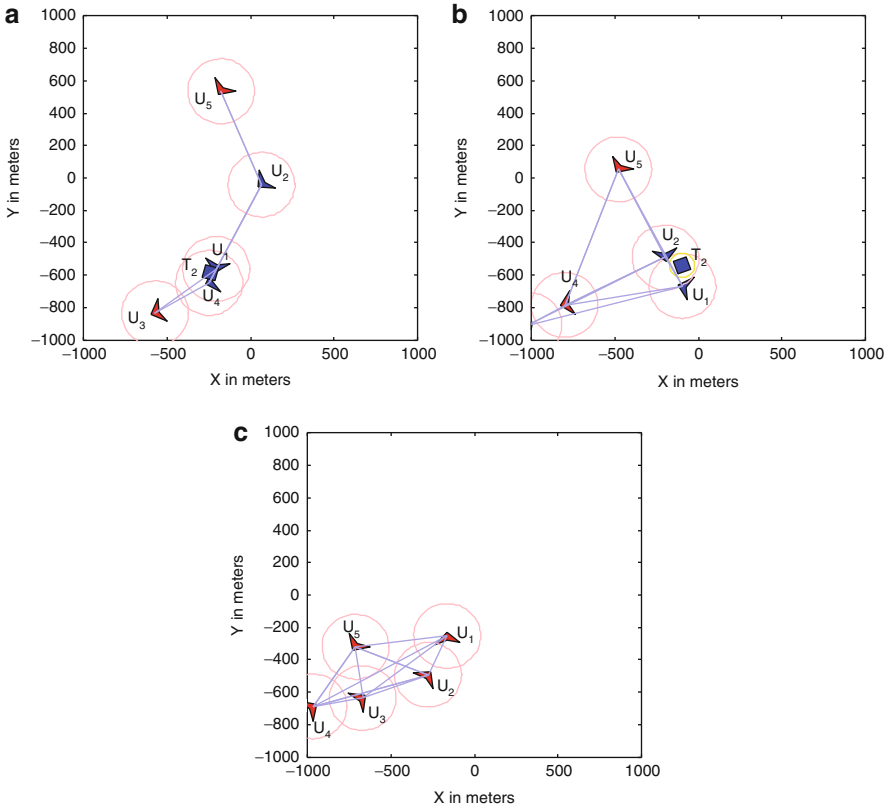


Fig. 83.7 (a) U_1 strikes T_2 at $t = 70.5$ s. (b) T_2 is in the sensor range of U_2 . (c) U_2 strikes T_2 at time $t = 81.3$ s to destroy it and complete the mission

The performance is measured in terms of the time required for mission completion. The parameters used for the simulation are as follows. A bounded region of $1,000 \times 1,000$ m is considered for simulations. The sensor range is 150 m and the communication range is 300 m. The maximum number of allowed hops for a message is 3 and $\Delta_{\text{hop}} = 0.1$ s, $\Delta_w = 0.2$ s, and $\Delta_c = 0.3$ s. The UAVs have fixed but different velocities in the range 10–20 m/s randomly chosen at the start of each simulation. To start with, every target has a fixed velocity randomly chosen between 0 and 10 m/s. The target behavior is similar to that described in the beginning of this section. Also, if the target is near the boundary of the search region, the UAV that detected it will track the target until it comes within the region of interest and then a coalition request is made. If a UAV that is part of a coalition detects a target on its way to prosecution of some other target, then it broadcasts this information to nearby UAVs. Those UAVs that receive this message moves to the region where the target was detected. This leads to a kind of directed search that will aid a quicker prosecution of targets which otherwise would have to be detected again.

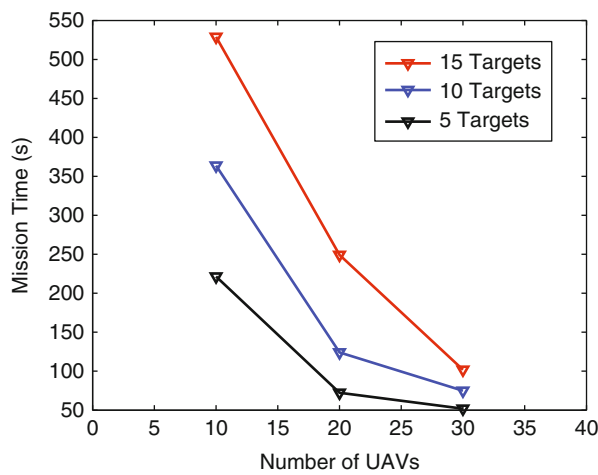


Fig. 83.8 Variation of mission time with increase in number of UAVs for various number of targets

The mission time is defined as the time taken to destroy all the targets within the region of interest. The simulation is done for a maximum simulation time of 1,500 s. If the mission is not complete (i.e., all the targets are not destroyed) within that time, then the mission time is deemed to be 1,500 s. The targets are assumed to have three type of resources. The quantity of each type is a random integer between 0 and 3. The UAVs also have similar resource distribution. However, if the total UAV resources are not enough to destroy the target, then resources are added to UAVs before the start of the mission so that the mission is feasible. The above setting is also used for rest of the simulations.

Figure 83.8 gives the total mission time, that is, the time required to find and destroy all the targets, as a function of the number of UAVs used in the mission. Cases where the number of target are 5, 10, and 15 are considered. The results reported are averages over 100 Monte Carlo runs with different initial positions and orientations of UAVs and targets. As seen in the figure, for a fixed number of targets, the time to destroy all the targets decreases as the number of UAVs deployed increases.

83.5.3 Effect of Varying Communication Range

The effect of change in UAV communication ranges on the completion time of the search, and destroy mission is studied in this section. Simulation settings are the same as described above except for the communication range which is varied. The results obtained are shown in Fig. 83.9. As expected, the mission performance increases with increase in communication ranges of UAVs. This is because, with

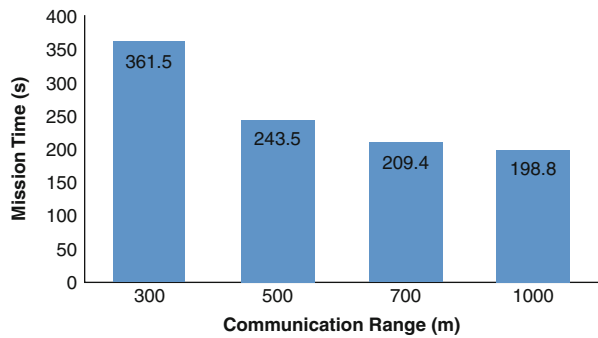


Fig. 83.9 Variation of mission time with increase in communication range of UAVs for a mission with 10 UAVs and 10 targets

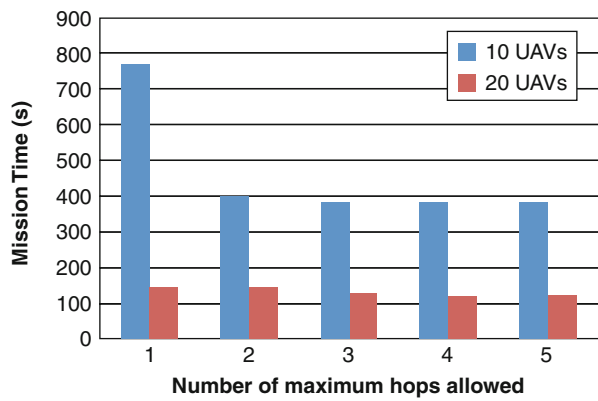


Fig. 83.10 Variation of mission time with increase in maximum hops allowed for 10 and 20 UAVs prosecuting 10 targets

more communication range, it is possible for the coalition leader to interact with more number of UAVs leading to a higher chance of a successful coalition being formed. At very high values of communication range, a UAV will be able to communicate with all other UAVs without the need for a relay agent, and then the problem becomes equivalent to that with global communication.

83.5.4 Effect of Hop Delay and Max-Hops

The goal of this experiment is to study the effects of communication delays and number of maximum allowed hops for a message in the network on the mission performance. The simulation setting is the same as described before.

Figure 83.10 shows the variation of average mission time (averaged over 100 runs) taken by 10 and 20 UAVs to destroy 10 targets with respect to H_{max} . Next, the

effect of increase in H_{\max} for a given $\Delta_{\text{hop}} = 0.1$ s is studied. Consider the change in mission time for H_{\max} varying from 1 to 5 as shown in Fig. 83.10. With increase in H_{\max} , the mission time decreases. This is because, as the network depth is more, the coalition leader can get more potential coalition members than being restricted to only its immediate neighbors as in the case for low value of H_{\max} say, $H_{\max} = 1$, and hence will be able to make feasible coalitions that can destroy the target quicker. However, further increasing H_{\max} will result in a slight degradation of performance. The degradation in performance occurs as the employed communication protocol requires a subnetwork that is invariant throughout the coalition formation period. Since it is harder to find a wider network (higher H_{\max}) that stays invariant for a specified time period, the mission time increases with H_{\max} and even more so in the case where Δ_{hop} is high (for the cases presented in Fig. 83.10, $\Delta_{\text{hop}} = 0.1$ s). Thus, the degradation in mission performance with increase in H_{\max} will be more prominent for high Δ_{hop} . This effect is clearly seen in Fig. 83.11 where the combined effect of H_{\max} and Δ_{hop} is captured. The figure shows the change in mission time for a search and destroy task with 10 UAVs and 10 targets with respect to maximum number of allowed hops for different values of communication delay. Note the degradation of mission performance for the cases where $\Delta_{\text{hop}} = 0.5$ and 1 s when the number of maximum allowed hops is more than 3.

Figure 83.12 shows the effect of communication delay on the mission completion time for search and destroy missions with 10 and 20 UAVs and 10 targets. As seen from the figure, the mission time increases with increase in communication delay. This is because an increase in communication delay will cause an increase in the coalition formation period. Since the communication protocol depends on finding a subnetwork that has an invariant topology throughout the coalition formation process, it becomes harder to find such a network under higher communication

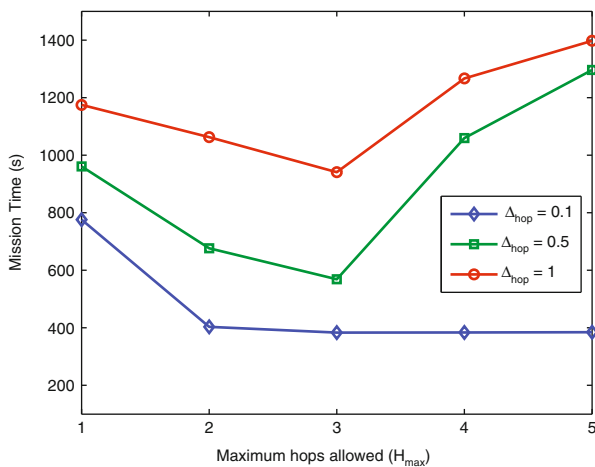


Fig. 83.11 Effect of hop delay Δ_{hop} and maximum number of allowed hops H_{\max} on mission performance (different curves correspond to $\Delta_{\text{hop}} = 0.1, 0.5$, and 1 s)

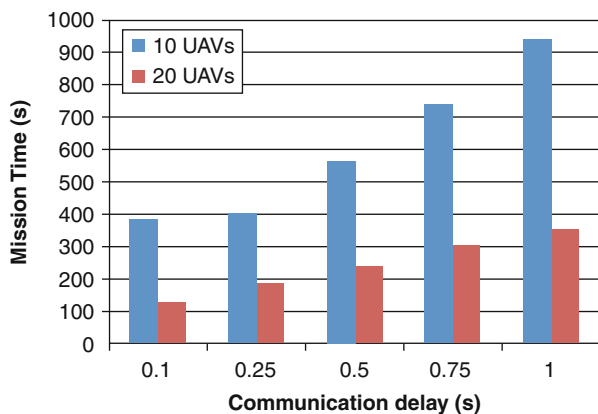


Fig. 83.12 Variation of mission time with increase in communication delay for 10 and 20 UAVs prosecuting 10 targets

delays. This results in the inability of coalition leader to find potential members to form feasible coalitions leading to higher mission completion times.

From these studies, it can be concluded that the hop delay in the network and the selection of H_{\max} are important for the performance of the mission in terms of the mission completion time. Also, note that all the simulations had UAVs with different speed which shows the efficacy of the algorithm for use with heterogeneous UAVs.

83.6 Conclusions

In this chapter, a coalition formation algorithm for UAVs having limited communication ranges with associated communication delays is presented. A novel mechanism is developed to determine potential coalition members in a dynamic network formed by UAVs moving around. The UAVs in the coalition, sequentially, use a track and strike method to deploy their resources on the moving target. The effect of communication delay and the maximum number of allowed hops for a message in the network to determine potential coalition members for a coalition is studied. The results showed that if the delay is large, then a coalition formed from the immediate neighbors are sufficient for a good performance in terms of mission completion time. Under smaller delays, the simulations show that including neighbors up to a few hops will increase performance, and any additional increase in hop count will degrade performance. In these studies, collision avoidance among the UAVs is not included. Thus, an important aspect of extending the problem lies in including collision avoidance into the UAV system, and to study how the UAVs will alter their routes to avoid collisions while tracking and striking the target. It will be interesting to investigate the performance of the multiple UAV search and destroy mission under such a situation.

References

- S. Alagar, S. Venkatesan, J. Cleveland, Reliable broadcast in mobile wireless networks, in *Proceedings of IEEE Military Communications Conference 1995*, Boston, 1995, pp. 236–240
- M. Alighanbari, Y. Kuwata, J.P. How, Coordination and control of multiple uavs with timing constraints and loitering. *Proc. Am. Control Conf.* **6**, 5311–5316 (2003)
- P. Chandler, M. Pachter, S.J. Rasmussen, C. Schumacher, Distributed control for multiple uavs with strongly coupled tasks, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, 2003, pp. AIAA 2003–5799
- J. Chen, M. Huang, A broadcast engagement ack mechanism for reliable broadcast transmission in mobile ad hoc networks. *IEICE Trans. Commun.* **E88-B**, 3570–3578 (2005)
- M. Clark, *Data Networks, IP and the Internet: Protocols, Design and Operation* (Wiley Publication, Chichester/Hoboken, 2003)
- D. Dionne, C.A. Rabbath, Multi-uav decentralized task allocation with intermittent communications: the dtc algorithm, in *Proceedings of the American Control Conference ACC '07*, New York, 2007, pp. 5406–5411
- L.E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**, 497–516 (1957)
- B. Gerkey, M.J. Mataric, Formal framework for the study of task allocation in multi-robot systems. *Int. J. Rob. Res.* **23**, 939–954 (2004)
- IEEE-SA Standards Board, *IEEE Wireless Lan Medium Access Control (mac) and Physical Layer (phy) Specifications* (Industry standard, IEEE Computer Society, New York, 2007)
- D.B. Kingston, C.J. Schumacher, Time-dependent cooperative assignment, in *Proceedings of the American Control Conference*, Portland, 2005, pp. 4084–4089
- L. Lamport, Using time instead of timeout for fault-tolerant distributed systems. *ACM Trans. Program. Lang. Syst.* **6**, 254–280 (1984)
- T. Lemaire, R. Alami, S. Lacroix, A distributed tasks allocation scheme in multi-uav context, in *Proceedings of the IEEE International Conference Robotics and Automation ICRA '04*, New Orleans, vol. 4, 2004, pp. 3622–3627
- J.G. Manathara, P.B. Sujit, R.W. Beard, Multiple uav coalitions for a search and prosecute mission. *J. Intell. Rob. Syst.* **62**(1), 125–158 (2011)
- J. Mitchell, P. Chandler, M. Pachter, S.J. Rasmussen, Communication delays in the cooperative control of wide area search munitions via iterative network flow, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin 2003, pp. AIAA 2003–5665
- K.E. Nygard, P.R. Chandler, M. Pachter, 2001, Dynamic network flow optimization models for air vehicle resource allocation, in *Proceedings of the American Control Conference*, Arlington, pp. 1853–1858
- K. Obraczka, K. Viswanath, G. Tsudik, Flooding for reliable multicast in multi-hop ad hoc networks. *Wirel. Netw.* **7**, 627–634 (2001)
- L.E. Parker, F. Tang, Building multi-robot coalitions through automated task solution synthesis. *Proc. IEEE* (special issue on Multi-Robot Systems) **94**, 1289–1305 (2006)
- P. Scerri, E. Liao, J. Lai, K. Sycara, Coordinating very large groups of wide area search munitions, in *Theory and Algorithms for Cooperative Systems* (World Scientific Pub Co Inc, New Jersey, USA, 2005)
- C. Schumacher, P. Chandler, M. Pachter, L. Pachter, Uav task assignment with timing constraints, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, 2003a, pp. AIAA 2003–5664
- C. Schumacher, P.R. Chandler, S.J. Rasmussen, D. Walker, Task allocation for wide area search munitions with variable path length, in *Proceedings of the American Control Conference*, Boston, vol. 4, 2003b, pp. 3472–3477
- O.M. Shehory, Methods for task allocation via agent coalition formation. *Artif. Intell.* **101**, 165–200 (1998)
- T. Shima, C. Schumacher, Assigning cooperating uavs to simultaneous tasks on consecutive targets using genetic algorithms. *J. Oper. Res. Soc.* **60**, 973–982 (2009)

- P.B. Sujit, R. Beard, Distributed sequential auctions for multiple uav task allocation, in *Proceedings of the American Control Conference ACC '07*, New York, 2007, pp. 3955–3960
- K. Tang, M. Gerla, MAC reliable broadcast in ad hoc networks, in *Proceedings of the IEEE Military Communications Conference 2001*, McLean, vol. 2, 2001, pp. 1008–1013
- D. Turra, L. Pollini, M. Innocenti, Fast unmanned vehicles task allocation with moving targets, in *Proceedings of the IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004
- M.M. Vanzin, K.S. Barber, *Decentralized Partner Finding in Multi-agent Systems* (Springer, Dordrecht, 2006), pp. 75–98
- L. Vig, A.J. Adams, Market-based multi-robot coalition formation, in *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, ed. by M. Gini, R. Voyles (Springer, Minneapolis, 2006), pp. 227–236
- B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in *Proceedings of MobiHoc 2002* (ACM, New York, 2002), pp. 194–205

Magnus Egerstedt, Amir Rahmani, and Shin-Yih (Ryan) Young

Contents

84.1 Introduction 2050

84.1.1 Problem 1: Time-Optimal Protection Paths 2050

84.1.2 Problem 2: Assigning and Dispatching UAVs 2053

84.2 Optimal Convoy Protection Maneuvers 2055

84.2.1 Single UAV Time-Optimal Paths 2055

84.2.2 Multi-UAV Convoy Protection 2061

84.2.3 Moving Convoy Protection 2063

84.3 The Assignment Problem 2066

84.3.1 Selection Policy and Task Allocation Algorithm 2069

84.4 Experimental Results 2072

References 2075

Abstract

This chapter investigates how to control and coordinate teams of unmanned aerial vehicles (UAVs) to provide protection to convoys of ground vehicles. This is a rich yet canonical problem when coordinating multiple UAVs in that coordinated movements, task assignments, and resource balancing must all be performed for a successful completion of the mission. Time-optimal paths for providing convoy

M. Egerstedt (✉)
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta,
GA, USA
e-mail: magnus@ece.gatech.edu

A. Rahmani
Department of Mechanical and Aerospace Engineering, University of Miami, Coral Gables,
FL, USA
e-mail: a.rahmani@miami.edu

S.-Y. (Ryan) Young
Advanced Technology Center, Rockwell Collins, Cedar Rapids, IA, USA
e-mail: syoung@rockwellcollins.com

protection to stationary ground vehicles are presented, and these algorithms are extended to moving ground vehicles. The assignment problems, associated with dispatching UAVs from the convoy to inspect and clear potential threats, are moreover discussed.

84.1 Introduction

Airplanes can be used to provide close air support or large-scale area surveillance for ground convoys in unknown and potentially dangerous environments. Wide-spread use of unmanned ground vehicles (UGVs) to conduct tasks in these environments has necessitated the design of practical approaches to effectively control and coordinate multiple UAVs to provide coverage, surveillance, tracking, and convoy protection for the UGVs (see, e.g., Beard et al. 2006; Cruz et al. 2007; Girard et al. 2004; Grocholsky et al. 2006; Lee et al. 2003; Spry et al. 2005). This chapter discusses this problem, which can be decomposed into two subproblems. The first problem involves the design of coordinated UAV maneuvers to ensure the ground convoy is protected. The second problem concerns how to assign and dispatch UAVs from the convoy to clear potential threats in such a way that the fuel consumption is balanced across the team of UAVs.

84.1.1 Problem 1: Time-Optimal Protection Paths

UAVs often use onboard Electro-optical/infrared cameras in order to obtain information about surrounding areas. In many surveillance applications with small UAVs (e.g., Morris and Holden 2000; Quigley et al. 2005), the motion of the camera is typically decoupled from that of the UAV, using a gyro-stabilized camera platform that keeps the camera pointing in the same direction regardless of the motion of the UAV. This approach provides the UAV with stable images even under high-frequency oscillations associated with the UAV itself. Hence, the assumption is made that the onboard camera always points down and, as such, the UAV monitors a circular disk on the ground.

To solve the problem of providing coordinated protection maneuvers, an optimal approach is adopted for path planning and coordination of multiple UAVs. The UAVs are modeled as Dubins vehicles flying at constant altitude. Limited ranges of sensors onboard the UAVs, together with their kinematic constraints, might make it impossible to provide coverage to the ground vehicles with a single UAV. In this case, it is of interest to find the best path for an individual UAV so that it can monitor the ground vehicles for the longest time and then coordinate the UAVs in such a way that the ground vehicles are visible to at least one UAV at any given time, as illustrated in Fig. 84.1.

A Dubins vehicle is a planar vehicle with bounded turning radius and constant forward speed. L.E. Dubins was the first to give a characterization of time-optimal trajectories for such a vehicle using geometric methods (Dubins 1957). Shortest-path problems for Dubins vehicles have been since studied extensively

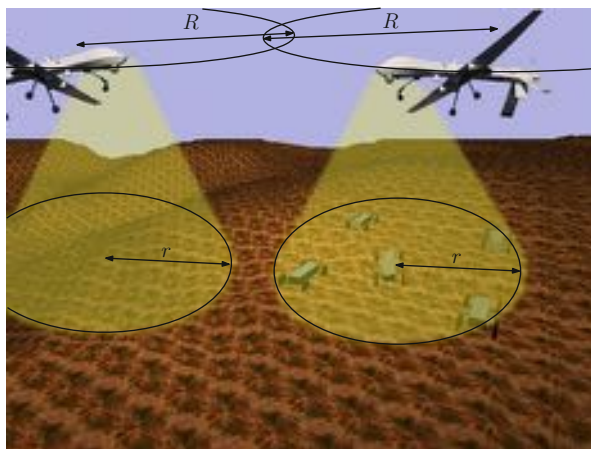


Fig. 84.1 UAVs providing convoy protection to UGVs. The UAVs are assumed to be kinematically restricted by their minimum turning radius R . The sensors onboard the UAVs also have limited range and are assumed to be able to observe a disk of radius r on the ground

(see Agarwal et al. 1998; Boissonnat et al. 1994; Dubins 1961, for example). A Dubins vehicle that can move backward was studied by Reeds and Shepp (1990), and the shortest-path problem for a Reeds-Shepp vehicle was further studied and classified by Souères and Laumond (1996). Walsh et al. (1994) found optimal paths using quadratic cost functions for a Dubins airplane in $SE(2)$, $SO(3)$, and $SE(3)$. McGee et al. (2005) obtained time-optimal paths for Dubins vehicles in constant wind. Dubins vehicles have moreover been used as a simplified model to describe planar motions of UAVs in Savla et al. (2008). Chitsaz and LaValle (2007) extend the Dubins' model from $SE(2)$ to $SE(2) \times \mathbb{R}$ to account for altitude changes and gave a characterization of the time-optimal trajectories for this model based on the final altitude.

Optimal trajectories of a Dubins vehicle are often constructed using motion primitives (see Belta et al. 2007; Frazzoli et al. 2005; Hauser et al. 2006, for example). For point-to-point minimum-time transfer problems, it has been shown that the optimal solutions are curves consisting of only three motion primitives: line segments and circular arcs turning maximally to the left and to the right (see Dubins 1957; Lavalle 2006; Souères and Boissonnat 1998). The optimal paths for the point-to-point minimum-time transfer problem are characterized by sequences of these three motion primitives. This chapter extends this to the case of stationary convoy protection, where resulting time-optimal paths are characterized by sequence of only two motion primitives (maximally turning left and turning right) and do not include any line segments. In fact, this chapter draws upon the results in Ding et al. (2009). Similar to many previous time-optimal path-planning work (e.g., Boissonnat et al. 1994; Chitsaz and LaValle 2007; Ding et al. 2009; McGee et al. 2005; Souères and Laumond 1996; Souères and Boissonnat 1998), it makes use of Pontryagin's Minimum Principle to derive optimal trajectories.

When addressing this problem, the first case under investigation is when the convoys are stationary, and global optimal paths are found for a single UAV that maximize the coverage time, that is, the time the UAV spends with the convoy being visible. It is then shown how to coordinate a group of UAVs to provide continuous coverage of the convoys and the minimum number of UAVs required to achieve continuous convoy protection for all time, as well as how to extend these results to moving rather than stationary ground convoys.

Dubins vehicles flying at constant altitude with unit speed (The unit speed assumption is justified since the results describe paths which are invariant under different forward speeds.) and minimum turn radius of R can be modeled as

$$\begin{cases} \dot{x} = \cos(\theta) \\ \dot{y} = \sin(\theta) \\ \dot{\theta} = \omega, \end{cases} \quad (84.1)$$

where x and y are the position of the UAV in the x - y plane at the altitude the UAV is flying and ω is the angular velocity of the vehicle. The angular velocity is bounded by the inverse of the minimum turn radius R of the vehicle, that is, $\omega \in [-\frac{1}{R}, \frac{1}{R}]$. The state of the system be given by $q(t) = [x(t), y(t), \theta(t)]^T$.

As already stated, it is assumed that the cameras onboard the UAVs can monitor a disk of radius r on the ground (see Fig. 84.1 for an illustration of this scenario). That is, it is assumed that the onboard camera is attached to a gyro and is always looking down, regardless of the bank and pitch angles of the UAV. A gimbaled camera system is commonly used in UAV surveillance applications to decouple camera from dynamics of the plane (see, e.g., Morris and Holden 2000; Quigley et al. 2005). In addition, to simplify the problem, the ground convoys are considered to be points located at the centroid of the convoys in the x - y plane. Hence, successful convoy protection is defined as being achieved when the centroid of the UGVs is visible to at least one of the UAVs at any time. Or, in other words, convoy protection is provided by the UAV if the horizontal distance between the UAV and the centroid of the UGVs is less than or equal to r .

The disk of observation and its radius r certainly depend on the altitude of the UAV, and the angular range of the camera but to ensure quality of observation and successful protection, cameras or sensors onboard the UAV usually have narrower field of view than the UAV's turning radius. This is especially true for cameras and sensors that carry out tasks using computer vision algorithms, which require certain level of image resolution. In these cases, $R > r$, and if the UGVs are stationary, then a single UAV is not capable of providing convoy protection to the ground vehicles indefinitely, and a control strategy is needed to optimize the time in which convoy protection is achieved. This can be seen by drawing a circle of radius r using the position of the centroid of UGVs as the center. If there is only one UAV, then due to the fact that $R > r$, the UAV will eventually fly out of this circle no matter where it starts.

Note that in case of static convoys, if $R \leq r$, then the convoy protection problem is trivial since it can be solved by using a single UAV flying on a circular path of

radius R with the center being the centroid of the ground vehicles. In this case, convoy protection is provided for all time using only one UAV.

In this chapter, the problem of controlling and coordinating UAVs is considered, with the task of the UAVs being that of providing convoy protection for both stationary UGVs and UGVs moving on a straight line. In both cases, it is assumed that $R > r$. Let the *convoy circle* denote a disk of radius r centered at the centroid of the UGVs. Using the convoy circle, it can be seen that convoy protection is achieved if the ground track of at least one UAV is present inside the convoy circle at any time. Because of the kinematic constraint (turning radius R of the UAV), the UAVs are required to be coordinated so that they collectively provide continuous convoy protection while individual UAVs enter and leave the convoy circle.

84.1.2 Problem 2: Assigning and Dispatching UAVs

In the convoy protection problem, being able to have the UAVs track the UGVs is not enough. Sometimes UAVs need to be dispatched in order to clear potential threats. In fact, given a single UAV and N known threats that must be cleared, together with a cost associated with traversing between the threats, the problem of selecting the order in which the threats should be visited (and cleared) is a variant of the well-studied traveling salesperson (TSP) problem (Burkard 2002; Garey and Johnson 1979). Unfortunately, TSP is a NP-hard problem, that is, there are no polynomial (in the number of threats)-time algorithms that can solve the problem. As a consequence, *any attempt at tackling the full-fledged TSP using limited computational resources is doomed to failure*. The reason for this computational intractability is that, in order to solve the TSP problem, one has to establish the order in which the threats are visited. If, on the other hand, this order is a priori given, the problem would already be solved.

If there are multiple UAVs (as is the case in the convoy protection scenario under consideration here), the complexity of the TSP problem is reduced if the number of UAVs is greater than or equal to the number of threats (including both known and pop-up threats). The reason for this is that one ends up with a so-called matching problem (Pardalos et al. 1994), where one simply has to decide which threat should be visited by which UAV. And, since there are no threats left unvisited after all UAVs have visited one threat, there is again no ordering of the threats that must be decided upon. The matching problem can be solved in cubic time (in the number of threats) using the Hungarian algorithm (Kuhn 1955), as is done, for example, in Ji et al. (2006) and Moore and Passino (2005). If there are fewer UAVs than threats, then the NP-hardness result still applies (Arsie et al. 2009; Pavone et al. 2009), and this is the situation (fewer UAVs than threats) that can be expected in real-world operations.

Numerous algorithms for approximating and addressing the TSP problem in a computationally tractable manner have been proposed. These algorithms typically fall under two categories, namely, (1) greedy algorithms (Beard et al. 2002; Psaraftis 1988; Richards et al. 2002; Schumacher et al. 2003), where each agent is trying

to locally maximize and minimize a cost based solely on the instantaneous cost rather than over the whole time horizon, and (2) auction-based and game theoretic algorithms (Arslan et al. 2007), where the agents get to “bid” on which tasks they want to be assigned to. In the greedy case, the solution is clearly suboptimal, even though it was shown in Arsie et al. (2009) and Pavone et al. (2009) that asymptotically (as the number of threats approach infinity), the solution gets to within a constant factor of the optimal solution. Auction-based solutions, on the other hand, typically require a significant amount of information to be shared by the agents, and despite this increase in shared information and the required communication bandwidth, the inherent complexity of the problem remains the same. As such, auction-based methods may provide some guidance as to finding suboptimal solutions, but they do not, unfortunately, overcome the complexity issue and they come with an increased communications overhead. Solutions that fall in between these two approaches were presented in Ponda et al. (2010), where decentralized suboptimal solutions were obtained by focusing on the feasibility of the solution over limited time horizons rather than the minimization of any particular cost function, and in Northrop Grumman, where the assignment and allocation problems were addressed for more specialized surveillance missions. This last approach (as exemplified by Northrop Grumman and Ponda et al. 2010) is one that will be followed in this chapter in that the special structure of the convoy protection problem will be leveraged to obtain good suboptimal solutions to an intractable problem.

The convoy protection problem is *not* a standard TSP (single or multi-UAV) problem even though it shares some of the same features. The two main differences are:

- Threat dynamics: The threats are not given all at once. Instead, they show up throughout the mission, and as a consequence, the problem must be resolved whenever a new threat appears.
- Overall cost: The cost to be minimized is not the total distance traveled or the total fuel consumed by the team of UAVs. Instead, it needs to reflect the fact that the fuel consumption should be balanced among the different UAVs, since success of convoy protection mission depends on participation of all UAVs in convoy protection effort. As such, it is a min-max cost in that what must be minimized is the fuel consumption of the UAV who has to expend the most fuel.

As a consequence of these considerations, the convoy protection problem is still NP-hard, and in order to be able to still solve it in a resource-constrained environment, some simplifying assumptions must be made. However, one can take the explicit *structure* of the convoy protection problem into account when making these assumptions.

A1 Bounded number of threats: At any given time, only a fixed number of threats will be considered (e.g., roadside bombs). Even though no a priori upper limit is placed on the number of threats that may appear, the algorithm will be restricted to only consider the closest M threats, where M is a fixed number. Without further assumptions, this will still make the complexity factorial in M , which may be a large number (depending on what M is), but at least, it is bounded.

- A2 *One-dimensional threat distribution*: As the convoy is moving along an established route, for example, a road, only threats on (or close to) the road will be considered. As such, the threats will be placed along a one-dimensional path, and they will be encountered sequentially. The result is that the order in which the threats are cleared is already given in that the closest threat is cleared first.
- A3 *Not all threats must be cleared*: The objective of the convoy protection algorithm is to ensure that the convoy is not endangered by any threats. There are two ways in which this can be achieved. The first is by clearing threats, as previously discussed. However, one can also replan the path the convoy takes in order to produce a detour around a threat. This will be done if the threat cannot be cleared in time which allows for threats to be ignored that would otherwise require too much fuel to be cleared.

These three assumptions will allow for a polynomial-time algorithm for solving the assignment aspect of the overall convoy protection problem.

84.2 Optimal Convoy Protection Maneuvers

84.2.1 Single UAV Time-Optimal Paths

The first problem to consider is the problem of using one UAV to provide convoy protection to some stationary UGVs for the maximum amount of time, which is equivalent of finding the longest feasible path inside the convoy circle. What needs to be determined is both the optimal path for a single UAV starting at a fixed initial condition and the optimal path if the UAV is allowed to pick the initial condition (position and heading) when entering the convoy circle.

Fix the origin of the x - y plane at the centroid of the UGVs. A maximum-time, optimal control problem with state constraint $x^2 + y^2 - r^2 \leq 0$ and input constraint $|w| \leq \frac{1}{R}$ is then obtained. Furthermore, it can be assumed that the UAV starts at a point on the state constraint boundary (convoy circle). This assumption does not limit the generality of the result, since, if the UAV starts inside the convoy circle, Bellman's Principle of Optimality can be used to obtain the optimal path for the UAV by integrating backward in time.

To facilitate the analysis, it is useful to impose an extra terminal manifold constraint. Since the optimal solution always has the terminal state (henceforth denoted as the exit state) on the boundary of the state constraint set (exiting the convoy circle), the terminal constraint of being on the convoy circle when exiting is enforced. To simplify, the following notation is introduced: $q_T := q(T)$ and $[x_T, y_T, \theta_T] := [x(T), y(T), \theta(T)]$. The terminal manifold can then be defined as the set of states that satisfy

$$M(q_T) = x_T^2 + y_T^2 - r^2 = 0. \quad (84.2)$$

The optimal control problem is given by

Problem 84.2.1

$$\min_{\omega(t)} J = \int_0^T -1 dt, \quad (84.3)$$

subject to the dynamics of (84.1) with a given initial condition $q(0)$, the input constraint

$$-\frac{1}{R} \leq \omega(t) \leq \frac{1}{R}, \quad (84.4)$$

the state constraint

$$x(t)^2 + y(t)^2 - r^2 \leq 0 \quad (84.5)$$

$$x(0)^2 + y(0)^2 - r^2 = 0, \quad (84.6)$$

and the terminal manifold constraint

$$M(q(T)) = x(T)^2 + y(T)^2 - r^2 = 0. \quad (84.7)$$

This problem is henceforth denoted as $\Pi_{q(0)}$.

When solving this problem, one immediately should exclude initial conditions that generate no paths entering the convoy circle. This occurs when the initial heading $\theta(0)$ points away from the convoy circle. The set of initial conditions Λ that are considered for the optimization problem are thus given by

$$\Lambda = \left\{ q = [x, y, \theta]^T : x^2 + y^2 = r^2, \text{ and } -\frac{\pi}{2} < \theta - \text{atan2}(y, x) < \frac{\pi}{2} \right\}. \quad (84.8)$$

The set Λ is called the feasible entry set. For simplicity of notation, it is assumed that all angles are taken modulo 2π .

$\Pi_{q(0)}$ is an optimal control problem with both input and state inequality constraints. Optimal control problem with state inequality constraints is usually hard or impossible to solve explicitly. Fortunately, in this problem, due to the special structure of the state inequality constraint (84.5), there are only two points on the state trajectory where the constraint is active. These two states correspond to when the UAV is entering and exiting the convoy circle. Due to this special structure, an auxiliary state is used to handle the state constraint. Define $\xi(x^2 + y^2 - r^2)$ as an inverted Heaviside function:

$$\xi(x^2 + y^2 - r^2) = \begin{cases} 0 & : x^2 + y^2 - r^2 \leq 0 \\ 1 & : \text{otherwise.} \end{cases} \quad (84.9)$$

Define a new state τ as

$$\dot{\tau}(t) = (x^2 + y^2 - r^2)^2 \xi(x^2 + y^2 - r^2). \quad (84.10)$$

The state of the UAV is then augmented as $\bar{q}(t) = [q(t), \tau(t)]^T$. Assuming that $\tau(0) = 0$ and imposing the terminal constraint that $\tau(T) = 0$, it follows that $\dot{\tau}(t) = 0, \forall t \in [0, T]$, since $\dot{\tau}(t) \geq 0, \forall t \in [0, T]$. Hence,

$$\tau(t) = 0, \forall t \in [0, T]. \quad (84.11)$$

Note that the terminal constraint $\tau(T) = 0$ enforces the state inequality constraint (84.5). Using this auxiliary state τ , the state inequality constraint is transformed into an equivalent terminal constraint. When there is no ambiguity, it is assumed that the state constraint (84.5) is satisfied and $q(t)$ is still referred to as the state trajectory.

The Hamiltonian for this optimal control problem is

$$\begin{aligned} \mathcal{H} = & -1 + \lambda_1 \cos \theta + \lambda_2 \sin \theta + \lambda_3 \omega \\ & + \lambda_4 (x^2 + y^2 - r^2)^2 \xi (x^2 + y^2 - r^2), \end{aligned} \quad (84.12)$$

where $\lambda = [\lambda_1, \dots, \lambda_4]^T$ are the costates. The costates satisfy the following differential equations in the time interval $[0, T]$:

$$\begin{aligned} \dot{\lambda}_1 &= -\frac{\partial \mathcal{H}}{\partial x} = -2x\lambda_4(x^2 + y^2 - r^2)\xi(x^2 + y^2 - r^2), \\ \dot{\lambda}_2 &= -\frac{\partial \mathcal{H}}{\partial y} = -2y\lambda_4(x^2 + y^2 - r^2)\xi(x^2 + y^2 - r^2), \\ \dot{\lambda}_3 &= -\frac{\partial \mathcal{H}}{\partial \theta} = \lambda_1 \sin \theta - \lambda_2 \cos \theta, \\ \dot{\lambda}_4 &= -\frac{\partial \mathcal{H}}{\partial \tau} = 0. \end{aligned}$$

When the state constraint (84.5) is satisfied, the last term in the Hamiltonian (84.12) does not contribute since $\lambda_4(x^2 + y^2 - r^2)^2 \xi(x^2 + y^2 - r^2) = 0, \forall t \in [0, T]$, and thus the Hamiltonian becomes

$$\mathcal{H} = -1 + \lambda_1 \cos \theta + \lambda_2 \sin \theta + \lambda_3 \omega. \quad (84.13)$$

The necessary optimality condition from the Pontryagin's Minimum Principle states that

$$\begin{aligned} \mathcal{H}(\bar{q}^*(t), \lambda^*(t), \omega^*(t), t) &\leq \mathcal{H}(\bar{q}^*(t), \lambda^*(t), \omega(t), t), \\ \forall \omega(t) &\in \left[-\frac{1}{R}, \frac{1}{R} \right], t \in [0, T], \end{aligned} \quad (84.14)$$

where $\bar{q}^*(t)$ denotes the optimal augmented state trajectory, $\lambda^*(t)$ denotes the optimal costate trajectory corresponding to $\bar{q}^*(t)$, and $\omega^*(t)$ is the optimal control.

Using the necessary optimality condition (84.14) on the Hamiltonian equation (84.13), one can see that the optimal control $\omega^*(t)$ is a function of the costate $\lambda_3^*(t)$:

$$\omega^*(t) = \begin{cases} -\frac{1}{R}, & \text{if } \lambda_3^*(t) > 0 \\ \frac{1}{R}, & \text{if } \lambda_3^*(t) < 0 \\ \text{undetermined,} & \text{if } \lambda_3^*(t) = 0. \end{cases} \quad (84.15)$$

Thus, it can be seen that when $\lambda_3^*(t) > 0$, the optimal control is maximum turning right and when $\lambda_3^*(t) < 0$, the optimal control is maximum turning left. When $\lambda_3^*(t) = 0$ for a finite time interval, then any control $\omega(t) \in [-\frac{1}{R}, \frac{1}{R}]$ satisfies the Minimum Principle, and the finite time interval when this case arises is called a singular interval (for discussions on singular intervals for optimal control problems, see any standard text on optimal control, such as Kirk 2004). Hence, the optimal control is in the form of bang-bang (if there is no singular interval) or bang-off-bang (if there are singular intervals). And if there is a singular interval for $\Pi_{q(0)}$, then it is necessary that there exists a time interval $[t_1, t_2]$ such that $\lambda_3(t) = 0$ and $\dot{\lambda}_3(t) = 0$ for all $t \in [t_1, t_2]$. For Dubins vehicles with dynamics specified in Eq. (84.1), singular intervals result in line segments as part of the optimal path, and as a consequence, line segments are usually part of the optimal paths for shortest-path (or minimum-time) Dubins vehicle problems.

Definition 84.2.1 For a state trajectory $q(t), t \in [0, T]$ satisfying the state constraint (84.5), if the costate trajectory $\lambda_3(t)$ and corresponding input $\omega(t)$ satisfy the control strategy (84.15), then $q(t)$ is referred to as a *candidate optimal trajectory* (COT).

Pontryagin's Minimum Principle states that being a COT is a necessary condition for being an optimal solution, and assuming that a trajectory $q(t)$ is a COT but only its terminal state q_T is given, the entire trajectory $q(t)$, its corresponding costate trajectory $\lambda(t)$ satisfying the necessary optimality condition (84.14), and the control $\omega(t)$ satisfying the optimal control strategy (84.15) can all be uniquely determined from the terminal state q_T , as is shown in Ding et al. (2009, 2010). In fact, in Ding et al. (2010), it was shown that an optimal trajectory cannot contain both a circular arc and a line segment. In addition, the COT can be either a line through the origin or a curve consisting of circular arcs of radius R . Hence, if the optimal control is not constant, then it can only change between $\omega = -\frac{1}{R}$ and $\omega = \frac{1}{R}$, since there is no singular interval in that case. Henceforth, the term *switching time* is used for the time instant when the control law switches between $\omega = -\frac{1}{R}$ and $\omega = \frac{1}{R}$. Furthermore, a switching point in a state trajectory is defined as the state when the controller switches.

For a number of terminal conditions q_T , the corresponding COTs are shown in Fig. 84.2. For all of these terminal conditions, x - y coordinates of the terminal state

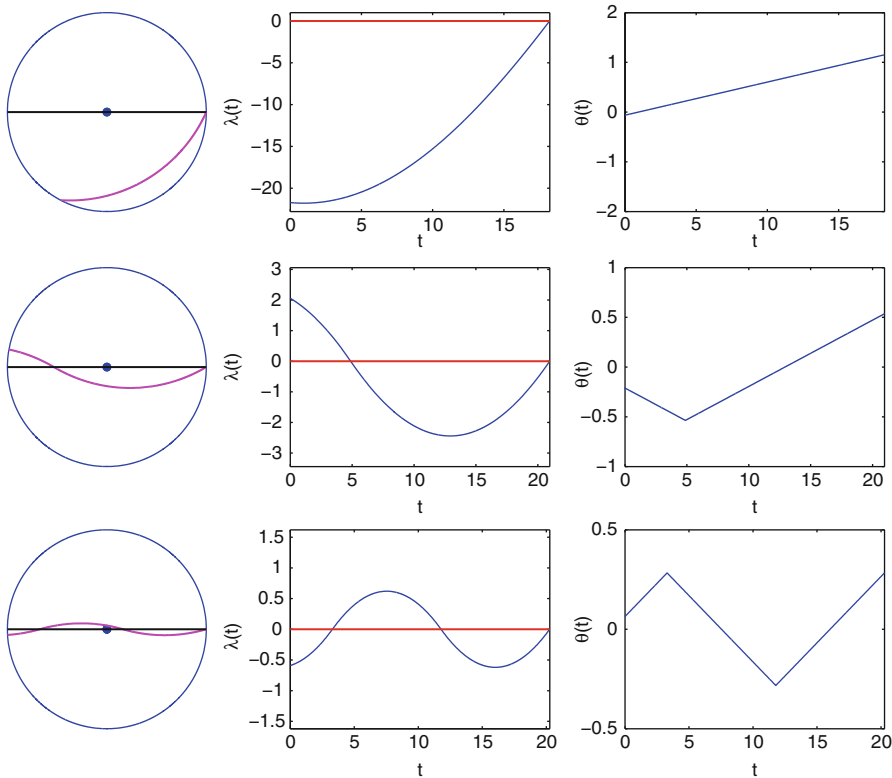


Fig. 84.2 A number of COTs are plotted for different exit angles θ_T resulting in different number of switchings. The *left plots* are the COTs in x - y plane, the *middle plots* are the costates λ_3 , and the *right plots* are the angles $\theta(t)$. The first, second, and third rows correspond to the cases when this is 0, 1, and 2 switchings, respectively

are fixed at $[r, 0]^T$ (hence $\psi_T = 0$), but the exit angle θ_T is allowed to vary. The path in the x - y plane $[x(t), y(t)]$, costate $\lambda_3(t)$, and angle $\theta(t)$ are plotted from left to right.

An unusual aspect of this problem is that, unlike the Dubins vehicle shortest-path problem, the optimal paths for Problem 84.2.1 do not in fact contain line segments, as shown in Ding et al. (2010), even though such maneuvers do constitute valid COTs. Additionally, it can be shown (Ding et al. 2010) that the optimal paths never switch more than once, which leads to the following key result when characterizing the best protection paths:

Theorem 84.2.1 *For any initial condition $q(0)$, the optimal solution to Problem 84.2.1 is a UAV trajectory that does not switch more than once and that does not contain any straight line segments.*

Based on this theorem, the set of optimal motion primitives does not contain the motion of going straight. As such, two motion primitives $\{\mathbf{L}, \mathbf{R}\}$ are defined, where \mathbf{L} and \mathbf{R} motion primitives turn the vehicle maximally to the left and right, respectively. Furthermore, since the optimal trajectory only switch once, there are only four possible sequences of the $\{\mathbf{L}, \mathbf{R}\}$ motion primitives, namely,

$$\{\mathbf{L}, \mathbf{R}, \mathbf{LR}, \mathbf{RL}\}, \quad (84.16)$$

where \mathbf{LR} stands for turning left then right and \mathbf{RL} for turning right then left. The actual optimal paths can now be determined directly from the costate equations. In fact, since there are only four possibilities for the motion sequences in an optimal trajectory, it is easy to determine the global optimal path for any initial condition. A set of optimal paths for initial conditions with heading $\theta(0) = \frac{\pi}{2}$ are shown in Fig. 84.3. Figure 84.3 also shows the optimal switching surface on which switchings are optimal, and it can be observed that when a switching is needed for the optimal trajectory, the switching point, the origin, and the exit point are on the same line.

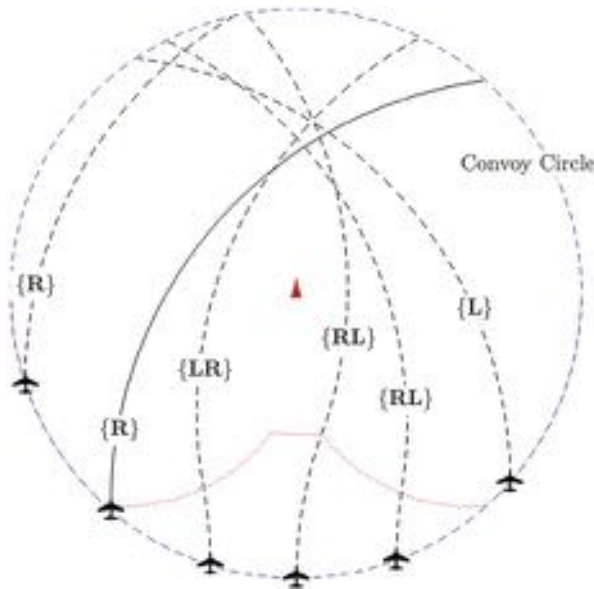


Fig. 84.3 A number of optimal state trajectories with initial heading $\frac{\pi}{2}$. The optimal switching points are plotted together to form the optimal switching surface. In this case, $R = 1.5r$. The *dashed paths* correspond to optimal paths. If $x(0) \in (-\frac{r^2}{R}, 0]$, then \mathbf{LR} is optimal. If $x(0) \in [0, \frac{r^2}{R})$, then \mathbf{RL} is optimal. If $x(0) \in [\frac{r^2}{R}, r)$, then \mathbf{L} is optimal. Otherwise, \mathbf{R} is optimal. The optimal switching points are plotted together to form the optimal switching surface. The *solid path* corresponds to the initial condition that $x(0) = -\frac{r^2}{R}$. This path and the path where $x(0) = \frac{r^2}{R}$ are equal in length and longer than all other optimal paths with the same initial heading

By rotating the initial state until the initial heading is $\frac{\pi}{2}$, the switching surface in Fig. 84.3 provides a control law which produces the optimal trajectory for any given initial condition. Now, this result is extended to address a perhaps more important problem: finding the optimal path inside the convoy circle with initial condition free to choose.

Problem 84.2.2

$$\min_{q(0)} J^*(q(0)), \quad (84.17)$$

where $J^*(q(0))$ is the solution of Problem 84.2.1 ($\Pi_{q(0)}$), with the initial state $q(0)$.

Let the optimal initial condition be denoted by $q^*(0)$; hence,

$$q^*(0) = \arg \min_{q(0)} J^*(q(0)). \quad (84.18)$$

The optimal path with this initial condition will be referred to as a globally (To clarify, the word *globally* is used here not within the context of local or global optimality. The solutions obtained for Problem 84.2.1 are global optimal trajectories each corresponds to a fixed initial condition. A globally optimal path in this chapter is defined as the longest path over all feasible initial conditions.) optimal path. $q^*(0)$ will be called an optimal entry state. It is apparent that any rotation of this state around the origin is also an optimal entry state. Hence, the globally optimal paths and optimal entry states are not unique. The set of optimal entry states, denoted by \mathbb{Q}^* , can be exactly determined by noting that optimal paths with optimal entry points do *not* switch. Instead, they are given by maximal left/right inputs while the entry point, the center of the convoy circle, and the exit point are located on the same line, as found in Ding et al. (2010).

84.2.2 Multi-UAV Convoy Protection

Due to kinematic constraint of the UAVs ($r < R$), it is impossible for one UAV to provide complete convoy protection for a group of UGVs. In this situation, multi-UAV coordination is required in order to successfully carry out convoy protection. The previous section laid out the groundwork to achieve optimal convoy protection by a group of UAVs. In particular, the set of optimal initial conditions that produces a set of globally time-optimal trajectory was characterized. It can moreover be shown that these optimal trajectories specify not only a path inside the convoy circle but also a path for a single UAV to come back to the convoy circle without changing direction. As shown in Fig. 84.4, this path constitutes a circle of radius R , and part of the path is a globally optimal path inside the convoy circle.

Definition 84.2.2 The circular paths of radius R entering and exiting the convoy circle at states located on the same line as the center of the convoy circle are referred to as the optimal convoy protection paths.

Figure 84.4 shows three examples of optimal convoy protection paths, and it is straightforward to show the following key result:

Theorem 84.2.2 *Over all simple closed paths of a single UAV, the optimal convoy protection paths each maximizes the ratio of the length inside the convoy circle over its total length.*

Intuitively, an optimal convoy protection path maximizes the coverage ratio because it is the quickest path to come back to the convoy circle, always reenters optimally, and repeats as a limit cycle. And, to ensure that all the UAVs maximize their time providing convoy protection, their paths should be set to the optimal convoy protection paths such as the ones shown in Fig. 84.4. In order to achieve successful convoy protection, it is required that the UGVs are visible to at least one UAV at all time. Thus, one can establish a lower bound on the number of UAV required to provide successful convoy protection for all time based on the length of the optimal convoy protection paths.

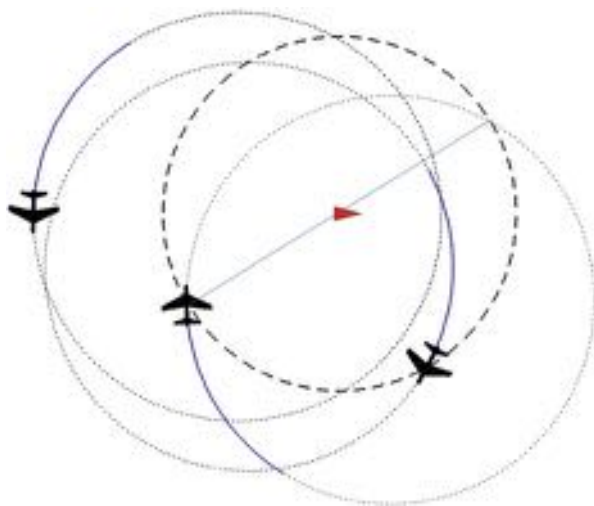


Fig. 84.4 Three optimal convoy protection paths are shown. They maximize the time spent inside the convoy circle over the time outside of the convoy circle. The *smaller dashed circle* is the convoy circle, and the *larger dotted circles* are optimal convoy protection paths. The *solid curve* is the past trajectories of the UAV. One can see that the key to coordinate the UAVs for optimal continuous convoy protection is to fly individual UAVs on optimal convoy protection paths and space them out so that there is at least one UAV inside the convoy circle for all time

Corollary 84.2.3 *Given the convoy circle of radius r for the UGVs and minimum turning radius R for the UAVs, the minimum number of UAVs needed to provide convoy protection for all time is*

$$N = \left\lceil \frac{\pi}{\arcsin(\frac{r}{R})} \right\rceil, \quad (84.19)$$

where $\lceil \cdot \rceil$ denotes the ceiling function.

Assume that there is N UAVs and they can start at an optimal initial condition $q^*(0) \in \mathbb{Q}^*$, the UAVs need to space themselves evenly in terms of the time entering the convoy circle. This can be achieved by slowing down and speeding up with respect to the other UAVs so that the i th UAV enters the convoy circle at time $\frac{2\pi R}{N}i$. This strategy is possible since the optimal paths derived for this problem remain the same for UAVs of any speed (instead of unit speed).

84.2.3 Moving Convoy Protection

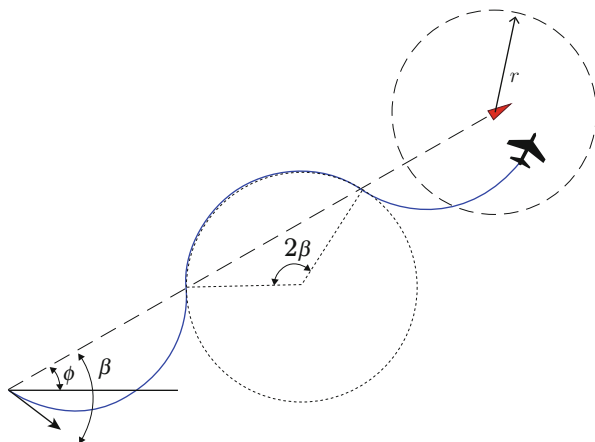
In this section, the focus is shifted to a convoy protection strategy for moving UGVs. Again, it is assumed that the location of the UGVs is represented by their centroid as a point. Instead of being static, the case is considered where this point is moving in a constant direction with a constant and bounded speed.

Denote the speed of the UGVs as V_G . The UGVs are assumed to be moving in a constant heading of angle ϕ . The UAV speed is again normalized to 1. Hence, the UAVs follow the dynamics in Eq. (84.1), and their states are denoted by $[x, y, \theta]^T$. The UAVs are assumed to be capable of flying with faster speed than the UGVs (this agrees with current state of technologies in terms of speed of unmanned ground vehicles versus UAVs). Hence, it is assumed that $V_G \leq 1$.

Now, a control strategy is proposed with a corresponding lower bound V_G^* so that if the speed of the UGVs is in this bound ($V_G \in [V_G^*, 1]$), then one UAV is guaranteed to provide convoy protection for all time. Inspired by the motion primitives defined in the static convoy protection problem, the motion of the UAV is fixed to a sequence of maximally left and right turns, that is, $\mathcal{M} = \{\mathbf{L}, \mathbf{R}, \mathbf{L}, \mathbf{R}, \dots\}$ or $\mathcal{M} = \{\mathbf{R}, \mathbf{L}, \mathbf{R}, \mathbf{L}, \dots\}$. It is assumed that the UAV and UGVs are initially on top of each other; that is, the initial x - y coordinate of the UGVs is $[x(0), y(0)]^T$. Define the angle between the heading of the UGVs and initial heading of the UAV as β . Hence, $\beta = \phi - \theta(0)$. Again, to simplify notations, it is assumed that all angles are taken modulo 2π .

The motion switches between \mathbf{L} and \mathbf{R} every time the paths of UAV and UGVs intersect. With this control strategy, the path of the UAV and the UGVs intersects every time the UAV flies for a circular arc of angle 2β . An example of the trajectory of the UAV and the UGVs is shown in Fig. 84.5. The initial motion primitive of the motion sequence \mathcal{M} depends on β . If $\beta \in [0, \pi)$, then the path of the UGVs

Fig. 84.5 Example trajectory of a UAV providing convoy protection for the UGVs with the proposed control strategy. The *solid curve* is the path of the UAV. The *dashed line* is the path of the UGVs. The *dashed circle* is the convoy circle. The angle of the circular arc for each motion primitive is 2β . In this case, $\beta \in [0, \frac{\pi}{2}]$



is to the left of the initial heading of the UAV and the first motion primitive is **L**; otherwise, the first motion primitive is **R**. The following discussion focuses on the case when $\beta \in [0, \pi)$ and the motion sequence is $\mathcal{M} = \{\mathbf{L}, \mathbf{R}, \mathbf{L}, \mathbf{R}, \dots\}$, since if $\beta \in (-\pi, 0]$, then the path of the UAV is symmetric to the path corresponding to the angle of $-\beta$.

It is desirable to control the UAV to meet the UGVs periodically. This goal can be achieved by carefully choosing the initial heading of the UAV based on the speed of the UGVs. If the UAV executes the proposed control strategy, then it flies for a circular arc of angle 2β for each motion primitive in \mathcal{M} . Assume that the UAV meets with the UGVs at the end of each motion primitive. For each motion primitive, the UAV travels for a distance of $2R\beta$ and the UGVs travel for a distance of $2R\sin(\beta)$. Since the UAV is unit speed, $V_G 2R\beta = 2R\sin(\beta)$, and therefore $V_G = \frac{\sin(\beta)}{\beta}$. In other words:

Lemma 84.2.4 Assume that the UGVs move with constant speed V_G and heading ϕ , and the UAV starts at the same position as the UGVs with the initial heading $\theta(0)$. $\beta = \phi - \theta(0)$. Assume that $\beta \in [0, \pi)$ and hence $\mathcal{M} = \{\mathbf{L}, \mathbf{R}, \mathbf{L}, \mathbf{R}, \dots\}$. Then if the UAV executes the proposed control strategy, and $V_G = \frac{\sin(\beta)}{\beta}$, then the UAV and the UGVs meet at the end of each motion primitive.

Note that if the UGVs travel with the same speed as the UAV, that is, $V_G = 1$, then from Lemma 84.2.4, it follows that $\beta = 0$. In this case, the UAV will fly exactly on top of the UGVs. Using Lemma 84.2.4, one can thus obtain the lower bound for the speed of UGVs to achieve perpetual convoy protection with the proposed strategy.

Theorem 84.2.5 Using the proposed control strategy, one UAV is sufficient to provide continuous convoy protection for all time, if V_G is bounded below by V_G^* , where

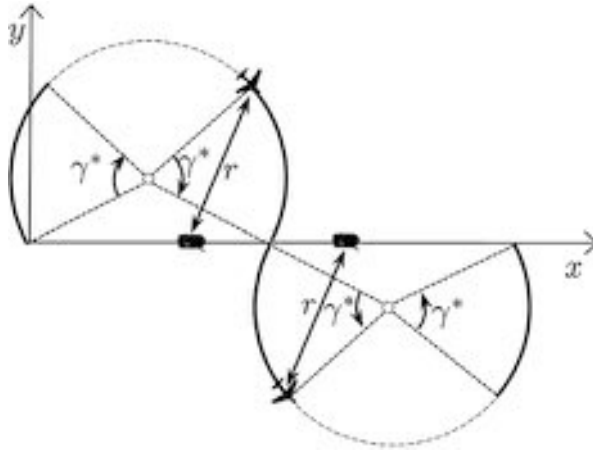


Fig. 84.6 For each execution of one motion primitive, there are two segments of the path corresponding to two circular arcs of angle γ^* , so that the distance between the UAV and the UGVs is less or equal to r when the UAV is on these segments. In this figure, the *dashed curve* is the path of the UAV, and the *solid curves* are the segments of the path in which convoy protection is provided. The UAV and the convoys are drawn at the times when the UAV enters and exits these segments

$$V_G^* = \frac{\sqrt{2rR - r^2}}{R \arccos\left(1 - \frac{r}{R}\right)}. \quad (84.20)$$

When $V_G < V_G^*$, convoy protection cannot be provided with a single UAV, and one needs to coordinate multiple UAVs to provide perpetual convoy protection. Using a similar approach as in the static convoys, case, it is possible to determine the minimum number of UAVs required. In this case, on the path of each execution of one motion primitive, there are two segments of the path when the distance between the UAV and the UGVs is less than or equal to r . Positions of UAV and UGVs and hence their distance can be uniquely characterized by the arc angle γ as $d(\gamma)$. Convoy protection is provided by one UAV for two circular arcs of angle γ^* for each execution of one motion primitive, where $d(\gamma^*) = r$. Refer to Fig. 84.6 for an example.

Similar to the multi-UAV coordination approach in the previous section, one can use a timing strategy to schedule the UAVs such that, at any time, one of the UAVs is inside the convoy circle. First, note that the minimum number of UAVs required to provide continuous convoy protection can be obtained by the following corollary, which directly follows from the fact that, for each motion primitive, the length of the path in which one UAV stays inside the convoy circle is $2R\gamma^*$, while the length of the entire path for the motion primitive is $2R\beta$.

Corollary 84.2.6 *Using the proposed control strategy, if $V_G < V_G^*$, then the minimum number of UAVs needed to provide continuous convoy protection for all*

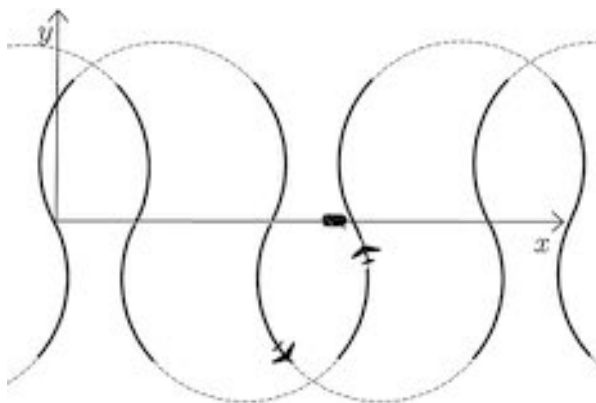


Fig. 84.7 Example of using two UAVs to provide continuous convoy protection using the proposed strategy. In this figure, the *dashed curves* are the paths of the UAVs, and the *solid curves* are the segments of the paths in which convoy protection is provided. In this case, every time one UAV exits the convoy circle, the other UAV is inside the convoy circle. This is always true if $N = \lceil \frac{\beta}{\gamma^*} \rceil = 2$ and the times when the UAVs synchronize with the UGVs are spaced out

time is $N = \lceil \frac{\beta}{\gamma^*} \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function. γ^* can be obtained by solving a nonlinear equation $d(\gamma^*) = r$, and β is obtained from V_G ($V_G = \frac{\sin(\beta)}{\beta}$).

Figure 84.7 shows how one can schedule the UAVs to provide continuous convoy protection for all time. The key is to synchronize the position of the UGVs with individual UAVs at different times, so that when one UAV exits the convoy circle, there is at least one UAV inside the convoy circle and it is on the segment of its path in which the distance to the UGVs is less or equal to r .

84.3 The Assignment Problem

Now that suitable maneuvers have been obtained for making the UAV team fly over the ground convoy, the next issue concerns what UAVs should be dispatched from the convoy in order to investigate and potentially clear threats. To address this issue, one must solve the so-called assignment problem. The assignment problem is one of the fundamental problems in combinatorial optimization. In the setting of a UAV team providing ground convoy protection to a number of UGVs, it is important that the task is allocated to the right UAV and a certain cost objective is optimized. For the convoy protection mission scenario, the UGVs are traveling along a nominal path. The task of the UAVs is to provide protection to the ground vehicles, while clearing threats as they emerge in front of the path of the UGVs.

A natural choice for the cost objective is to balance the fuel consumptions of the UAVs. In a realistic convoy protection scenario, the fuel carried by each UAV is limited. By balancing the fuel consumption among the UAVs, the duration in

which all UAVs remain operational is being maximized. Therefore, the maximum fuel consumption among the UAVs will be used as the cost function when designing the scheduling and task allocation algorithm for the UAVs (see e.g., Egerstedt 2011).

To formulate the problem in a mathematical setting, one first needs to provide a model for the overall mission of the UGVs and the possible threats. Furthermore, since the convoy protection problem is highly related to the TSP problem, as already discussed, a number of simplifying assumptions are made in order to lower the complexity of the problem. This is of paramount importance as the algorithm must run repeatedly on a computationally limited resource in order to respond to pop-up threats.

In Sect. 84.1.2, a number of assumptions were discussed, and here, these assumptions are related to their impact on the formulation of the assignment problem. Assumption A2 simply says that the UGVs follow a set path. This is a realistic assumption as the UGV is expected to follow roads or other easily traversable structures in the environment. Along the nominal convoy path, there are a number of threats that the UAVs must survey and possibly clear. The threats are classified into two categories. One category of threats are *persistent*. These are threats that the UGVs must avoid since they cannot be cleared by the UAVs. And, as per Assumption A3, one also wants to allow for the possibility of threats not being cleared if they cannot be surveyed by a UAV in time. In fact, it is assumed that the UAV-UGV team can only detect the threats up to a distance D away from the location of the UGVs, and as a consequence of Assumption A1, the threat density is such that no more than a given number M of threats can appear within this distance at any given time. But, if a threat pops up in front of the UGVs within distance D and if all available UAVs are dispatched beyond the threat, clearing of the threat may not be possible. If this is the case, they will be treated as persistent threats. In both of these cases (inspected persistent threats and threats that cannot be inspected in time), the path of the UGVs will be replanned to avoid the threats.

The other category of threats are *nonpersistent*, in which case the threats are removed after being checked by the UAVs. Regardless of the type, all threats must, if possible, be checked and surveyed by the UAVs, and the threat category cannot be established without a UAV inspecting the location of the threat. If a UAV does not get to a threat in time, it will be treated as a persistent threat and, as such, force a replanning of the UGV convoy.

In order to formulate the assignment problem, the UGV convoy vehicles are modeled as single-point masses moving along a one-dimensional (not necessarily straight) path as shown in Fig. 84.8. The UAVs are either flying alongside the UGVs (meaning they are at the same position along the path) or they can fly ahead of the UGVs. As they fly ahead of the UGVs, they do not have to follow the path. Instead, they are assumed to follow the Euclidean shortest distance to the target location. Hence, for this model, the motion of the UGVs is viewed as moving along a one-dimensional corridor which, as already discussed, allows one to establish an order among the threats which significantly cuts down on the complexity of the problem. At the same time, this is a realistic assumption that does not significantly limit the applicability of the proposed method.

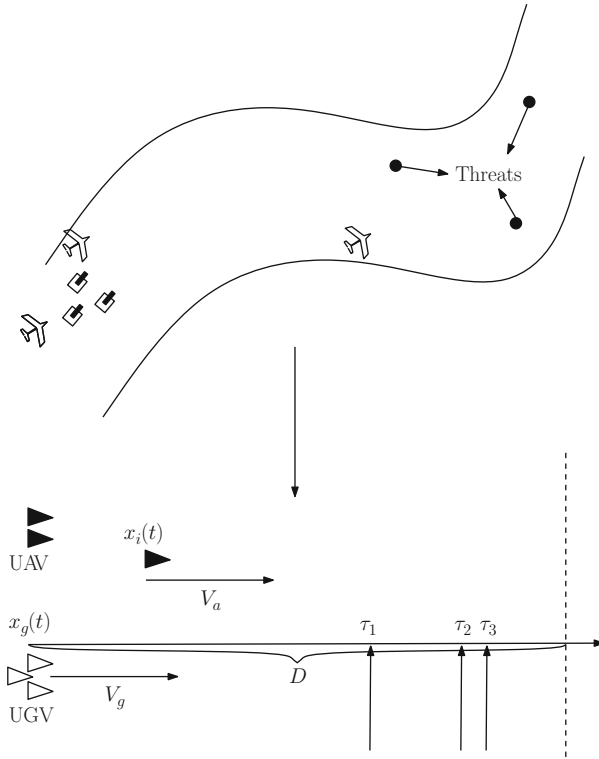


Fig. 84.8 The assignment problem model

With this model, the assignment algorithm does not consider the kinematic constraints of the UAVs and UGVs, and the UAVs are assumed to be able to change direction and turn around while checking threats. At time t , the position of the UGV team is denoted to be $x_g(t)$. The positions of the individual UAVs are denoted as $x_i(t)$, where $i \in \mathcal{I}$ denotes the index of the individual UAVs and \mathcal{I} is the set of all UAV indices.

It is assumed that the team can detect threats for a distance of D in front of the UGVs. Hence, only threats contained in this window are known to the UGV-UAV team. At time t , it is assumed that there are $N(t)$ threats in this range. The locations of threats are assumed to be fixed, and they are denoted as $\tau_j, j = 1, 2, \dots, N(t)$. The sequence of the known threats at time t is denoted as $\bar{\tau}(t) = [\tau_1, \tau_2, \dots, \tau_{N(t)}]$. This known threat sequence has time-varying length $N(t)$. $N(t)$ increases when a pop-up threat is detected and decreases when a threat is cleared. Note that all threats (pop-up or not) are in front of the UGVs on the intended path.

Furthermore, let the total number of threats be denoted by N (where, for notational convenience, the explicit dependence on t is suppressed, as will be done in the remaining chapter), and the sequence of all threats is denoted as

$\bar{\tau} = [\tau_1, \tau_2, \dots, \tau_N]$. The overall mission of the UAVs is to clear all the threats. If any of the threats are hostile after being checked out by a UAV, then the path taken by the UGVs is replanned, the problem is re-initialized, and the assignment algorithm is restarted.

Naturally, when the UAV is checking a threat, it should fly with a higher speed than when it is cruising along with the UGVs. The velocity of the UGV team is assumed to be V_g . It is assumed that when the UAVs are flying alongside the UGVs, they fly with the same speed V_g . However, when they are assigned to clear a threat or coming back after clearing a threat, they fly with the speed V_a . Throughout this chapter, it is assumed that $V_a > V_g$.

Let the starting time of the overall mission be denoted by t_0 and the time when all N threats are cleared by T . It should be noted that since the aim is to develop a real-time algorithm to select and assign the UAVs to clear the threats, the only information available at time t is the $N(t)$ threats within range D , and the assignment algorithm only uses this information to make decisions.

The problem is to devise algorithms to dispatch the UAVs to clear all upcoming threats while balancing the fuel consumptions of the UAVs. The accumulated fuel consumption of i th UAV at time t is denoted as $f_i(t)$. $f_i(t)$ is considered to be the state of the UAVs, and they are known. It should be noted that at the beginning of the mission, the accumulated fuel consumption of all the UAVs is 0; hence, $f_i(t_0) = 0, \forall i$. With this information, one can formulate the UAV assignment problem.

Problem 84.3.1 *Design an algorithm that assigns UAVs such that the maximum fuel consumption among the UAVs is minimized when all threats $\bar{\tau}$ are cleared. That is, solve the optimization problem:*

$$\min_{\Pi} \max_{j \in \mathcal{I}} \{f_j(T)\}, \quad (84.21)$$

where Π is a assignment algorithm that maps from time to \mathcal{I} .

84.3.1 Selection Policy and Task Allocation Algorithm

In this section, an algorithm is introduced that assigns the UAVs based on their current level of accumulated fuel consumption. The algorithm uses an optimization-based selection policy that selects (and assigns) one threat to one of the UAVs when an event triggers. Events that trigger the reevaluation of the selection policy can be one of the following:

1. A threat is cleared by any of the assigned UAV.
2. A pop-up threat is detected.

In other words, the selection policy is only reevaluated when the known threat sequence $\bar{\tau}(t)$ is modified. In order to allow real-time execution of the algorithm,

the optimization problem needed for the selection policy is chosen so that it can be solved quickly online. In order to reduce the problem from NP-hard to polynomial complexity, only the closest threat is considered when an event triggers. This does not cause inefficiency in clearing multiple threats since each time a threat is cleared, the selection policy is reevaluated. The position of the closest threat is denoted by $\tau \in \mathbb{R}^2$. Hence,

$$\tau = \min_{j \in \{1, 2, \dots, N(t)\}} \tau_j. \quad (84.22)$$

The selection policy is a function that maps time t to the set of the UAVs (\mathcal{I}), and it is the solution of an optimization problem. This optimization policy is based on computing the fuel consumption associated with letting a UAV clear the target threat (τ) and returning to the convoy. Even though this may actually not be what is done (as new threats are constantly reconsidered), the resulting algorithm minimizes the accumulated fuel consumption over all other assignments assuming that the assigned UAV will return to the convoy after the threat is inspected. Here, it is taken into account that the UGVs are moving constantly along the path with speed V_g and that the closest threat τ may be a pop-up threat.

As fuel consumption is at the heart of the assignment costs, one needs to establish an appropriate model for this consumption. Although the developed algorithm must be quite general and should not rely on any particular choice of fuel model, the fuel consumption rate at any given moment is a nonlinear function of the UAV's speed, weight, and altitude. Therefore, the fuel consumption rate of the i th UAV at time t is given by

$$\frac{df_i(t)}{dt} = Q(v(t), w(t), h(t)), \quad (84.23)$$

where $v(t)$, $w(t)$, and $h(t)$ are the UAVs speed, gross weight, and altitude at time t , respectively.

When an event triggers at time t , the following optimization problem is solved:

$$P_1(t) = \arg \min_{i \in \mathcal{I}} \left\{ \max_{j \in \mathcal{I}} \left\{ f_j(t) + \begin{cases} Q(V_a, w^*, h^*)T_i(t), & \text{if } j = i \\ Q(V_a, w^*, h^*)S_j(t) + Q(V_g, w^*, h^*)(T_i(t) - S_j(t)), & \text{if } j \neq i \end{cases} \right\} \right\}. \quad (84.24)$$

The optimization problem is a min-max problem. $T_i(t)$ represents the time for i th UAV to clear the threat and return to the UGVs (with speed V_a). If the j th UAV is not assigned, $S_j(t)$ represents the time it takes for it to return to the UGVs (with speed V_a). $T_i(t) - S_j(t)$ therefore represents the time for the j th UAV to fly alongside the UGVs with speed V_g until the assigned UAV flies back. If the UAV j is flying alongside the UGV at time t , then $x_j(t) = x_g(t)$ and $S_j(t) = 0$. Hence, the quantity

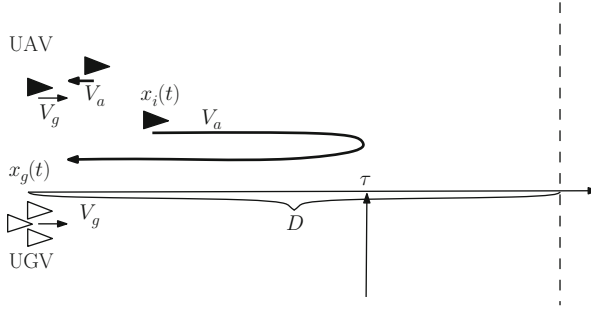


Fig. 84.9 The operation of the selection policy. In this example, the threat is not a pop-up

$$f_j(t) + \begin{cases} Q(V_a, w^*, h^*)T_i(t), & \text{if } j = i \\ Q(V_a, w^*, h^*)S_j(t) + Q(V_g, w^*, h^*)(T_i(t) - S_j(t)), & \text{if } j \neq i \end{cases} \quad (84.25)$$

represents the accumulated fuel consumption of the j th UAV, assuming UAV i is assigned.

It is important to note here that the convoy protection problem formulation, along with the corresponding task assignment algorithm presented in this section, is independent of how the quantities $T_i(t)$ and $S_j(t)$ are computed for a specific operational scenario. In the experiments, these quantities were computed assuming piecewise-linear convoy path defined by a series of waypoints. As shown in Fig. 84.9, $P_1(t)$ assigns the UAV so that the UAV with most accumulated fuel consumed after the clearance of the threat is minimized.

Min-max problems are generally hard to solve, and the complexity of computing $P_1(t)$ is $O(|\mathcal{I}|^2)$, where $|\mathcal{I}|$ is the number of UAVs. However, under certain design choices of UAV speeds V_a and V_g , weight w^* , and altitude h^* , one can reduce the optimization problem to one that has $O(|\mathcal{I}|)$ (linear) complexity. The assumption needed for this is that it is more costly in terms of fuel consumption to be inspecting a threat than to remain with the convoy. And, under this assumption, one can rewrite the assignment problem in the following manner:

Lemma 84.3.1 *If UAV speeds V_a and V_g , gross weight w^* , and altitude h^* are chosen such that*

$$Q(V_a, w^*, h^*) > Q(V_g, w^*, h^*), \quad (84.26)$$

then the optimization problem $P_1(t)$ can be solved by solving another optimization problem:

$$P(t) = \arg \min_{i \in \mathcal{I}} \{f_i(t) + Q(V_a, w^*, h^*)T_i(t)\}. \quad (84.27)$$

Hence, $P(t) = P_1(t)$.

This reformulation is useful in that it actually allows one to solve the assignment problem directly. To describe in words, $P(t)$ assigns the UAV so that, after the threat

is cleared and returned to the UGVs, the total fuel consumed for the assigned UAV is minimum over all the UAVs. The projected time for the clearance of the threat is $T_i(t)$, where i denotes the assigned UAV. It should be noted that during this time, the UGVs have moved forward by $V_g T_i(t)$ distance.

Based on this reformulation, one can in fact formulate the corresponding task allocation algorithm which uses the above selection policy.

Algorithm: Task Allocation Algorithm

Initialize: Set t_0 .

Iterate: until end of mission.

If an event is triggered:

1. Update $\bar{\tau}(t) = [\tau_1, \tau_2, \dots, \tau_{N(t)}]$.
2. Find $\tau = \min_{j \in \{1, 2, \dots, N(t)\}} \tau_j$.
3. Solve $i^* = P(t)$ in Eq. (84.27) and use the solution i^* as the assignment for the i th UAV to clear τ .

Wait until another event is triggered.

An illustration of the intended operation is depicted in Fig. 84.10a–f. The UAVs will follow the ground vehicles in a set formation. If a threat is detected, a UAV will be assigned that will balance fuel usage among the UAVs, to clear the threat and return to the formation.

84.4 Experimental Results

To demonstrate the practical aspects of the proposed algorithms as well as to show that they can indeed be successfully deployed in an operational environment characterized by numerous computational and communications limitations, the algorithms were implemented on a test bed consisting of both UAVs and UGVs.

The experiment consists of a mission that incorporates all details highlighted in this chapter. The mission is structured as follows:

1. The UGVs are given a set of waypoints which define a path to be taken through the (possibly hostile) environment. All UAVs are assigned to protect the convoy by flying over it while maintaining some spacing from one another.
2. Once a possible threat has been identified, the projected fuel consumption for each UAV is computed, which in turn dictates which UAV to dispatch so as to balance fuel consumption.
3. The chosen UAV flies over to visit the threat, while the remaining UAVs remain with the convoy.
4. After the threat is neutralized (cleared), the assigned UAV returns to protect the convoy.
5. In the event that a threat cannot be neutralized (is persistent), the UAV signals the convoy to replan its path to avoid the threat.

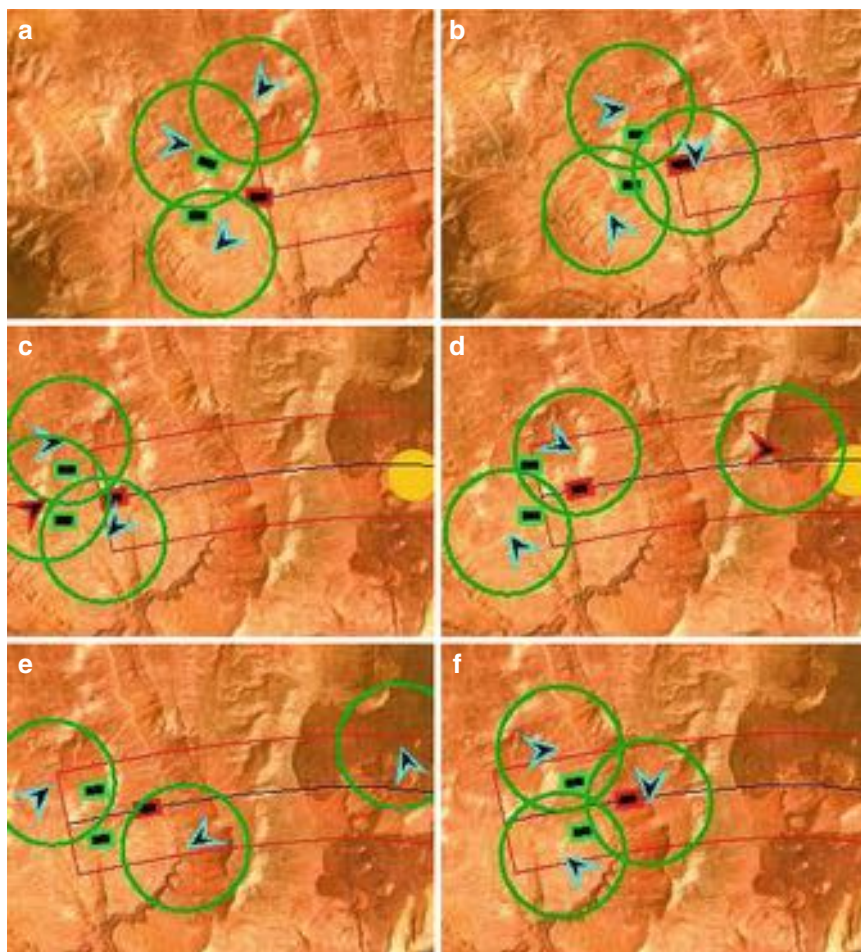


Fig. 84.10 Simulation of three UAVs executing the convoy protection algorithm. (N is the number of UAVs along the UGV convoy.) (a) UAVs forming a formation of $N = 3$. (b) UAVs in $N = 3$ formation. (c) A UAV is splitting. (d) UAVs form new $N = 2$ formation. (e) A UAV is rejoining. (f) UAVs return to a formation of $N = 3$

The plots in Fig. 84.12 show the recorded trajectories of the UAVs and convoy during the experiment. In particular, Fig. 84.12a shows both UAVs performing convoy protection by flying above the UGVs as they follow their intended path. Notice that the UAVs maintain a safe separation from one another. A photograph of this operation is shown in Fig. 84.11a.

Figure 84.12c shows the UAVs detecting a persistent threat which blocks the intended path of the convoy. The UAVs execute the task allocation algorithm to determine which UAV should be dispatched so as to balance fuel consumption. When the dispatched UAV determines that the threat is persistent, the UAV informs



Fig. 84.11 Photos showing the convoy protection and threat neutralization as carried out by the hardware platform. (a) Two UAVs protect the convoy while maintaining spacing from one another. (b) One UAV is dispatched to visit a threat, while the other remains with the convoy

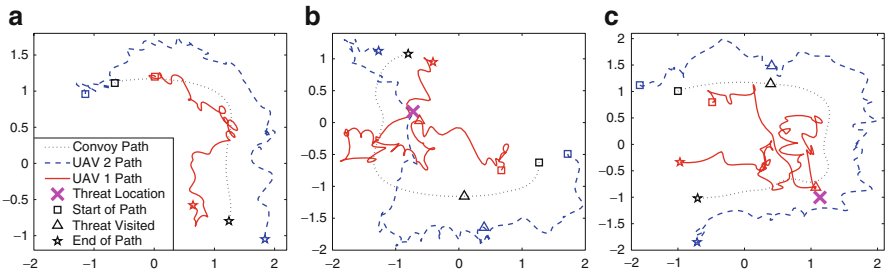


Fig. 84.12 Plots showing the trajectories of UAVs and the convoy which it protects as both a nonpersistent and persistent threat are encountered. (a) Two UAVs perform convoy protection on UGVs. (b) A UAV is dispatched to clear a nonpersistent threat. (c) A UAV identifies a persistent threat and the convoy replans its path

the convoy that it must replan its path so as to avoid the threat. Afterward, the UAV returns to the convoy, and the UGVs proceed in following the recomputed path to safely reach their destination.

To further illustrate effects of convoy path replanning, Fig. 84.13a shows the originally intended path of the convoy corresponding to the trajectories in Fig. 84.12a, b. Meanwhile, Fig. 84.13b shows the recomputed path taken by the convoy so as to avoid the persistent threat.

To validate that the task allocation algorithm performs as expected, a separate experiment was conducted in which 16 threats were presented to the convoy over an extended period of time. Figure 84.14 shows the fuel consumption of the two UAVs over this time and marks when a UAV is dispatched to clear a threat. As seen in the plot, the algorithm successfully determines which UAV to dispatch when a threat is encountered such that the fuel consumption is balanced among the two UAVs over time. This oftentimes results in the same UAV being dispatched multiple times consecutively so as to clear threats that are located within its vicinity.

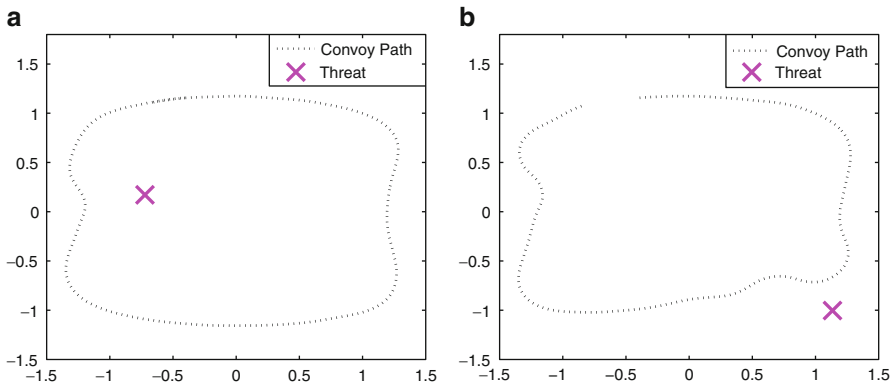


Fig. 84.13 Plots showing the difference in path taken by the ground convoy before and after being informed about the presence of a persistent threat. (a) The original intended path taken by the convoy, corresponding to Fig. 84.12a, b. (b) The recomputed path of the convoy upon being informed of a persistent threat, corresponding to Fig. 84.12c

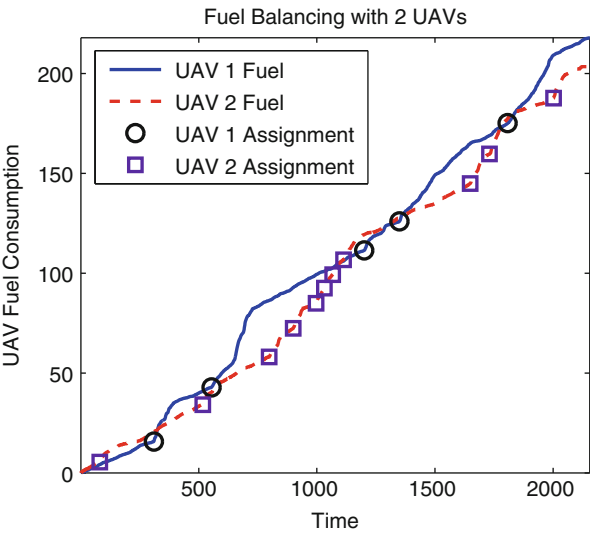


Fig. 84.14 Plot showing the fuel consumption of the two UAVs while performing convoy protection

References

P. Agarwal, T. Biedl, S. Lazard, S. Robbins, S. Suri, S. Whitesides, Curvature-constrained shortest paths in a convex polygon, in *Proceedings of the ACM Symposium on Computational Geometry*, Minneapolis, 1998, pp. 392–401

A. Arsie, K. Savla, E. Frazzoli, Efficient routing algorithms for multiple vehicles with no explicit communications. *IEEE Trans. Autom. Control* **54**(10), 2302–2317 (2009)

- G. Arslan, J.R. Marden, J.S. Shamma, Autonomous vehicle-target assignment: a game theoretical formulation. *ASME J. Dyn. Syst. Meas. Control* **129**(5), 584–596 (2007)
- R.W. Beard, T.W. McLain, M.A. Goodrich, E.P. Anderson, Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans. Rob. Autom.* **18**(6), 911–922 (2002)
- R. Beard, T. McLain, D. Nelson, D. Kingston, Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. *IEEE Proc.* **94**(7), 1306–1324 (2006)
- C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, G.J. Pappas, Symbolic planning and control of robot motion: state of the art and grand challenges. *IEEE Rob. Autom. Mag.* **14**(1), 61–70 (2007)
- J. Boissonnat, A. Cérézo, J. Leblond, Shortest paths of bounded curvature in the plane. *J. Intell. Rob. Syst.* **11**(1–2), 5–20 (1994)
- R.E. Burkard, Selected topics in assignment problems. *Discret. Appl. Math.* **123**, 257–302 (2002)
- H. Chitsaz, S.M. LaValle, Time-optimal paths for a Dubins airplane, in *IEEE Conference on Decision and Control*, New Orleans, 2007, pp. 2379–2384
- D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, R. Fierro, Decentralized cooperative control – a multivehicle platform for research in networked embedded systems. *IEEE Control Syst.* **27**(3), 58–78 (2007)
- X.C. Ding, A. Rahmani, M. Egerstedt, Optimal multi-UAV convoy protection, in *International Conference on Robot Communication and Coordination*, Odense, 2009
- X.C. Ding, A. Rahmani, M. Egerstedt, Multi-UAV convoy protection: an optimal approach to path planning and coordination. *IEEE Trans. Rob.* **26**(2), 256–268 (2010)
- L. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**, 497–516 (1957)
- L. Dubins, On plane curves with curvature. *Pac. J. Math.* **11**(2), 471–481 (1961)
- M. Egerstedt, *Interface Control Document for Heterogeneous Multi-vehicle Ground Convoy Protection* (Georgia Institute of Technology, Atlanta, 2011)
- E. Frazzoli, M.A. Dahleh, E. Feron, Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Trans. Rob.* **21**(6), 1077–1091 (2005)
- M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979)
- A.R. Girard, A.S. Howell, J.K. Hedrick, Border patrol and surveillance missions using multiple unmanned air vehicles, in *IEEE Conference on Decision and Control*, Atlantis, Bahamas, 2004, pp. 620–625
- B. Grocholsky, J. Keller, V. Kumar, G. Pappas, Cooperative air and ground surveillance: a scalable approach to the detection and localization of targets by a network of UAVs and UGVs. *IEEE Rob. Autom. Mag.* **13**(3) 16–26 (2006)
- K. Hauser, T. Bretl, K. Harada, J.C. Latombe, Using motion primitives in probabilistic sample-based planning for humanoid robots, in *In Workshop on the Algorithmic Foundations of Robotics (WAFR)*, New York, 2006
- M. Ji, S. Azuma, M. Egerstedt, Role-assignment in multi-agent coordination. *Int. J. Assist. Rob. Mechatron.* **7**(1), 32–40 (2006)
- D.E. Kirk, *Optimal Control Theory, An Introduction* (Dover, Mineola, 2004)
- H.W. Kuhn, The hungarian method for the assignment problem. *Naval Res. Logist. Q.* **2**, 83–97 (1955)
- S.M. Lavalle, *Planning Algorithms* (Cambridge University Press, New York, 2006)
- J. Lee, R. Huang, A. Vaughn, X. Xiao, J.K. Hedrick, M. Zennaro, R. Sengupta, Strategies of path-planning for a UAV to track a ground vehicle, in *The Second Annual Symposium on Autonomous Intelligent Networks and Systems*, Menlo Park, 2003
- T.G. McGee, S. Spry, J.K. Hedrick, Optimal path planning in a constant wind with a bounded turning rate, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, 2005
- B. Moore, K. Passino, Distributed balancing of AAVs for uniform surveillance coverage, in *IEEE Conference on Decision and Control*, Seville, 2005, pp. 7060–7065

- S. Morris, M. Holden, Design of micro air vehicles and flight test validation, in *Proceedings of the Fixed, Flapping and Rotary Wing Vehicles at Very Low Reynolds Numbers*, Notre Dame, 2000, pp. 153–176
- Northrop Grumman, Heterogeneous Aerial Reconnaissance Team (HART), http://en.wikipedia.org/wiki/Heterogeneous_Aerial_Reconnaissance_Team
- P.M. Pardalos, F. Rendl, H. Wolkowicz, The quadratic assignment problem, in *The Quadratic Assignment and Related Problems*, ed. by P.M. Pardalos, H. Wolkowicz. DIMACS Series, vol. 16 (American Mathematical Society, Providence, 1994), pp. 1–41.
- M. Pavone, E. Frazzoli, F. Bullo, Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. *IEEE Trans. Autom. Control* **54**(10), 2302–2317 (2009)
- S. Ponda, J. Redding, H.L. Choi, J.P. How, M. Vavrina, J. Vian, Decentralized planning for complex missions with dynamic communication constraints, in *American Control Conference*, Baltimore, Maryland, 2010
- H. Psaraftis, Dynamic vehicle routing problems, in *Vehicle Routing: Methods and Studies*, ed. by B. Golden, A. Assad. Studies in Management Science and Systems (Elsevier, Amsterdam, 1988)
- M. Quigley, M.A. Goodrich, S. Griffiths, A. Eldredge, R.W. Beard, Target acquisition, localization, and surveillance using a fixed-wing, mini-UAV and gimbaled camera. *International Conference on Robotics and Automation*, Barcelona, 2005, pp. 18–221
- J. Reeds, L. Shepp (1990) Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **145**(2), 367–393
- A. Richards, J. Bellingham, M. Tillerson, J. How, Coordination and control of multiple UAVs, in *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, Monterey, 2002
- K. Savla, E. Frazzoli, F. Bullo, Traveling salesperson problems for the Dubins vehicle. *IEEE Trans. Autom. Control* **53**(6), 1378–1391 (2008)
- C. Schumacher, P.R. Chandler, S.J. Rasmussen, D. Walker, Task allocation for wide area search munitions with variable path length, in *Proceedings of the American Control Conference*, Denver, 2003, pp. 3472–3477
- P. Souèeres, J. Boissonnat, Optimal trajectories for nonholonomic mobile robots, in *Robot Motion Planning and Control* (Springer, London, 1998), pp. 93–170
- P. Souèeres, J. Laumond, Shortest paths synthesis for a car-like robot. *IEEE Trans. Autom. Control* **41**(5), 672–688 (1996)
- S.C. Spry, A.R. Girard, J.K. Hedrick, Convoy protection using multiple unmanned air vehicles: organization and coordination, in *American Control Conference*, Portland, 2005
- G.C. Walsh, R. Montgomery, S. Sastry, Optimal path planning on matrix Lie groups. *IEEE Conf. Decis. Control* **2**(14–16), 1258–1263 (1994)

John J. Enright, Emilio Frazzoli, Marco Pavone, and Ketan Savla

Contents

| | | |
|--------|--|------|
| 85.1 | Introduction | 2080 |
| 85.2 | Approaches for UAV Routing in Dynamic Environments | 2081 |
| 85.2.1 | Static and Dynamic Vehicle Routing | 2082 |
| 85.2.2 | Heuristic Algorithms | 2083 |
| 85.2.3 | Online Algorithms | 2083 |
| 85.2.4 | Spatial Queueing Theory | 2084 |
| 85.3 | Spatial Queueing Theory: The Single-Server Case | 2085 |
| 85.3.1 | UAV Routing with No Motion Constraints and Unlimited Sensing | 2086 |
| 85.3.2 | UAV Routing with Motion Constraints and Unlimited Sensing | 2089 |
| 85.3.3 | UAV Routing with Limited Sensing Capabilities | 2092 |
| 85.4 | Spatial Queueing Theory: The Multi-server Case | 2094 |
| 85.4.1 | Multi-UAV Routing with No Motion Constraints and Unlimited Sensing | 2095 |
| 85.4.2 | Multi-UAV Routing with Motion Constraints and Unlimited Sensing | 2098 |
| 85.4.3 | Multi-UAV Routing with Limited Sensing Capabilities | 2103 |

J.J. Enright (✉)

Kiva Systems, North Reading, MA, USA

e-mail: jenright@kivasystems.com

E. Frazzoli

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

e-mail: frazzoli@mit.edu

M. Pavone

Aeronautics and Astronautics Department, Stanford University, Stanford, CA, USA

e-mail: pavone@stanford.edu

K. Savla

Sonny Astani Department of Civil and Environmental Engineering, University of Southern California, Los Angeles, CA, USA

e-mail: ksavla@usc.edu

85.5 Conclusions 2104

Appendix A 2104

 A.1 The Continuous Multi-median Problem 2104

Appendix B 2105

 B.1 The Euclidean Traveling Salesman Problem 2106

References 2107

Abstract

Recent years have witnessed great advancements in the science and technology for unmanned aerial vehicles (UAVs), for example, in terms of autonomy, sensing, and networking capabilities. This chapter surveys algorithms on task assignment and scheduling for one or multiple UAVs in a dynamic environment, in which targets arrive at random locations at random times, and remain active until one of the UAVs flies to the target’s location and performs an on-site task. The objective is to minimize some measure of the targets’ activity, for example, the average amount of time during which a target remains active. The chapter focuses on a technical approach that relies upon methods from queueing theory, combinatorial optimization, and stochastic geometry. The main advantage of this approach is its ability to provide analytical estimates of the performance of the UAV system on a given problem, thus providing insight into how performance is affected by design and environmental parameters, such as the number of UAVs and the target distribution. In addition, the approach provides provable guarantees on the system’s performance with respect to an ideal optimum. To illustrate this approach, a variety of scenarios are considered, ranging from the simplest case where one UAV moves along continuous paths and has unlimited sensing capabilities, to the case where the motion of the UAV is subject to curvature constraints, and finally to the case where the UAV has a finite sensor footprint. Finally, the problem of cooperative routing algorithms for multiple UAVs is considered, within the same queueing-theoretical framework, and with a focus on control decentralization.

85.1 Introduction

This chapter discusses current solution approaches for the design of cooperative control and task allocation strategies for networks of unmanned aerial vehicles (UAVs). The focus is on *uncertain* and *dynamically changing environments*, in which new task requests are generated in real time, and on routing algorithms with performance guarantees, as opposed to heuristic algorithms.

As a motivating example, consider the following scenario: a team of unmanned aerial vehicles (UAVs) is responsible to investigate possible threats over a region of interest. As possible threats are detected, by intelligence, high-altitude, or orbiting platforms, or by ground sensor networks, one of the UAVs must visit its location and investigate the cause of the alarm in order to enable an appropriate response if necessary. Performing this task may require the UAV not only to fly to the possible threat’s location but also to spend additional time on site. The objective

is, in general, to minimize the average time between the appearance of a possible threat and the time one of the UAVs completes the close-range inspection task. In a variation of this problem, which will be referred to as persistent patrolling, the UAVs must detect possible threats using limited-range onboard sensors. Other variations may include priority levels or time windows during which the inspection task must be completed.

In order to perform the required mission, the UAVs (or, more in general, mission control) need to repeatedly solve three *coupled* decision-making problems:

1. *Task allocation among the UAVs*: Which UAV shall pursue each task? What policy is used to assign tasks to UAVs?
2. *Service scheduling for each UAV*: Given the list of tasks to be pursued, what is the most efficient ordering of these tasks?
3. *Loitering paths*: What should UAVs without pending assignments do?

In general, the combined problem, which one can refer to as dynamic vehicle routing, falls within the class of heterogeneous, stochastic (possibly distributed) decision-making problems with uncertain information, with additional complexity stemming from the differential and algebraic constraints on the UAV motion and the local sensing of the environment. This problem is generally *intractable*, and solution approaches have been devised that look either at heuristics algorithms or at approximation algorithms with some guarantee on their performance.

The chapter is structured as follows. Section 85.2 presents an overview of current solution approaches for UAV routing in uncertain and dynamic environments. First, the differences and commonalities between static and dynamic UAV routing problems are discussed. Then, some of the main classes of algorithms used in the relevant literature are introduced, namely, heuristic algorithms (without performance guarantees), as well as online algorithms and spatial queueing theory (which provide guaranteed approximations to optimal performance). Section 85.3 considers in some detail the application of spatial queueing theory to some prototypical scenarios, involving a single UAV. The main motivation is to show how to model a specific UAV routing problem within this framework and how to solve it. Section 85.4 considers the same scenarios, extending the theory to the case of multiple UAVs. Finally, Sect. 85.5 summarizes the chapter.

85.2 Approaches for UAV Routing in Dynamic Environments

The objective of this section is to first discuss the differences between static and dynamic environments and then to present a broad overview of current solution approaches. Broadly speaking, there are three main approaches available in the literature to tackle such dynamic vehicle routing problems. The first approach relies on heuristic algorithms. In the second approach, called “online algorithms,” routing policies are designed to minimize the worst-case ratio between their performance and the performance of an optimal offline algorithm which has a priori knowledge of the entire input sequence. In the third approach, the routing problem is embedded within the framework of queueing theory, and routing policies are designed to minimize typical queueing-theoretical cost functions such as the expected time

the tasks remain in the queue. Since the generation of tasks and motion of the vehicles is within an Euclidean space, one can refer to this third approach as spatial queueing theory.

85.2.1 Static and Dynamic Vehicle Routing

In the recent past, considerable efforts have been devoted to the problem of how to cooperatively assign and schedule tasks that are defined over an extended geographical area (Arslan et al. 2007; Alighanbari and How 2008; Beard et al. 2002; Moore and Passino 2007; Smith and Bullo 2009). In these papers, the main focus is in developing distributed algorithms that operate with knowledge about the task locations and with limited communication between robots. However, the underlying mathematical model is *static*, in that no new tasks arrive over time, and fits within the framework of the static vehicle routing problem, whereby (i) a team of m vehicles is required to service a set of n tasks in a two-dimensional space; (ii) each task requires a certain amount of on-site service; (iii) the goal is to compute a set of routes that minimizes the cost of servicing the tasks; see Toth and Vigo (2001) for a thorough introduction to this problem. In general, most of the available literature on routing for robotic networks focuses on static environments and does not properly account for scenarios in which dynamic, stochastic, and adversarial events take place.

The problem of planning routes through service tasks that arrive *during* a mission execution is known as the “dynamic vehicle routing problem” (abbreviated as the DVR problem in the operations research literature). There are two key differences between static and dynamic vehicle routing problems. First, planning algorithms should actually provide *policies* (in contrast to preplanned routes) that prescribe how the routes should evolve as a function of those inputs that evolve in real time. Second, dynamic tasks (i.e., tasks that arrive and vary over time) add *queueing phenomena* to the combinatorial nature of vehicle routing. In such a dynamic setting, it is natural to focus on steady-state performance instead of optimizing the performance for a single task. Additionally, system stability in terms of the number of waiting tasks is an issue to be addressed (Fig. 85.1).

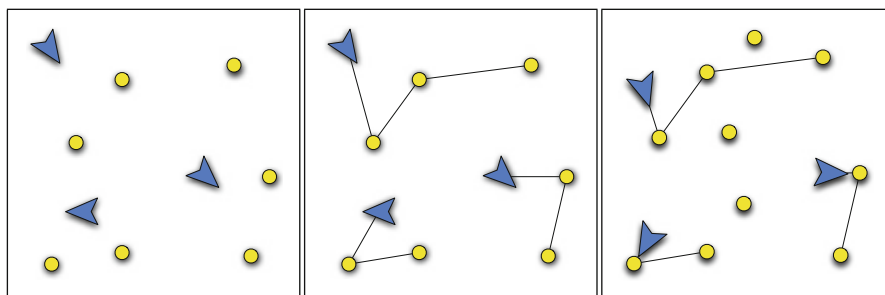


Fig. 85.1 An illustration of *dynamic routing problems* for a robotic system. From left to right: (i) tasks are generated, (ii) vehicles are assigned to tasks and select routes, (iii) new tasks appear, requiring an update to assignments and routes

85.2.2 Heuristic Algorithms

A naïve, yet reasonable approach to devise an heuristic algorithm (i.e., an algorithm without performance guarantees) would be to adapt classic queueing policies to spatial queueing systems. However, perhaps surprisingly, this adaptation is not at all straightforward. For example, policies based on a First-Come First-Served discipline, whereby tasks are fulfilled in the order in which they arrive, are unable to stabilize the system for all stabilizable task arrival rates, in the sense that under such policies, the average number of tasks grows over time without bound, even though there exist other policies that would maintain the number of tasks uniformly bounded (Bertsimas and van Ryzin 1991).

The most widely applied approach is to combine static routing methods (e.g., VRP-like methods or heuristic methods such as nearest neighbor or genetic algorithms) and sequential re-optimization, where the re-optimization horizon is chosen heuristically. (A similar approach, incidentally, is at the core also of the approximation algorithms presented in the following sections.) However, the joint selection of a static routing method and of the re-optimization horizon in presence of UAV and task constraints (e.g., differential motion constraints, or task priorities) makes the application of this approach far from trivial. First, one can show that an erroneous selection of the re-optimization horizon can lead to pathological scenarios where no task *ever* receives service (Pavone 2010). Second, direct application of VRP-like methods might lead to infeasible paths for vehicles with differential motion constraints. Additionally, performance criteria in dynamic settings commonly differ from those of the corresponding static problems. For example, in a dynamic setting, the time needed to complete a task may be a more important factor than the total vehicle travel cost.

85.2.3 Online Algorithms

An online algorithm is one that operates based on input information available up to the current time. Thus, these algorithms are designed to operate in scenarios where the entire input is not known at the outset, and new pieces of the input should be incorporated as they become available. The distinctive feature of the online algorithm approach is the method used to evaluate an algorithm's performance, which is called *competitive analysis* (Sleator and Tarjan 1985). In competitive analysis, the performance of an online algorithm is compared to the performance of a corresponding offline algorithm (i.e., an algorithm that has a priori knowledge of the entire input) in the worst-case scenario. Specifically, an online algorithm is c -competitive if its cost on *any* problem instance is at most c times the cost of an optimal offline algorithm:

$$\text{Cost}_{\text{online}}(I) \leq c \text{Cost}_{\text{optimal offline}}(I), \quad \forall \text{ problem instances } I.$$

In the recent past, several dynamic vehicle routing problems have been successfully studied in this framework, under the name of the online traveling repairman

problem (Irani et al. 2004; Jaillet and Wagner 2006; Krumke et al. 2003), and many interesting insights have been obtained. However, the online algorithm approach has some disadvantages. First, competitive analysis is a *worst-case* analysis; hence, the results are often overly pessimistic for normal problem instances. Moreover, in many applications, there is some probabilistic problem structure (e.g., distribution of the inter-arrival times, spatial distribution of future tasks, distribution of on-site service times), that can be advantageously exploited by the vehicles. In online algorithms, this additional information is not taken into account. Second, competitive analysis is used to bound the performance relative to the optimal offline algorithm, and thus it does not give an absolute measure of performance. In other words, an optimal online algorithm is an algorithm with minimum “cost of causality” in the worst-case scenario, but not necessarily with the minimum worst-case cost. Finally, many important real-world constraints for DVR, such as time windows, priorities, differential constraints on vehicle’s motion, and the requirement of teams to fulfill a task, “have so far proved to be too complex to be considered in the online framework” (Golden et al. 2008, p. 206). Some of these drawbacks have been recently addressed by Van Hentenryck et al. (2009) where a combined stochastic and online approach is proposed for a general class of combinatorial optimization problems and is analyzed under some technical assumptions.

85.2.4 Spatial Queueing Theory

Spatial queueing theory embeds the dynamic vehicle routing problem within the framework of queueing theory and overcomes some of the limitations of the online algorithm approach; in particular, it allows to take into account several real-world constraints, such as time constraints and priorities. The name *spatial queueing theory* stems from the fact that the generation of the tasks and the motion of the servers (i.e., UAVs) happens in a metric space. This chapter concentrates on an algorithmic approach to spatial queueing theory whose objective is to *synthesize* an efficient control policy, whereas in standard queueing theory, the objective is usually to *analyze* the performance of a specific policy. Within this context, an efficient policy is one whose *expected* performance is either optimal or within a constant factor of the optimum. Specifically, the expected performance of a policy is the expected value of the performance over all possible inputs (i.e., task arrival sequences). A policy performs within a constant factor κ of the optimum if the ratio between the policy’s expected performance and the optimal expected performance is upper bounded by κ .

In order to make the model tractable, tasks are usually considered “statistically independent,” and their arrival process is assumed stationary (with possibly unknown parameters). These assumptions, however, can be unrealistic in some scenarios, in which case the online algorithms approach may represent a better alternative. Pioneering work in this context is that of Bertsimas and van Ryzin (1991, 1993a, b), who introduced queueing methods to solve the simplest DVR problem

(a vehicle moves along straight lines and visits tasks whose time of arrival, location, and on-site service are stochastic; information about task location is communicated to the vehicle upon task arrival); see also the earlier related work (Psaraftis 1980). Recently, by integrating ideas from dynamics, combinatorial optimization, teaming, and distributed algorithms, this approach has been applied to scenarios with complex models for the tasks such as time constraints, service priorities, and translating tasks; problems concerning robotic implementation such as adaptive and decentralized algorithms, complex vehicle dynamics, limited sensing range, and team forming; and even integration of humans in the design space; see Bullo et al. (2011) and references therein.

An interesting feature of this approach is that the performance analysis of these algorithms usually yields scaling laws for quality of performance in terms of mission parameters. These scaling laws can serve as useful guidelines for operators to select mission parameters when feasible (e.g., number of UAVs, sensing range) to provide a desired quality of service.

85.3 Spatial Queueing Theory: The Single-Server Case

This section presents the basic ideas and tools for an algorithmic approach to spatial queueing theory (more details can be found in, e.g., Bullo et al. 2011). This approach consists of three main steps, namely, development of a spatial queueing model, establishment of fundamental limitations of performance, and design of algorithms with performance guarantees. More specifically, the formulation of a model entails detailing four main aspects:

1. A model for the *dynamic* component of the environment: This is usually achieved by assuming that new events are generated (either adversarially or stochastically) by an exogenous process.
2. A model for targets/tasks: Tasks are usually modeled as points in a physical environment distributed according to some (possibly unknown) distribution, might require a certain level of on-site service time, and can be subject to a variety of constraints, for example, time windows and priorities.
3. A model for the UAVs and their motion: Besides their number, one needs to specify whether the UAVs are subject to algebraic (e.g., obstacles) or differential (e.g., minimum turning radius) constraints, sensing constraints, and fuel constraints. Also, UAVs might be able to communicate directly only with other UAVs (or static nodes) that lie within a certain radius, or might not have any communication capability (e.g., when cheap micro UAVs are used, or stealthiness is required). Finally, the control could be centralized (i.e., coordinated by a central station) or decentralized.
4. Performance criterion: Examples include the minimization of the waiting time before service, loss probabilities, and expectation-variance analysis.

Once the model is formulated, one seeks to characterize fundamental limitations of performance (in the form of lower bounds for the best achievable cost); the purpose of this step is essentially twofold: it allows to quantify the degree of optimality of

a routing algorithm and provides structural insights into the problem. As for the last step, the design of a routing algorithm usually relies on a careful combination of static routing methods with sequential re-optimization. Desirable properties for the static methods are as follows: (i) the static problem can be solved (at least approximately) in polynomial time, and (ii) the static method is amenable to a statistical characterization (this is essential for the computation of performance bounds). Formal performance guarantees on a routing algorithm are then obtained by quantifying the ratio between an upper bound on the cost delivered by that algorithm and a lower bound for the best achievable cost. Such a ratio, being an estimate of the degree of optimality of the algorithm, should be close to one and possibly independent of systems's parameters (i.e., a *constant factor* guarantee, as defined in Sect. 85.2).

In the remainder of this section, three problems will be considered, all involving only one UAV (the server in the queueing model: in the remainder of this chapter, the terms “server” and “UAV,” as well as “task” and “target,” will be used interchangeably). In the first problem, the UAV moves along continuous paths and visits targets whose time of arrival, location, and on-site service are stochastic; information about target location is communicated to the UAV upon target arrival. In the second problem, the motion of the UAV is subject to differential constraints, but still the UAV has full knowledge of newly arrived targets. Finally, in the third problem, the information available to the UAV is limited, that is, the UAV is not aware of a target's existence or location upon its arrival time but must first detect it using onboard sensors.

The purpose of these examples is twofold: on the one hand, to provide some concrete examples about how to apply spatial queuing theory to devise UAV routing algorithms in dynamic settings, on the other hand, to provide guidelines for UAV routing in a variety of scenarios of interest. The case of multi-UAV coordination will be addressed in the next section.

85.3.1 UAV Routing with No Motion Constraints and Unlimited Sensing

Consider a basic scenario where one UAV moves along continuous paths (with no differential constraints, e.g., on the curvature) and visits spatially localized targets whose time of arrival, location, and on-site service are stochastic; information about target location is communicated to each UAV upon target arrival. This problem has been studied in the literature as the dynamic traveling repairman problem (DTRP) in Bertsimas and van Ryzin (1991) and can be summarized as follows:

85.3.1.1 The Single-Server DTRP Problem

In a geographical region \mathcal{Q} of area A , a dynamic process generates spatially localized tasks. The process generating tasks is modeled as a spatiotemporal Poisson process, that is, (i) the time between each pair of consecutive events has an exponential distribution with intensity $\lambda > 0$, and (ii) upon arrival, the locations

of tasks are independently and uniformly distributed in \mathcal{Q} . The location of the new tasks is assumed to be immediately available to the UAV. The UAV provides service in \mathcal{Q} , flying at constant speed v ; the UAV is assumed to have unlimited fuel and task-servicing capabilities. Each task requires an independent and identically distributed amount of on-site service with finite mean duration $\bar{s} > 0$. A task is completed when the UAV moves to its location and performs its on-site service. The objective is to design a *routing policy* that maximizes the quality of service delivered by the UAV in terms of the average steady-state time delay \bar{T} between the generation of a task and the time it is completed (in general, in a dynamic setting, the focus is on the quality of service as perceived by the “end user,” rather than, e.g., fuel economies achieved by the UAV). Other quantities of interests are the average number \bar{N} of tasks waiting to be completed and the waiting time \bar{W} of a task before its location is reached by a UAV. These quantities, however, are related according to $\bar{T} = \bar{W} + \bar{s}$ (by definition) and by Little’s law, stating that $\bar{N} = \lambda \bar{W}$, for stable queues (Little 1961).

One comment is in order: the queueing models used to model UAV routing problems are inherently different from traditional, nonspatial queueing models. In particular, one might be tempted to consider the queueing model for the single-server DTRP as a standard $M/G/1$ queue (where M stands for Poisson arrival process, G indicates that the service times are identically and *independently* distributed according to a general distribution, and 1 is the number of servers). The main reason is that in UAV routing, the “service time” has both a *travel* and an *on-site* component. Although the on-site service requirements are “statistically” independent (by assumption), the travel times generally are not.

85.3.1.2 Stability

Before proceeding further, it is necessary to ensure the stability of the system. The system is considered stable if the expected number of waiting tasks is uniformly bounded at all times, or equivalently, that tasks are removed from the system at least at the same rate at which they are generated. In the case at hand, the time to complete a task is the sum of the time to reach its location (which depends on the routing policy) plus the time spent at that location in on-site service (which is independent of the routing policy). Since, by definition, the service time is no shorter than the on-site service time \bar{s} , then a weaker necessary condition for stability is $\varrho := \lambda \bar{s} < 1$; the quantity ϱ measures the fraction of time the UAV is performing on-site service. Remarkably, it turns out that this is also a sufficient condition for the single-server DTRP; note that this stability condition is independent of the size and shape of \mathcal{Q} and of the speed of the vehicle.

85.3.1.3 Lower Bounds

To derive lower bounds, the main difficulty usually consists in bounding (possibly in a statistical sense) the amount of time spent to reach a target location. The derivation of these bounds becomes simpler in asymptotic regimes, that is, looking at cases when $\varrho \rightarrow 0^+$ and $\varrho \rightarrow 1^-$, which are often called “light load” and “heavy load” conditions, respectively.

For example, consider first the case in which $\varrho \rightarrow 0^+$ (light load regime). The median of \mathcal{Q} is defined as the point that minimizes the expected distance to a random point sampled uniformly from \mathcal{Q} ; this distance can be written as $H_1^* \sqrt{A}$, where H_1^* is a constant that only depends on the shape of \mathcal{Q} (see also Appendix A.1). Assuming that the UAV would have enough time to return to the median location before the appearance of each new task (in other words, assuming light load conditions), the expected system time can be lower bounded as

$$\bar{T} \geq \frac{H_1^* \sqrt{A}}{v} + \bar{s} \quad (\text{as } \varrho \rightarrow 0^+).$$

Consider now the case in which $\varrho \rightarrow 1^-$ (heavy load). Let \bar{D} be the average travel distance per task for some routing policy. By using arguments from geometrical probability (independent of algorithms), one can show that $\bar{D} \geq \beta_2 \sqrt{A} / \sqrt{2N}$ as $\varrho \rightarrow 1^-$, where β_2 is a constant that will be specified later. As discussed, for stability, one needs $\bar{s} + \bar{D}/v < 1/\lambda$. Combining the stability condition with the bound on the average travel distance per task, one obtains

$$\bar{s} + \frac{\beta_2 \sqrt{A}}{v \sqrt{2N}} \leq \frac{1}{\lambda}.$$

Since, by Little's law, $\bar{N} = \lambda \bar{W}$ and $\bar{T} = \bar{W} + \bar{s}$, one finally obtains (recall that $\varrho = \lambda \bar{s}$)

$$\bar{T} \geq \frac{\beta_2^2 A}{2} \frac{\lambda}{v^2 (1 - \varrho)^2} + \bar{s}, \quad (\text{as } \varrho \rightarrow 1^-).$$

This lower bound allows to draw the following conclusions for the single-server DTRP: (i) the condition $\varrho < 1$ is also sufficient for stability, and (ii) the quality of service, which is proportional to $1/(1 - \varrho)^2$, degrades much faster as the target load increases than in a nonspatial queueing systems (where the growth rate is proportional to $1/(1 - \varrho)$).

85.3.1.4 Routing for the Single-Server DTRP

Consider the following routing policy for the single-server DTRP, based on a partition of \mathcal{Q} into $p \geq 1$ subregions $\{Q_1, Q_2, \dots, Q_p\}$ of equal area A/p . Such a partition can be obtained, for example, as sectors centered at the median of \mathcal{Q} . Define a cyclic ordering for the subregion, such that, for example, if the vehicle is in region Q_i , the “next” region is Q_j , where j follows i in the cyclic ordering (in other words, $j = (i + 1) \bmod p$).

1. If there are no outstanding targets, move to the median of the region \mathcal{Q} .
2. Otherwise, visit the “next” subregion; subregions with no tasks are skipped. Compute a minimum-length path from the UAV's current position through all the outstanding tasks in that subregion. Complete all tasks on this path, ignoring new tasks generated in the meantime.
3. Repeat.

In the above policy, two static optimization methods are applied, depending on whether or not there are outstanding tasks. Computing the median of \mathcal{Q} is a standard problem in geometric optimization (see Appendix A.1). The problem of computing the shortest path through a number of points is related to the well-known traveling salesman problem (TSP). While the TSP is a prototypically hard combinatorial problem, it is well known that the Euclidean version of the problem is very easy to approximate; see Appendix B.1 for more details. Furthermore, the length $\text{ETSP}(n)$ of a Euclidean TSP through n points independently and uniformly sampled in \mathcal{Q} is known to satisfy the following property:

$$\lim_{n \rightarrow \infty} \text{ETSP}(n)/\sqrt{n} = \beta_2 \cdot \sqrt{A}, \quad \text{almost surely,}$$

where $\beta_2 \approx 0.712$ is a constant; this is the same constant that appears in the lower bound for \bar{T} . The convergence to this limit is very fast: Larson and Odoni (1981) report that for “fairly compact and fairly convex” regions, the estimate $\text{ETSP}(n) \approx \beta_2 \sqrt{nA}$ is within a few percent from the true value for as few as 15 points.

It can be shown (see, e.g., Bertsimas and van Ryzin 1993a) that, using the above routing policy, the average system time \bar{T} satisfies

$$\bar{T} = \frac{H_1^* \sqrt{A}}{v} + \bar{s}, \quad (\text{as } \varrho \rightarrow 0^+),$$

$$\bar{T} \leq \gamma(p) \frac{A}{v^2} \frac{\lambda}{(1 - \varrho)^2} + \bar{s}, \quad (\text{as } \varrho \rightarrow 1^-),$$

where $\gamma(1) = \beta_2^2$ and $\gamma(p) \rightarrow \beta_2^2/2$ for large p . These results critically exploit the statistical characterization of the length of an optimal TSP tour. Hence, the proposed policy achieves a quality of service that is arbitrarily close to the optimal one, in the asymptotic regimes of light or heavy load.

While the simple policy stated above is provably optimal in light and heavy load, in the sense that no policy can provide a strictly lower system time, several variations have been proposed that improve the performance in other scenarios or operating conditions, at the expense of some additional complexity in implementation and analysis. The interested reader can find more information in Bullo et al. (2011) and references therein.

85.3.2 UAV Routing with Motion Constraints and Unlimited Sensing

In this section, the complexity of the single-server DTRP problem is increased, by imposing differential constraints on the trajectories that the UAV can follow.

85.3.2.1 The Single-Server Dubins DTRP Problem

In this version of the problem, the task generation process and performance metrics are assumed to be the same as in the standard DTRP problem. On the other hand, the UAV is modeled as a non-holonomic vehicle, constrained to move on the plane at

constant speed v , along paths of bounded curvature. In particular, the instantaneous radius of curvature is constrained to be no less than ρ . This model is often referred to as the Dubins vehicle, in recognition of Dubins' work in computing minimum-length paths for such model (Dubins 1957), and is typically considered appropriate to model the kinematics of UAVs (Chandler et al. 2000; Beard et al. 2002). The UAVs are assumed to be identical and have unlimited range. In the course of this chapter, the term *Dubins frame* shall be used to refer to a coordinate frame with the origin attached to the Dubins vehicle and its first axis aligned with the vehicle's velocity vector. For simplicity, the region \mathcal{Q} will be assumed to be a rectangle of height H and width W , with $H < W$, and $WH = A$. The on-site service time will be assumed to be identically zero, for example, as in flyby requests. This version of the DTRP will be henceforth called the Dubins DTRP.

85.3.2.2 Lower Bounds

The lower bounds from the basic case still hold. Note that in this case, since $\bar{s} = 0$, the load is entirely determined by λ : in the light-load case $\lambda \rightarrow 0^+$, and in the heavy-load case $\lambda \rightarrow +\infty$. So, for example, the light-load lower bound is

$$\bar{T} \geq H_1^*/v, \quad (\text{as } \lambda \rightarrow 0^+).$$

For the heavy-load case, it is possible to derive a lower bound specific to the Dubins DTRP. Let $\bar{D}_\rho(n)$ be the expected distance along a Dubins path, between a UAV situated in the interior of \mathcal{Q} and the closest among n points independently and uniformly sampled from \mathcal{Q} . Reachability arguments show that, for large n ,

$$\bar{D}_\rho(n) \geq \bar{\gamma}_D \left(\frac{\rho A}{n} \right)^{1/3},$$

where $\bar{\gamma}_D = \frac{3}{4} \sqrt[3]{3}$. In the case $\bar{s} = 0$, stability requires $\bar{D}_\rho(\bar{N})/v \leq 1/\lambda$, and Little's condition states that $\bar{N} = \lambda \bar{T}$. Hence, for large λ ,

$$\frac{1}{\lambda} \geq \frac{\bar{D}_\rho(\bar{N})}{v} \geq \bar{\gamma}_D \left(\frac{\rho A}{\lambda \bar{T}} \right)^{1/3} \frac{1}{v},$$

and finally

$$\bar{T} \geq \bar{\gamma}_D^3 \frac{\rho A}{v^3} \lambda^2, \quad (\text{as } \lambda \rightarrow +\infty).$$

85.3.2.3 Routing for the Single-Server Dubins DTRP Problem

Consider the following routing policy for the single-server Dubins DTRP problem. As in the case of the "standard" single-server DTRP, the policy is based on a partition of \mathcal{Q} . However, in this case, the partition is computed through a tiling of the plane into "beads" of length

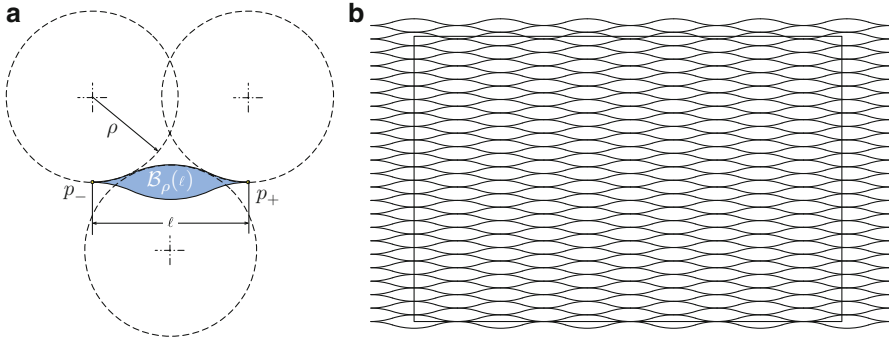


Fig. 85.2 (a) Construction of the “bead.” The figure shows how the upper half of the boundary is constructed and the bottom half is symmetric. (b) Bead tiling of the plane, with the beads aligned with the width of \mathcal{Q}

$$\ell = \min \left\{ \frac{7 - \sqrt{17}}{4} \left(1 + \frac{7\pi\rho H}{3A} \right)^{-1} \frac{v}{\lambda}, 4\rho \right\},$$

aligned along the width direction of \mathcal{Q} ; see Fig. 85.2. All beads with a nonempty intersection with \mathcal{Q} are ordered in a cyclic fashion, in such a way that beads sharing a cusp are adjacent in the ordering. The beads are constructed in such a way that a Dubins vehicle arriving at point p_- (refer Fig. 85.2) with heading toward point p_+ can service at least one target anywhere inside the bead and reach point p_+ with the same heading as it had when it arrived at point p_- . This feature allows the Dubins vehicle to service at least one target per bead in a cyclic fashion.

1. If there are no outstanding targets, loiter on a circular trajectory of radius ρ about the median of the region \mathcal{Q} . The direction of loitering is irrelevant and can be chosen in such a way that the loitering circle is returned to in minimum time after servicing the target.
2. Otherwise, visit beads in order, servicing at least one target per nonempty bead. Shortcuts skipping empty beads can be taken, as long as the cyclic ordering of the beads is preserved.
3. Repeat.

In the light-load case, in which most of the time there are no outstanding tasks, it is clear that the above policy attempts to replicate the policy used in the basic case, with no differential constraints. In the heavy-load case, the construction of the beads and their cyclic ordering ensure that (i) the number of beads in \mathcal{Q} is proportional to λ^3 , and hence the rate at which tasks are generated within a single bead is proportional to λ^{-2} , and (ii) the length of the cycle is proportional to λ^2 , hence the rate at which the UAV is able to complete tasks within a single bead (at least once per cycle), is proportional to λ^{-2} . Indeed, the proposed policy is able to stabilize the system and in fact provides a provable constant-factor approximation to the optimal system time.

Summarizing, it can be shown that, using the stated policy, the average system time \bar{T} satisfies

$$\bar{T} \leq \frac{H_1^* \sqrt{A} + 7/3\pi\rho}{v}, \quad (\text{as } \lambda \rightarrow 0^+),$$

and

$$\bar{T} \leq 71 \frac{\rho A}{v^3} \left(1 + \frac{7}{3}\pi \frac{\rho}{W}\right)^3 \lambda^2, \quad (\text{as } \lambda \rightarrow +\infty).$$

In other words, the performance of the stated policy is approximately optimal in light load if the minimum turning radius ρ is negligible with respect to \sqrt{A} . In the heavy-load case, the policy is provably stabilizing and is within a constant factor of the optimum.

85.3.3 UAV Routing with Limited Sensing Capabilities

In some applications, UAVs are not aware of new tasks upon their arrival; rather, the UAVs must search for and detect the tasks with limited-range onboard sensors before being able to complete them. This section is devoted to this problem, which can be called *persistent patrolling* problem.

85.3.3.1 The Persistent Patrolling Problem (PPP)

In this version of the problem, the task generation process and performance metrics are the same as in the DTRP problem, as well as the model for the motion of the UAVs, which can fly at constant speed along any continuous path, with no differential constraints imposed. However, the UAVs are not aware of the location of new tasks until they detect them using limited-range onboard sensor. For the sake of simplicity, the sensing region of a UAV is modeled as a disk of radius σ centered at the position of the UAV. Other shapes of the sensor footprint can be considered with minor modifications to specifics of the algorithms, without affecting their performance significantly. The analysis will focus on the case in which σ is small when compared to the region \mathcal{Q} (or, more precisely, to a characteristic length, e.g., \sqrt{A}). Also, in order to highlight some interesting aspects of the problem, the task generation process will be modeled with a nonuniform spatial distribution, described by a probability density function φ supported on \mathcal{Q} .

This problem is related to the well-studied problem of optimal search (Stone 1975). However, the fact that tasks are dynamically generated changes the problem in a fundamental way. In particular, new tasks may be generated in areas that have already been explored, hence requiring the UAV to return to previously visited locations. Problems of this nature have been studied, for example, in Song et al. (2010), considering uniform distribution of tasks; however, standard sweep methods do not work well if φ is not uniform. Motivated by search, patrolling, or foraging applications, other works such as (Mathew and Mezic 2009; Mesquita

2010; Cannata and Sgorbissa 2011) give algorithms that ensure that the distribution of the UAV (over time) asymptotically matches a given distribution, often taken to be the distribution of an underlying stochastic process. However, as discussed in the following, for the case at hand, the desired spatial distribution of the UAV's position is dependent on, but not equivalent to, the spatial distribution φ of tasks.

85.3.3.2 Lower Bounds

The problem will be discussed with a focus on the case in which $\sigma \rightarrow 0^+$, that is, the sensing range is very small, and hence search time is the main factor determining the system time. It can be shown that in this case

$$\bar{T} \geq \frac{1}{4v\sigma} \left(\int_{\mathcal{Q}} \sqrt{\varphi(q)} dq \right)^2 + \bar{s}.$$

Since $\left(\int_{\mathcal{Q}} \sqrt{\varphi(q)} dq \right)^2 \leq \int_{\mathcal{Q}} \varphi(q) dq = 1$ (e.g., by Jensen's inequality), any nonuniformity in the task distribution is beneficial in terms of detection time and should be exploited by the patrolling UAV.

85.3.3.3 Routing for the Single-Server PPP

In the case in which the density φ is uniform, the optimal patrolling strategy consists of following a “lawnmower pattern,” that is, a cyclic path that allows the vehicle's footprint to cover all of \mathcal{Q} . For small σ (i.e., ignoring boundary effects), and uniform φ , this achieves the lower bound on the system time and is hence optimal.

The case in which φ is not uniform, the design of a good patrolling strategy is more involved. As in previous cases, a routing policy can be designed based on an appropriate partition of the environment. For simplicity of exposition, assume that the distribution φ is piecewise constant, that is, there is a partition $\{\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_l\}$ of \mathcal{Q} such that $\varphi(q) = \varphi_i$ for all $q \in \mathcal{Q}_i$, $i = 1, \dots, l$. Further partition each subregion \mathcal{Q}_i into $p_i = \lceil \bar{p}/\sqrt{\varphi_i} \rceil$ tiles of equal area, $i = 1, \dots, l$, where \bar{p} is a number that is large enough that the values $\bar{p}/\sqrt{\varphi_i}$ are well approximated by integers. Define a cyclic ordering of the subregions \mathcal{Q}_i and cyclic orderings of the p_i tiles within each subregion, $i = 1, \dots, l$. Each of these ordering defines uniquely the “next” subregion (or tile), given the current subregion (or tile).

1. If any outstanding tasks have been revealed, use the single-server DTRP routing policy to complete these tasks.
2. Otherwise, move to the “next” tile in the “next” subregion, and sweep this tile using a lawnmower pattern. (The initial tile and subregion can be chosen arbitrarily.)
3. Repeat.

The idea in the above policy is to ensure that the time-averaged “density” of the UAV matches $\sqrt{\varphi}$. As $\sigma \rightarrow 0^+$, the system time using this patrolling strategy matches the lower bound and is hence optimal.

Interestingly, in the case in which σ is given, and $\varrho \rightarrow 1^-$, it turns out that the sensing-range limitation does not constrain the system's performance. In other words, in heavy-load conditions, the rate at which new tasks will be detected while servicing previously detected targets is high enough that sensing limitations do not make a difference, and the performance for the "standard" single-server DTRP is recovered.

85.4 Spatial Queueing Theory: The Multi-server Case

The previous section presented an algorithmic approach to spatial queueing theory for single-server problems; this section extends such an approach to the multi-server case. The focus is on *control decentralization*: in fact, a decentralized architecture can provide robustness to failures of single servers and can guarantee better time efficiency; also, it might reduce the total implementation and operation cost, increase reactivity and system reliability, and add flexibility and modularity with respect to the centralized counterpart.

The extension to the multi-server case coupled with the constraint of control decentralization adds significant challenges with respect to the single-server case. Fortunately, there are a number of cases where a simple, yet systematic approach allows to lift single-server routing policies to *decentralized* multi-server routing policies with provable performance guarantees. The idea is to have the servers partition the workspace into regions of dominance via a decentralized partitioning algorithm and then have each server follow a single-server policy within its own region. Specifically, one defines an *m-partition* of \mathcal{Q} as a collection of m closed subsets $\{Q_i\}_{i=1}^m$ with disjoint interiors and whose union is \mathcal{Q} . Given a single-server policy π and an *m-partition* of \mathcal{Q} , a *π -partitioning policy* is a multi-server policy such that (i) one server is assigned to each subregion (thus, there is a one-to-one correspondence between servers and subregions), and (ii) each server executes the single-server policy π to service demands that fall within its own subregion. Note that a partitioning policy is parametrized by the single-server policy π and by the *m-partition* of \mathcal{Q} , possibly computed with a decentralized partitioning algorithm. Which partitions should one consider, and to what extent this *decoupling* strategy affects optimality? These questions are discussed in detail in Pavone et al. (2009), where the authors illustrate a number of scenarios, partitioning schemes, and *decentralized* partitioning algorithms whereby one can retain optimality, or at least some degree of optimality, under this (systematic) decomposition.

In the following, paralleling the structure of the previous section, the multi-server dynamic routing problem will be studied for three problems: (1) the simplest case of teams of UAVs without motion constraints and with unlimited sensing, (2) the case of teams of UAVs with differential motion constraints and with unlimited sensing, and (3) the case of teams of UAVs without motion constraints but with

limited information about the environment. In all three cases, the aforementioned partitioning procedure will be pivotal for the design of provably efficient multi-UAV routing strategies.

85.4.1 Multi-UAV Routing with No Motion Constraints and Unlimited Sensing

The problem has the same definition of the DTRP problem, with the exception that m UAVs are available to provide service to the targets.

85.4.1.1 Lower Bounds

The lower bounds (and the techniques to derive them) are similar to the ones for the single-server case. Consider first the light-load case (i.e., $\varrho \rightarrow 0^+$). The m -median of \mathcal{Q} is defined as the set of m points that minimizes the expected distance between a random point sampled uniformly from \mathcal{Q} and the closest point in such set (in other words, the m -median of \mathcal{Q} is the global minimizer $P_m^* := \operatorname{argmin}_{(p_1, \dots, p_m) \in \mathcal{Q}^m} \mathbb{E}[\min_{k \in \{1, \dots, m\}} \|p_k - q\|]$). This distance can be written as $H_m^*(\mathcal{Q})\sqrt{A/m}$, where $H_m^*(\mathcal{Q})$ lies in the interval $[0.3761, c(\mathcal{Q})]$ where $c(\mathcal{Q})$ is a constant that depends only on the shape of \mathcal{Q} . The m -median of \mathcal{Q} induces a Voronoi partition that is called *median Voronoi tessellation*. Recall that the *Voronoi diagram* $\mathcal{V}(P_m^*) = (V_1, \dots, V_m)$ of \mathcal{Q} generated by points $P_m^* = (p_1, \dots, p_m)$ is defined by

$$V_i = \left\{ q \in \mathcal{Q} \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i, j \in \{1, \dots, m\} \right\}.$$

See Fig. 85.3 for an example of a Voronoi diagram. The expected system time in light load can then be lower bounded as

$$\bar{T} \geq \frac{H_m^*(\mathcal{Q})\sqrt{A}}{v\sqrt{m}} + \bar{s}, \quad (\text{as } \varrho \rightarrow 0^+).$$

Hence, in light load, the optimal system time scales with the square root of the number of UAVs. In the heavy-load case, one can show

$$\bar{T} \geq \frac{\beta_2^2 A}{2 v^2} \frac{\lambda}{m^2 (1 - \varrho/m)^2}, \quad (\text{as } \varrho/m \rightarrow 1^-).$$

A salient feature of the above lower bound is that it scales *quadratically* with the number of servers; note, however, that congestion effects are not included in this model.

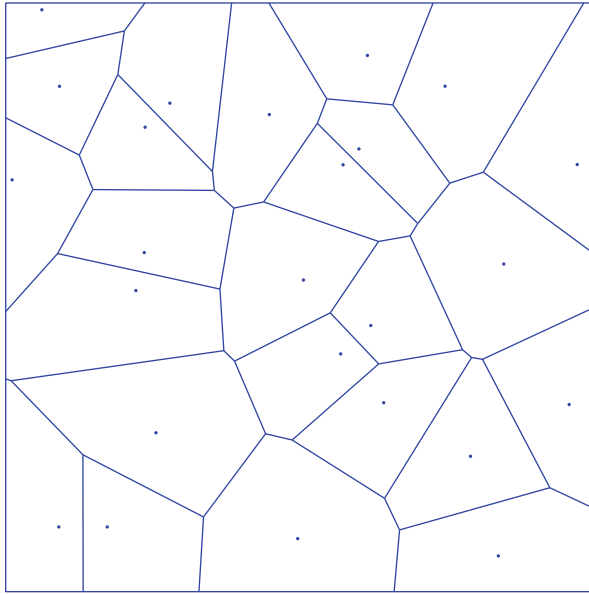


Fig. 85.3 A Voronoi diagram with 25 generators

85.4.1.2 Routing for the Multi-server DTRP

The design of a partitioning policy relies on three basic steps, that is,

1. Characterization of an optimal (or constant-factor optimal) single-server routing policy
2. Characterization of efficient partitioning schemes
3. Design of (possibly decentralized) algorithms for workspace partitioning

Consider Step 2. Let π be an optimal (or constant-factor optimal) single-server policy for the DTRP. Assuming heavy-load conditions (when the problem resembles one of workload balance), one can show that a “ π -partitioning policy” that uses m -partitions whose subregions have *equal area* (i.e., *equitable* m -partitions) has the same optimality properties as π . This remarkable result extends to the case where the distribution of targets is not uniform, targets have priorities and targets have time windows and even to the case of non-holonomic UAVs, as illustrated in the next section. This result, however, only holds under the heavy-load assumption; when the load is only moderate, the shape of subregions can have a significant effect. In moderate load, a solution that turns out to be effective for the DTRP is to adopt equitable partitioning policies in which the subregions are “fat,” that is, with a small diameter for a given area, rather than long and thin. Finally, in light-load conditions, when the problem resembles one of geometric optimization, the relevant partitions are the *median Voronoi tessellations* defined in the previous lower bound section.

This discussion leads to the following (centralized) routing policy for the multi-server DTRP:

1. Compute an m median of \mathcal{Q} , and the corresponding Voronoi partition.
2. Assign one vehicle to each Voronoi region,
3. Each UAV executes the single-server DTRP policy in its own subregion.

Using the above routing policy, the average system time \bar{T} satisfies

$$\bar{T} = \frac{H_m^*(\mathcal{Q})\sqrt{A}}{\sqrt{mv}} + \bar{s}, \quad (\text{as } \varrho \rightarrow 0^+),$$

$$\bar{T} \leq \gamma \frac{A}{v^2} \frac{\lambda}{m^2 (1 - \varrho/m)^2}, \quad (\text{as } m \rightarrow +\infty \text{ and } \varrho/m \rightarrow 1^-).$$

where γ is the optimality factor of the single-server routing policy.

Note that the heavy-load result above relies on the fact that for large m , each Voronoi region in a median Voronoi tessellation has the same area (see Appendix A.1). For general m , this may not be the case; furthermore, one might wonder whether equitable partitions can be computed in a decentralized fashion (this corresponds to the third step in the aforementioned policy design procedure).

In the solution proposed in Pavone et al. (2011), power diagrams are the key geometric concept to obtain in a decentralized fashion, *equitable*, and *median Voronoi* partitions (or “good” approximations when they do not exist). Define

$$P_W := \left((p_1, w_1), \dots, (p_m, w_m) \right) \in (\mathcal{Q} \times \mathbb{R})^m.$$

The pair (p_i, w_i) is called a *power point*. The *power diagram* $\mathcal{V}(P_W) = (V_1, \dots, V_m)$ of \mathcal{Q} generated by power points P_W is defined by

$$V_i(P_W) = \left\{ q \in \mathcal{Q} \mid \|q - p_i\|^2 - w_i \leq \|q - p_j\|^2 - w_j, \forall j \neq i, j \in \{1, \dots, m\} \right\}.$$

The set P_W is the set of *power generators* of $\mathcal{V}(P_W)$, and V_i is the power cell of the i -th power generator. When all weights are the same, the power diagram coincides with the Voronoi diagram. One can show that given $m \geq 1$ distinct points (p_1, \dots, p_m) in \mathcal{Q} , there exist weights $w_i, i \in \{1, \dots, m\}$, such that the set of power points $\left((p_1, w_1), \dots, (p_m, w_m) \right)$ generates a power diagram that is equitable, that is, where all power regions have the same area. The basic idea, then, is to associate to each UAV i a *virtual* power generator (virtual generator for short) (p_i, w_i) ; then, the power cell V_i becomes the region of dominance for UAV i (see Fig. 85.4). A virtual generator (p_i, w_i) is simply an artificial (or logical) variable whose value is locally controlled by the i th UAV. In general, the position of an UAV and the position of its virtual generator are *distinct*, that is, the position of an UAV inside its own region of dominance V_i is independent from the position of its virtual generator (see Fig. 85.4). Note that an equitable power diagram can be obtained by just changing the values of the weights (while keeping the generators’s positions fixed). Thus, the degrees of freedom given by the positions

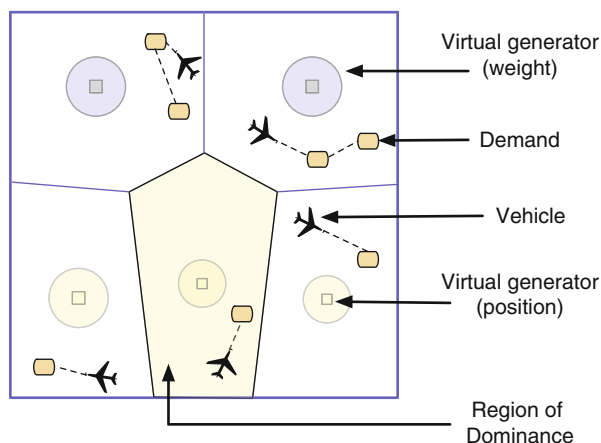


Fig. 85.4 Vehicles, virtual generators, tasks/demands, and regions of dominance. A positive weight w is represented by a yellow circle with radius \sqrt{w} ; a negative weight w is represented by a blue circle with radius $\sqrt{|w|}$. Note that the position of a UAV and the position of its virtual generator are, in general, distinct

of the generators can be used to “steer” the partition toward an equitable and *median* Voronoi partition. Specifically, the idea is to construct an energy function with the properties that (1) it depends on the weights and positions of the virtual generators, and (2) all its critical points correspond to vectors of weights and positions yielding an equitable power diagram. Then, each UAV updates its own virtual generator by updating the weight according to a *decentralized* gradient-descent law (with respect to the energy function) and by updating the virtual generator’s position so as to steer the partition toward a median Voronoi diagram, when this motion does not increase the disagreement between the areas of the neighboring regions (see Pavone et al. 2011 for details). Under this decentralized partitioning algorithm, an equitable partition is always achieved; hence the resulting partitioning policy is optimal in heavy load. Furthermore, if an m -median of \mathcal{Q} that induces a Voronoi partition that is equitable exists, this partitioning algorithm will locally converge to it; thus the resulting decentralized partitioning policy is locally optimal in light load.

85.4.2 Multi-UAV Routing with Motion Constraints and Unlimited Sensing

The definition of the problem is the same as the one for the single-server Dubins DTRP problem, but now there are m UAVs providing service. In the following, for brevity, this problem is referred to as the m -Dubins DTRP.

85.4.2.1 Lower Bounds

In contrast to the standard DTRP, the lower bound for the m -Dubins DTRP does not follow easily from the results for the single-server Dubins DTRP. Consider first the light-load case. In this case, the lower bound is a function of a non-dimensional parameter called *non-holonomic vehicle density*:

$$d_\rho := \frac{\rho^2 m}{A},$$

which is proportional to the ratio of the area of a disk whose radius is the turning radius and the area of \mathcal{Q} available per vehicle. Hence, d_ρ is representative of the significance of the turning radius with respect to the typical distance that an individual vehicle travels. One can give the following lower bounds for the m -Dubins DTRP:

$$\begin{aligned} \bar{T} &\geq \frac{H_m^*(\mathcal{Q})}{v} \left(\frac{A}{m} \right)^{1/2}, & (\text{as } \lambda \rightarrow 0^+ \text{ and } d_\rho \rightarrow 0^+), \\ \bar{T} &\geq \frac{3\sqrt[3]{3}}{4v} \left(\frac{\rho A}{m} \right)^{1/3}, & (\text{as } \lambda \rightarrow 0^+ \text{ and } d_\rho \rightarrow +\infty). \end{aligned}$$

The first bound is obtained by approximating the Dubins distance (i.e., the length of the shortest feasible path for a Dubins vehicle) with the Euclidean distance. The second lower bound is obtained by explicitly taking into account the Dubins turning cost. Note that when the motion constraint becomes “active” as $d_\rho \rightarrow +\infty$, the lower bound scales with $m^{-1/3}$.

Reachability arguments show that when λ is large (i.e., in heavy load), a lower bound for the heavy-load case is as follows:

$$\bar{T} \geq \frac{81}{64} \frac{\rho A}{v^3} \frac{\lambda^2}{m^3}, \quad (\text{as } \lambda/m \rightarrow +\infty).$$

Hence, for the m -Dubins DTRP, the lower bound scales with the inverse of the *cube* of the number of UAVs (in contrast to the standard DTRP where the scaling is inverse of the square); note, again, that the model does not include congestion effects.

85.4.2.2 Routing Policies for the m -Dubins DTRP

Consider, first, the case when λ is small (i.e., light load) and the non-holonomic density is low; in this case, the problem resembles the standard multi-server DTRP problem. Accordingly, the relevant partition scheme is a median Voronoi tessellation, and the following partitioning policy is efficient in this case.

1. Compute a m -median of \mathcal{Q} ,
2. Assign one UAV to each Voronoi region,

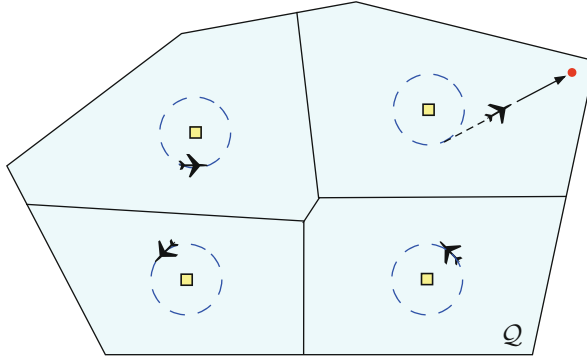


Fig. 85.5 Illustration of the light-load policy for low values of non-holonomic density. The squares represent elements of $P_m^*(Q)$, the m -median of Q . Each vehicle loiters about its respective generator at a radius ρ . The regions of dominance are the Voronoi partition generated by $P_m^*(Q)$. In this figure, a task has appeared in the subregion roughly in the upper-right quarter of the domain. The vehicle responsible for this subregion has left its loitering orbit and is en route to service the demand

- Each UAV visits the demands in the Voronoi region in the order in which they arrive. When no demands are available, the UAV returns to the median and loiters on a circular trajectory of radius ρ centered at the median of the subregion.

This policy is illustrated in Fig. 85.5. The performance of this policy is given by

$$\bar{T} \leq \frac{H_m^*(Q)}{v} \left(\frac{A}{m} \right)^{1/2}, \quad (\text{as } \lambda \rightarrow 0^+ \text{ and } d_\rho \rightarrow 0^+),$$

that is, this policy is optimal. Hence, in this asymptotic regime, the system time scales as $1/(v\sqrt{m})$.

In light load, when the non-holonomic density is large, one should instead consider *dynamic* partitions; in particular, an effective policy is as follows:

- Bound the environment Q with a rectangle of minimum *height*, where height denotes the smaller of the two side lengths of a rectangle. Let W and H be the *width* and *height* of this bounding rectangle, respectively. Divide Q into strips of width w , where

$$w = \min \left\{ \left(\frac{4}{3\sqrt{\rho}} \frac{WH + 10.38\rho H}{m} \right)^{2/3}, 2\rho \right\}.$$

- Orient the strips along the side of length W .
- Construct a closed Dubins path which runs along the longitudinal bisector of each strip, visiting all strips in top-to-bottom sequence, making U-turns between

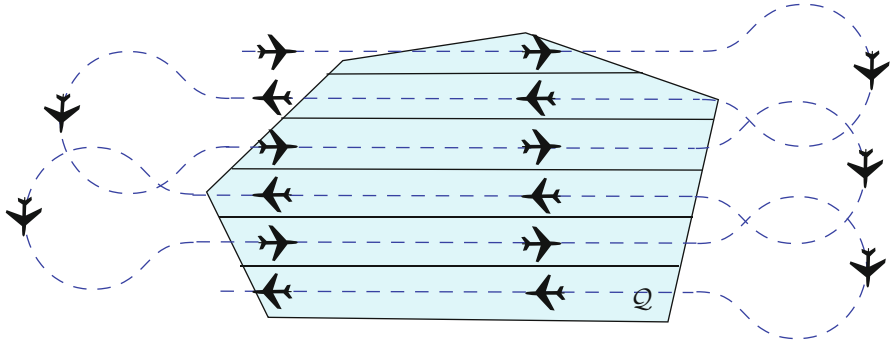


Fig. 85.6 Illustration of the light-load policy for large non-holonomic densities. The segment providing closure of the loitering path (returning the UAVs from the end of the last strip to the beginning of the first strip) is not shown here for clarity of the drawing

strips at the edges of Q , and finally returning to the initial configuration. The m UAVs loiter on this path, equally spaced, in terms of path length.

A depiction of this policy is shown in Fig. 85.6. At the instant a target arrives, one constructs a circle of radius ρ which is tangent to the loitering path and intersects the target. The UAV responsible for visiting the target is the one closest in terms of loitering path length to the point of departure, at the time of target arrival. After a UAV has serviced a target, it must return to its place in the loitering pattern.

Note that the dynamic partition associated with a particular UAV is in fact fixed in the reference frame of that UAV and that in the global frame, these partitions could be regarded as a *dynamic version of an equitable partition* of Q , modulo the boundary effects. The reason for using dynamic equitable partitions is as follows. For large values of non-holonomic density, the area per vehicle is small in comparison to the disk of radius ρ , and hence the efficient regions of responsibility for the UAVs in such a regime coincide with their *small-time* reachable sets. Since the UAVs modeled as a Dubins vehicle cannot stall, a simple way to ensure that the regions of responsibility are contained inside the small-time reachable sets all the time is to have the regions of responsibility fixed in the reference frame of the UAVs.

The system time of the above policy that uses dynamic partitions satisfies

$$\bar{T} \leq \begin{cases} \frac{1.238}{v} \left(\frac{\rho WH + 10.38 \rho^2 H}{m} \right)^{1/3} + \frac{W + H + 6.19\rho}{mv} & \text{for } m \geq 0.471 \left(\frac{WH}{\rho^2} + \frac{10.38H}{\rho} \right), \\ \frac{WH + 10.38\rho H}{4\rho mv} + \frac{W + H + 6.19\rho}{mv} + \frac{1.06\rho}{v} & \text{otherwise.} \end{cases}$$

Hence this policy is within a constant factor of the optimal in the asymptotic regime where $\lambda/m \rightarrow 0^+$ and $d_\rho \rightarrow +\infty$. Moreover, in such asymptotic regime, the optimal system time scales as $1/(v\sqrt[3]{m})$.

Finally, for the heavy load, the relevant partitions are *additively weighted equitable* partitions, as follows:

1. Divide the environment into regions of dominance with lines parallel to the bead rows. Let the area and height of the i th vehicle's region be denoted with A_i and H_i . Place the subregion dividers in such a way that

$$A_i + \frac{7}{3}\pi\rho H_i = \frac{1}{m} \left(A + \frac{7}{3}\pi\rho H \right), \quad \text{for all } i \in \{1, \dots, m\}.$$

2. Allocate one subregion to every vehicle.
3. Each vehicle executes the single vehicle policy in its own region, where the size of bead in region i is chosen to be $\ell_i = \min\{C_{BT,i}v/\lambda_i, 4\rho\}$, with $\lambda_i = \lambda A_i/A$ being the arrival rate in region i , and $C_{BT,i} = \frac{7-\sqrt{17}}{4} \left(1 + \frac{7\pi\rho H_i}{3A_i}\right)^{-1}$.

The system time for this policy satisfies

$$\bar{T} \leq 71 \frac{\rho A}{v^3} \left(1 + \frac{7\pi\rho H}{3A}\right)^3 \frac{\lambda^2}{m^3}, \quad (\text{as } \lambda/m \rightarrow +\infty).$$

Hence, this policy is within a constant factor of the optimal in heavy load and the optimal system time in this case scales as $\lambda^2/(mv)^3$.

Note that all the above three policies are partitioning policies; the partitions used are, however, quite different, ranging from median Voronoi tessellations, to dynamic partitions, to additively weighted equitable partitions. Median Voronoi tessellations and additively weighted equitable partitions can be computed in a *decentralized* fashion with the partitioning algorithm discussed for the multi-DTRP problem (indeed, a median Voronoi tessellation can be computed even without interagent communication). No decentralized algorithms have been developed so far for dynamic partitions.

It is instructive to compare the scaling of the optimal system time with respect to λ , m , and v for the m -DTRP and for the m -Dubins DTRP. Such comparison is shown in Table 85.1. One can observe that in heavy load, the optimal system time for the m -Dubins DTRP is of the order $\lambda^2/(mv)^3$, whereas for the m -DTRP, it is of the order $\lambda/(mv)^2$. This analysis rigorously establishes the following intuitive fact: bounded-curvature constraints make the optimal system much more sensitive to increases in the demand generation rate. Perhaps less intuitive is the fact that the optimal system time is also more sensitive with respect to the number of vehicles and the vehicle speed in the m -Dubins DTRP as compared to the m -DTRP. In the light load, the optimal system time for the m -DTRP is of the order $1/(v\sqrt{m})$, which is the same for the m -DTRP but only when the non-holonomic density is small. When the non-holonomic density is high, the optimal system time is of the order $1/(v\sqrt[3]{m})$, that is, it is more sensitive to the number of vehicles than in the low non-holonomic density case. This suggests the existence of a critical value of d_ρ below which the partitioning policy using median Voronoi partitions is efficient and

Table 85.1 A comparison between the scaling of the optimal system time for the multi-server DTRP and for the m -Dubins DTRP

| | $\bar{T}^*, m\text{-DTRP}$ | $\bar{T}^*, m\text{-Dubins DTRP}$ |
|--|--|--|
| Heavy load ($\lambda/m \rightarrow +\infty$) | $\Theta\left(\frac{\lambda}{m^2 v^2}\right)$ | $\Theta\left(\frac{\lambda^2}{m^3 v^3}\right)$ |
| Light load ($\lambda/m \rightarrow 0^+$) | $\Theta\left(\frac{1}{v\sqrt{m}}\right)$ | $\Theta\left(\frac{1}{v\sqrt{m}}\right)$ if $d_\rho \rightarrow 0^+$ |
| | | $\Theta\left(\frac{1}{v^3\sqrt{m}}\right)$ if $d_\rho \rightarrow +\infty$ |

above which the dynamic partitioning policy is efficient. The details of such phase transitions can be found in Enright et al. (2009).

85.4.3 Multi-UAV Routing with Limited Sensing Capabilities

The definition of the problem is the same as in the single-server persistent patrolling problem, with the only difference being the number of UAVs providing service.

85.4.3.1 Lower Bounds

As in the single-server case, the discussion will focus on the case in which $\sigma \rightarrow 0^+$. In this case, the system time is bounded by

$$\bar{T} \geq \frac{1}{4mv\sigma} \left(\int_{\mathcal{Q}} \sqrt{\varphi(q)} dq \right)^2 + \bar{s}.$$

85.4.3.2 Routing Policies for the m -Server PPP

Optimal static partitioning policies can be designed using partitions that are simultaneously equitable with respect to φ and $\sqrt{\varphi}$. (Such partitions can always be found.) However, a simpler strategy is based on a “dynamic” partition, as follows:

1. If any outstanding tasks have been revealed, use the multi-server DTRP routing policy to complete these tasks.
2. Otherwise, move to the “next” tile in the “next” subregion, and sweep this tile using a lawnmower pattern. (The initial tile and subregion for each vehicle are chosen in such a way that vehicles are uniformly spaced along the subregion/tile cycle.)
3. Repeat.

The above policy matches the lower bound and is hence optimal as $\sigma \rightarrow 0^+$. In the case in which σ is finite, and $\varrho \rightarrow 1^-$, the sensing-range limitation does not

constrain the system's performance, and the performance for the "standard" multi-server DTRP is recovered. As discussed above, no decentralized algorithms have been developed so far for dynamic partitions.

85.5 Conclusions

In this chapter, we presented a dynamic vehicle routing framework for coordination of UAVs performing spatially distributed tasks in uncertain and dynamic environments. The technical approach relies on using algorithmic spatial queueing theory for designing efficient algorithms for single vehicles, and then augmenting it with spatial partitioning policies to extend it to multiple vehicles. To illustrate this approach, a variety of problems were considered, ranging from the simplest case with no motion or sensing constraints to the cases with motion differential constraints and with limited sensing constraints. Additionally, multi-UAV versions for each of these cases were considered. For each scenario, UAV routing policies have been provided, which are provably optimal or approximately optimal in the asymptotic regimes of light and heavy load. The dependence of the performance of these algorithms on mission parameters such as the number, speed, sensor footprint and turning radius of the UAVs, and the dimension of the workspace was discussed. A system designer could interpolate between such scaling laws for the extreme cases as established in this chapter to choose a policy that best fits given problem specifications. Moreover, the algorithms and their underlying principles presented in this chapter for canonical scenarios could also provide guidelines to design algorithms for other scenarios not explicitly considered in this chapter.

Appendix A

A.1 The Continuous Multi-median Problem

Given a set $\mathcal{Q} \subset \mathbb{R}^d$ and a vector $P = (p_1, \dots, p_m)$ of m distinct points in \mathcal{Q} , the expected distance between a random point q , generated according to a probability density function φ , and the closest point in P is given by

$$\begin{aligned} H_m(P, \mathcal{Q}) &:= \mathbb{E} \left[\min_{i \in \{1, \dots, m\}} \|p_i - q\| \right] \\ &= \sum_{i=1}^m \int_{\mathcal{V}_i(P, \mathcal{Q})} \|p_i - q\| \varphi(q) dq, \end{aligned}$$

where $\mathcal{V}(P, \mathcal{Q}) = (\mathcal{V}_1(P, \mathcal{Q}), \dots, \mathcal{V}_m(P, \mathcal{Q}))$ is the Voronoi partition of the set \mathcal{Q} generated by the points P . In other words, $q \in \mathcal{V}_i(P, \mathcal{Q})$ if $\|q - p_i\| \leq \|q - p_k\|$, for all $k \in \{1, \dots, m\}$. The set \mathcal{V}_i is referred to as the Voronoi cell of the generator p_i .

The function H_m is known in the locational optimization literature as the continuous Weber function or the continuous multi-median function; see (Agarwal and Sharir 1998; Drezner 1995) and references therein.

The m -median of the set \mathcal{Q} , with respect to the measure induced by φ , is the global minimizer

$$P_m^*(\mathcal{Q}) = \operatorname{argmin}_{P \in \mathcal{Q}^m} H_m(P, \mathcal{Q}).$$

Let $H_m^*(\mathcal{Q}) = H_m(P_m^*(\mathcal{Q}), \mathcal{Q})$ be the global minimum of H_m . It is straightforward to show that the map $P \mapsto H_1(P, \mathcal{Q})$ is differentiable and strictly convex on \mathcal{Q} . Therefore, it is a simple computational task to compute $P_1^*(\mathcal{Q})$. It is convenient to refer to $P_1^*(\mathcal{Q})$ as the median of \mathcal{Q} . On the other hand, the map $P \mapsto H_m(P, \mathcal{Q})$ is differentiable (whenever (p_1, \dots, p_m) are distinct) but not convex, thus making the solution of the continuous m -median problem hard in the general case. It is known (Agarwal and Sharir 1998; Megiddo and Supowit 1984) that the discrete version of the m -median problem is NP-hard for $d \geq 2$. Gradient algorithms for the continuous m -median problems can be designed by means of the equality

$$\frac{\partial H_m(P, \mathcal{Q})}{\partial p_i} = \int_{\mathcal{V}_i(P, \mathcal{Q})} \frac{p_i - q}{\|p_i - q\|} \varphi(q) dq.$$

The set of critical points of H_m contains all configurations (p_1, \dots, p_m) with the property that each p_i is the generator of the Voronoi cell $\mathcal{V}_i(P, \mathcal{Q})$ as well as the median of $\mathcal{V}_i(P, \mathcal{Q})$. We refer to such Voronoi diagrams as *median Voronoi diagrams*. It is possible to show that a median Voronoi diagram always exists for any bounded convex domain \mathcal{Q} and density φ .

The dependence of $H_m^*(\mathcal{Q})$ on m plays a crucial role in the design and analysis of algorithms relying on geometric optimization. However, finding the exact relationship for the general case is difficult; hence, it is of great interest to provide bounds on $H_m^*(\mathcal{Q})$. This problem is studied thoroughly in Papadimitriou (1981) for square regions and in Zemel (1984) for more general compact regions. It is known that, in the asymptotic case ($m \rightarrow +\infty$), $H_m^*(\mathcal{Q}) = c_{\text{hex}} \sqrt{A/m}$ almost surely, where $c_{\text{hex}} \approx 0.377$ is the first moment of a hexagon of unit area about its center. This optimal asymptotic value is achieved by placing the m points at the centers of the hexagons in a regular hexagonal lattice within \mathcal{Q} (the honeycomb heuristic). Working towards the above result, it is also known that for any $m \in \mathbb{N}$,

$$\frac{2}{3} \sqrt{\frac{A}{\pi m}} \leq H_m^*(\mathcal{Q}) \leq c(\mathcal{Q}) \sqrt{\frac{A}{m}},$$

where $c(\mathcal{Q})$ is a constant depending on the shape of \mathcal{Q} .

Appendix B

B.1 The Euclidean Traveling Salesman Problem

The Euclidean TSP is formulated as follows: given a finite set D of n points in \mathbb{R}^d , find the minimum-length closed curve through all points in D . In graph theoretical language, a tour of the point set D is a spanning cycle of the complete graph with vertex set D ; the length of a tour is the sum of all Euclidean distances between points in the tour.

The asymptotic behavior of stochastic TSP problems for large n exhibits the following interesting property. Let $\text{ETSP}(n)$ be a random variable returning the length of the Euclidean TSP tour through n points, independently and uniformly sampled from a compact set \mathcal{Q} of unit area; in Beardwood et al. (1959), it is shown that there exists a constant β_2 such that, almost surely,

$$\lim_{n \rightarrow +\infty} \frac{\text{ETSP}(n)}{\sqrt{n}} = \beta_2.$$

In other words, the optimal cost of stochastic TSP tours approaches a deterministic limit and grows as the square root of the number of points to be visited; the current best estimate of the constant appearing in the limit is $\beta_2 = 0.7120 \pm 0.0002$; see Percus and Martin (1996) and Johnson et al. (1996). Similar results hold in higher dimensions, and for nonuniform point distributions from Steele (1990), the above limit takes the general form

$$\lim_{n \rightarrow +\infty} \frac{\text{ETSP}(n)}{n^{1-1/d}} = \beta_d \int_{\mathcal{Q}} \bar{\varphi}(q)^{1-1/d} dq \quad \text{almost surely,}$$

where $\bar{\varphi}$ is the density of the absolutely continuous part of the distribution φ from which the n points are independently sampled. Notice that the bound holds for all compact sets: the shape of the set only affects the convergence rate to the limit. According to Larson and Odoni (1981), if \mathcal{Q} is a “fairly compact and fairly convex” set in the plane, the estimate $\text{ETSP}(n) \approx \beta_2 \sqrt{n}$ for values of n as low as 15. Remarkably, the asymptotic cost of the stochastic TSP for uniform point distributions is an upper bound on the asymptotic cost for general point distributions, that is,

$$\lim_{n \rightarrow +\infty} \frac{\text{ETSP}(n)}{n^{1-1/d}} \leq \beta_d.$$

This follows directly from an application of Jensen’s inequality, that is,

$$\int_{\mathcal{Q}} \bar{\varphi}(q)^{1-\frac{1}{d}} dq \leq \left(\int_{\mathcal{Q}} \bar{\varphi}(q) dq \right)^{1-\frac{1}{d}} \leq \varphi(\mathcal{Q})^{1-\frac{1}{d}} = 1.$$

The TSP is known to be NP-hard, which suggests that there is no general algorithm capable of finding the optimum tour in an amount of time polynomial in the size of the input. Even though the exact optimal solutions of a large TSP can be very hard to compute, several exact and heuristic algorithms and software tools are available for the numerical solution of Euclidean TSPs.

The most advanced TSP solver to date is arguably *concorde* (Applegate et al. 1998). Heuristic polynomial-time algorithms are available for constant-factor approximations of TSP solutions, such as Christofides' algorithm, providing a $3/2$ approximation factor (Christofides 1972). On a more theoretical side, Arora (1997) proved the existence of polynomial-time approximation schemes, providing a $(1 + \varepsilon)$ constant-factor approximation for any $\varepsilon > 0$.

A modified version of the Lin-Kernighan heuristic (Lin and Kernighan 1973) is implemented in *linkern*; this powerful solver yields approximations in the order of 5 % of the optimal tour cost very quickly for many instances. For example, in numerical experiments on a 2.4 GHz Pentium machine, approximations of random TSPs with 1,000 points typically required about 2 s of CPU time. Both *concorde* and *linkern* are written in ANSI C and, at the time of writing, are freely available for academic research use at <http://www.tsp.gatech.edu/concorde/index.html>.

In this chapter, several routing policies were presented requiring on-line solutions of large TSPs. Practical implementations of the algorithms will rely on heuristics, such as Lin-Kernighan's or Christofides'. If a constant-factor approximation algorithm is used, the effect on the asymptotic performance guarantees of our algorithms can be simply modeled as a scaling of the constant β_d .

References

- P.K. Agarwal, M. Sharir, Efficient algorithms for geometric optimization. *ACM Comput. Surv.* **30**(4), 412–458 (1998)
- M. Alighanbari, J.P. How, A robust approach to the UAV task assignment problem. *Int. J. Robust Nonlinear Control* **18**(2), 118–134 (2008)
- D. Applegate, R. Bixby, V. Chvátal, W. Cook, On the solution of traveling salesman problems, in *Proceedings of the International Congress of Mathematicians, Extra Volume ICM III*, Berlin, 1998, pp. 645–656. *Documenta Mathematica, Journal der Deutschen Mathematiker-Vereinigung*
- S. Arora, Nearly linear time approximation scheme for Euclidean TSP and other geometric problems, in *Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science*, Miami Beach, 1997, pp. 554–563
- G. Arslan, J.R. Marden, J.S. Shamma, Autonomous vehicle-target assignment: a game theoretic formulation. *ASME J. Dyn. Syst. Meas. Control* **129**(5), 584–596 (2007)
- R.W. Beard, T.W. McLain, M.A. Goodrich, E.P. Anderson, Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans. Robot. Autom.* **18**(6), 911–922, 2002
- J. Beardwood, J. Halton, J. Hammersley, The shortest path through many points. *Proc. Camb. Philos. Soc. Math. Phys. Sci.* **55**(4), 299–327 (1959)
- D.J. Bertsimas, G.J. van Ryzin, A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Oper. Res.* **39**, 601–615 (1991)
- D.J. Bertsimas, G.J. van Ryzin, Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles. *Oper. Res.* **41**(1), 60–76 (1993a)

- D.J. Bertsimas, G.J. van Ryzin, Stochastic and dynamic vehicle routing with general interarrival and service time distributions. *Adv. Appl. Probab.* **25**, 947–978 (1993b)
- F. Bullo, E. Frazzoli, M. Pavone, K. Savla, S.L. Smith, Dynamic vehicle routing for robotic systems. *Proc. IEEE* **99**(9), 1482–1504 (2011)
- G. Cannata, A. Sgorbissa, A minimalist algorithm for multirobot continuous coverage. *IEEE Trans. Robot.* **27**(2), 297–312 (2011)
- P. Chandler, S. Rasmussen, M. Pachter, UAV cooperative path planning, in *AIAA Conference on Guidance, Navigation, and Control*, Denver (2000)
- N. Christofides, Bounds for the travelling-salesman problem. *Oper. Res.* **20**, 1044–1056 (1972)
- Z. Drezner (ed.), *Facility Location: A Survey of Applications and Methods*. Series in Operations Research (Springer, New York, 1995). ISBN:0-387-94545-8
- L.E. Dubins, On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**, 497–516 (1957)
- J.J. Enright, K. Savla, E. Frazzoli, F. Bullo, Stochastic and dynamic routing problems for multiple UAVs. *AIAA J. Guid. Control Dyn.* **32**(4), 1152–1166 (2009)
- B. Golden, S. Raghavan, E. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*. Operations Research/Computer Science Interfaces, vol. 43 (Springer, New York/London, 2008). ISBN:0387777776
- S. Irani, X. Lu, A. Regan, On-line algorithms for the dynamic traveling repair problem. *J. Sched.* **7**(3), 243–258 (2004)
- P. Jaillet, M.R. Wagner, Online routing problems: value of advanced information and improved competitive ratios. *Transp. Sci.* **40**(2), 200–210 (2006)
- D.S. Johnson, L.A. McGeoch, E.E. Rothberg, Asymptotic experimental analysis for the held-karp traveling salesman bound, in *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, Atlanta, 1996, pp. 341–350
- S.O. Krumke, W.E. de Paepe, D. Poensgen, L. Stougie, News from the online traveling repairman. *Theor. Comput. Sci.* **295**(1–3), 279–294 (2003)
- R. Larson, A. Odoni, *Urban Operations Research* (Prentice Hall, Englewood Cliffs, 1981)
- S. Lin, B.W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem. *Oper. Res.* **21**, 498–516 (1973)
- J.D.C. Little, A proof for the queuing formula: $L = \lambda W$. *Oper. Res.* **9**(3), 383–387 (1961). ISSN:0030364X. <http://www.jstor.org/stable/167570>
- G. Mathew, I. Mezić, Spectral multiscale coverage: a uniform coverage algorithm for mobile sensor networks, in *Proceedings of the 48th IEEE Control and Decision Conference*, Shanghai, 2009, pp. 7872–7877
- N. Megiddo, K.J. Supowit, On the complexity of some common geometric location problems. *SIAM J. Comput.* **13**(1), 182–196 (1984). ISSN:0097–5397
- A.R. Mesquita, *Exploiting Stochasticity in Multi-agent Systems*. PhD thesis, University of California at Santa Barbara, Santa Barbara, 2010
- B.J. Moore, K.M. Passino, Distributed task assignment for mobile agents. *IEEE Trans. Autom. Control* **52**(4), 749–753 (2007)
- C.H. Papadimitriou, Worst-case and probabilistic analysis of a geometric location problem. *SIAM J. Comput.* **10**(3), 542–557 (1981)
- M. Pavone, Dynamic vehicle routing for Robotic networks. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2010
- M. Pavone, K. Savla, E. Frazzoli, Sharing the load. *IEEE Robot. Autom. Mag.* **16**(2), 52–61 (2009)
- M. Pavone, A. Arsie, E. Frazzoli, F. Bullo, Distributed algorithms for environment partitioning in mobile robotic networks. *IEEE Trans. Autom. Control* **56**(8), 1834–1848 (2011)
- G. Percus, O.C. Martin, Finite size and dimensional dependence of the Euclidean traveling salesman problem. *Phys. Rev. Lett.* **76**(8), 1188–1191 (1996)
- H.N. Psaraftis, Dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transp. Sci.* **14**(2), 130–154 (1980)
- D.D. Sleator, R.E. Tarjan, Amortized efficiency of list update and paging rules. *Commun. ACM* **28**(2), 202–208 (1985)

- S.L. Smith, F. Bullo, Monotonic target assignment for robotic networks. *IEEE Trans. Autom. Control* **54**(9), 2042–2057 (2009)
- D. Song, C.Y. Kim, J. Yi, Stochastic modeling of the expected time to search for an intermittent signal source under a limited sensing range, in *Proceedings of Robotics: Science and Systems*, Zaragoza, 2010
- J.M. Steele, Probabilistic and worst case analyses of classical problems of combinatorial optimization in Euclidean space. *Math. Oper. Res.* **15**(4), 749 (1990)
- L.D. Stone, *Theory of Optimal Search* (Academic, New York, 1975)
- P. Toth, D. Vigo (eds.), *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications (SIAM, Philadelphia, 2001). ISBN:0898715792
- P. Van Hentenryck, R. Bent, E. Upfal, Online stochastic optimization under time constraints. *Ann. Oper. Res.* **177**(1), 151–183 (2009)
- E. Zemel, Probabilistic analysis of geometric location problems. *Ann. Oper. Res.* **1**(3), 215–238 (1984)

Section XVI

UAV Integration into the National Airspace

*Konstantinos Dalamagkidis and
Richard S. Stansbury*

Kimon P. Valavanis and George J. Vachtsevanos

UAV Integration into the National Airspace reviews and presents policies, procedures, and requirements for manned aviation and relates them to existing and under development equivalent policies and requirements for unmanned aviation. Integration of unmanned aviation into the (any) national airspace requires, again, that “UAVs function as if there were a human pilot onboard.” This requirement and restriction dictates that equivalent levels of safety (ELOS) must be derived for unmanned aviation (depending on the UAV classification), in addition to reliability and fault tolerance requirements. The major deviation from manned aviation requirements is that a controlled crash may be allowed for UAVs provided that human lives are not endangered and that collateral damage is minimized.

► [Aviation Regulation](#) by Dalamagkidis presents a brief introduction to aviation regulation. Key terms are defined and an overview of current manned aviation regulations is provided using the U.S. Federal Aviation Regulation as an example. This includes airworthiness certification, operation rules, and airspace classes.

► [Human Factors of Unmanned Aircraft System Integration in the National Airspace System](#) by Kaliardos and Lyall focuses on identifying human factor challenges to integrating UAS in the National Airspace System (NAS) for both pilots and air traffic controllers. The method for identifying these challenges is primarily based on the differences or “gaps” between manned aircraft in the NAS today and the unique aspects of UAS. The goal is not to generate a comprehensive

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

list of human factor issues, but to focus on those that are traceable to fundamental characteristics of UAS and that are also considered challenging with respect to the current NAS and its regulatory framework. It is assumed that for integration into the NAS, future UAS will be much more standardized in many respects than in the past – closer to that of today’s civil manned aircraft, which are subject to rigorous design, operational, manufacturing, and training approvals by the FAA. Under such assumptions, human factors are not only central to the very definition of UAS but are also central to some of the most important challenges facing the integration of UAS in the NAS.

► **Methodologies for Regulatory Compliance and Harmonization** by Marshall proposes a methodology for analysis of current regulations for applicability to UAS activities. Where the rules are in development, or merely being contemplated, a structure for harmonization with international standards is proposed. The focus is on the USA as the nation that has the most comprehensive aviation regulations, while also supporting major efforts to promulgate regulations and standards to supplement those regulations with UAS specific requirements and criteria. To provide global context, the International Civil Aviation Organization’s (ICAO’s) structure and procedures are reviewed in detail and contrasted with state-sponsored regulatory processes, with an analysis of the applicability of ICAO’s standards to UAS operations.

► **A Certification Strategy for Small Unmanned Aircraft Performing Nomadic Missions in the U.S. National Airspace System** by Stachura, Elston, Argrow, Frew, and Dixon discusses the specifics of the Certificates of Authorization (COA) obtained for the second Verification of the Origin of Rotation in Tornadoes Experiment (VORTEX2) project and how the operations are conducted to satisfy the COA requirements. A strategy is outlined for operating these nomadic missions with small UAS within the confines of FAA regulations. This includes information on getting FAA COAs for a large area, specifically focusing on area selection, airworthiness, and emergency procedures, which are the keys to these applications.

► **Hazard and Safety Risk Modeling** by Dalamagkidis presents aspects of risk modeling with a focus on UAS. It provides an overview of the current level of safety of manned aviation in terms of accident statistics. These are then mapped as target levels for UAS under the “Equivalent Level of Safety” principle to provide a glimpse at what that may entail for UAS regulations. Different methodologies are presented for estimating the risk of ground impact and midair collision accidents and how these estimates can be translated to system requirements. Guidelines are provided on the use of different risk models followed by applying a selection of them to five different UAS in two distinct scenarios, to compare the results of different choices.

► **Safety Risk Management of Unmanned Aircraft Systems** by Clothier and Walker provides existing risk practitioners with a high-level introduction to some of the unique issues and challenges in the application of the safety risk management process to UAS. The scope is limited to safety risks associated with the operation of unmanned aircraft in the civil airspace system and over inhabited areas. This chapter notes the unique aspects associated with the application of the safety risk management process to UAS compared to that of conventionally piloted aircraft.

Key challenges discussed include the specification of high-level safety criteria; the identification, analysis, and evaluation of the risks; and the effectiveness of available technical and operational mitigation strategies. Some solutions to these challenges are examined including those currently in practice and those still under research and development.

► **Certification of Small UAS** by Leijgraaf describes certification procedures for (small) UAS. It focuses on the certification process, the requirements for the safe design of a UAS, and the organizational requirements for the company designing the UAS.

► **Technology Surveys and Regulatory Gap Analyses of UAS Subsystems Toward Access to the NAS** by Stansbury and Wilson provides the necessary details on how to conduct a technology survey and regulatory gap analysis of UAS technology subsystems. Four past studies performed by Embry-Riddle Aeronautical University for the FAA's William J. Hughes Technology Center are discussed. These studies address UAS propulsion systems, sense and avoid technologies and procedures, command control and communication, and emergency recovery and flight termination systems. A recommended process for future studies is provided.

► **Concept of Operations of Small Unmanned Aerial Systems: Basis for Airworthiness Towards Personal Remote Sensing** by Stark, Coopmans, and Chen discusses the challenges of UAS integration into the NAS for a class of small UAS for personal remote sensing (PRS). This approach is centered on the three-pillared foundations for NAS integration presented by the U.S. DoD Unmanned Aircraft System Airspace Integration Plan and focuses on the specific challenges that are unique to the PRS UAS platform. This chapter presents a concept of operations for these unmanned aircraft along with an application scenario example and concludes with a discussion of the future use of UAS in the NAS and how it is achievable through adequate regulations and standards.

► **Standards and Certification of a UAS Sense and Avoid Capability** by Zeitlin centers on "certifiable" and "standardized" UAV sense and avoid (SAA) capabilities that are needed to mitigate for the lack of a remote pilot's ability to "see and avoid" other aircraft, which is an operational requirement. Regulators certify the airworthiness of aircraft, assuring their compliance with applicable regulations. Approvals of SAA implementations constitute part of that process. Each applicant needs to develop a Project Specific Certification Plan in close coordination with the regulator. Various tools for safety analysis and configuration control support this effort. Specification of the certification basis for SAA needs close attention, since the system addresses functions traditionally allocated to an onboard pilot. Algorithms and software will receive scrutiny due to their safety roles. Standards are developed to capture common requirements in support of systems that may use some degree of unique design innovation. A standard should suggest means of demonstrating compliance to its requirements, and so doing will greatly simplify the burden for both applicant and regulator. The closest existing standard to SAA is that for the Traffic Alert and Collision Avoidance System II (TCAS II), but SAA includes the additional function of self-separation. SAA can be implemented with various technologies in a variety of architectures. SAA should provide two

basic functions, each performing prescribed subfunctions. Considerable modeling and simulation will be required to validate system performance and support the safety case. The regulator may offer guidance to applicants through publication of an Advisory Circular.

Collectively, this section provides nothing but a “glimpse” of challenges and issues that need to be addressed and solved before unmanned aircraft fly in unison with manned aircraft. Regardless, the information provided in this section is, to some degree, complete and comprehensive allowing for the reader to understand airspace regulation, certification procedures, airworthiness, UAV safety, and reliability.

Konstantinos Dalamagkidis

Contents

| | | |
|--------|-------------------------------------|------|
| 87.1 | Introduction | 2118 |
| 87.2 | Important Definitions | 2121 |
| 87.3 | Airworthiness Certification | 2122 |
| 87.3.1 | Type Certificate | 2123 |
| 87.3.2 | Standard Certificates | 2123 |
| 87.3.3 | Special Certificates | 2124 |
| 87.4 | Special Aircraft Categories | 2125 |
| 87.4.1 | Vehicles | 2126 |
| 87.4.2 | R/C Models | 2126 |
| 87.5 | Pilot Certification | 2126 |
| 87.6 | Operation Rules | 2127 |
| 87.6.1 | Flight Rules | 2128 |
| 87.6.2 | Emergency Rules | 2129 |
| 87.6.3 | Maintenance Requirements | 2130 |
| 87.7 | Airspace Classes | 2130 |
| 87.8 | Regulation Development Models | 2132 |
| | References | 2133 |

Abstract

This chapter presents a brief introduction to aviation regulation. Key terms are defined and an overview of current manned aviation regulations is provided using the U.S. Federal Aviation Regulation as an example. This includes airworthiness certification, operation rules, and airspace classes.

K. Dalamagkidis

Institut für Informatik I6, Technische Universität München, Garching bei München, Germany

e-mail: dalamagkidis@tum.de

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_109,

© Springer Science+Business Media Dordrecht 2015

2117

87.1 Introduction

The need to regulate civil aviation ensuring safety and healthy competition dates back to the 1920s, with several relevant conventions addressing such issues and concerns. The most significant such convention took place in Chicago in 1944, right after the end of the Second World War with more than 50 States attending. The accomplishments of that conference set the groundwork for aviation safety and international cooperation on regulations, standards, and procedures development, all relevant even to this day. Attending States also founded the International Civil Aviation Organization (ICAO) as a means to secure progress accomplished during the conference, as well as to facilitate future cooperation.

Although UAV operations were very limited before the 1944 Chicago Convention, Article 8 refers specifically to pilotless aircraft, and provisions within still apply to current systems. Some of those provisions are that a UAV cannot fly over another State without special authorization by that State (Article 8), UAVs are required to bear registration marks (Article 20), and they must have a certificate of airworthiness (Article 31). It should be noted that the Chicago Convention applies to civil aircraft, and as a result, aircraft used in military or law enforcement services may have different restrictions.

In the United States, aviation regulations are collected and codified in the Code of Federal Regulations (CFR), Title 14, Chap. I, also known as Federal Aviation Regulation (FAR). Similarly, in Europe the Joint Aviation Authorities (JAA) has issued the Joint Aviation Requirements (JAR), while other countries/regions may have other similar regulatory documents. Due to an ongoing effort for harmonization between the aviation regulations, part and section numbers between the JAR and the Title 14 CFR are largely the same.

This chapter presents an overview of key parts of current manned aviation regulations as defined in the Title 14 CFR, with the understanding that the provisions of other aviation regulations of other countries will be similar, if not the same. It should be noted that these regulations are currently applicable to manned aircraft but are more than likely to influence UAV regulations. Where appropriate, the relative section in the Title 14 CFR will be given. The aim of this chapter is to familiarize the reader with the general regulatory framework and with terms and concepts that are used throughout the handbook. For the latest and most accurate information, one is advised to consult with the civil aviation authority of his/her country and the current version of the respective regulations. For example, the Title 14 CFR is publicly available both online and in print from the Government Printing Office.

The main focus of aviation regulation is to ensure that the aviation industry operates in a safe manner. For example, the U.S. Federal Law gives the Secretary of Transportation and the Administrator of the Federal Aviation Administration (FAA) the authority to *conduct investigations, prescribe regulations, standards, and procedures, and issue orders* [49 USC §40113(a)]. The paragraph on safety considerations in public interest [49 USC §40101(d)] reads:

...the Administrator shall consider the following matters, among others, as being in the public interest:

1. assigning, maintaining, and enhancing safety and security as the highest priorities in air commerce.
2. regulating air commerce in a way that best promotes safety and fulfills national defense requirements.
3. encouraging and developing civil aeronautics, including new aviation technology.
4. controlling the use of the navigable airspace and regulating civil and military operations in that airspace in the interest of the safety and efficiency of both of those operations.
5. consolidating research and development for air navigation facilities and the installation and operation of those facilities.
6. developing and operating a common system of air traffic control and navigation for military and civil aircraft.
7. providing assistance to law enforcement agencies in the enforcement of laws related to regulation of controlled substances, to the extent consistent with aviation safety.

The statutory mandate of the FAA also includes regarding safety:

...before authorizing new air transportation services, evaluating the safety implications of those services; and preventing deterioration in established safety procedures, recognizing the clear intent, encouragement, and dedication of Congress to further the highest degree of safety in air transportation and air commerce, and to maintain the safety vigilance that has evolved in air transportation and air commerce and has come to be expected by the traveling and shipping public. [49 USC §40101(a)]

The Title 14 CFR comprises of several subchapters as follows:

- A Definitions (Parts 1–3)
- B Procedural rules (Parts 11–17)
- C Aircraft (Parts 21–50). Includes airworthiness certification (21–39), maintenance (43), as well as aircraft registration and marking (45–49)
- D Airmen (Parts 60–67)
- E Airspace (Parts 71–77)
- F Air Traffic and General Operating Rules (Parts 91–106) which includes operating rules (91–99) and special classes of vehicles (101–105)
- G Air Carriers and Operators for Compensation or Hire: Certification and Operations (Parts 110–139)
- H Schools and Other Certified Agencies (Parts 140–147)
- I Airports (Parts 150–169)
- J Navigational Facilities (Parts 170–171)
- K Administrative Regulations (Parts 183–193)
- N War Risk Insurance (Parts 198–199)

The provisions of the Title 14 CFR notwithstanding, the FAA issues supplementary nonregulatory material like handbooks, orders, Advisory Circulars (AC), and Technical Standard Orders (TSO) that clearly define appropriate procedures, standards, and practices that may be followed to ensure compliance with current regulations. This material helps ensure that aircraft manufacturers and operators are able to establish the minimum level of safety and reliability required for civil operations ASTM International (2004). Nevertheless, the following sections will draw material only from the CFR since the focus of this chapter is on regulatory requirements.

With respect to the ACs, of particular interest are AC 00-2 and AC 00-44. The first provides a checklist of all ACs, as well as the status of other FAA publications, while the second contains a list of any changes made by the FAA to the CFR. The FAA also publishes the Aeronautical Information Manual (AIM), a regularly updated document, mainly aimed at pilots. The AIM contains the basic flight information and procedures when flying in the U.S. National Airspace System (NAS). It consists of the following chapters:

1. Air Navigation
2. Aeronautical Lighting and Other Airport Visual Aids
3. Airspace
4. Air Traffic Control
5. Air Traffic Procedures
6. Emergency Procedures
7. Safety of Flight
8. Medical Facts for Pilots
9. Aeronautical Charts and Related Publications
10. Helicopter Operations

The AIM is supplemented by the International Flight Information Manual (IFIM) which provides information for planning international flights. Other handbooks published by the FAA that might be of interest include the “Airplane Flying Handbook,” the “Helicopter Flying Handbook,” and the “Pilot’s Handbook of Aeronautical Knowledge,” all aimed at pilots, as well as the “Aviation Maintenance Technician Handbook” and the “Aviation Instructor’s Handbook” for technicians and instructors, respectively. These and a number of other documents are available for download from the FAA website free of charge.

Several of the aforementioned documents adopt established standards prepared by government agencies like the U.S. Department of Defense, standards development organizations, as well as other organizations, national or international. A non-exhaustive list of organizations that have been involved with the development of aerospace-related standards is provided below:

- Aeronautical Radio, Incorporated (ARINC)
- American Institute of Aeronautics and Astronautics (AIAA)
- American National Standards Institute (ANSI)
- American Society of Testing & Materials (ASTM)
- American Welding Society (AWS)
- Electronic Industries Alliance (EIA)
- Electrostatic Discharge Association (ESDA)
- European Organization for Civil Aviation Equipment (EUROCAE)
- Institute of Electrical and Electronics Engineers (IEEE)
- Institute of Environmental Sciences and Technology (IEST)
- International Civil Aviation Organization (ICAO)
- International Electrotechnical Commission (IEC)
- International Organization for Standardization (ISO)
- National Aeronautics and Space Administration (NASA)
- National Institute of Standards and Technology (NIST)

- The North Atlantic Treaty Organization Standards Agency (NSA)
- Occupational Safety and Health Administration (OSHA)
- Radio Technical Commission for Aeronautics (RTCA)
- Society of Automotive Engineers (SAE)

87.2 Important Definitions

In this section, some definitions will be provided that are important from a regulatory viewpoint. The first concerns the definition of the term *aircraft*, which according to the ICAO is “... *any machine that can derive support in the atmosphere from the reactions of the air other than the reactions of the air against the earth’s surface*” [ICAO Annex 1, Annex 6 Part I]. The FAA defines an aircraft as “... *a device that is used or intended to be used for flight in the air*” [Title 14 CFR §1.1]. As such, among many other things, UAVs also fall within that definition.

An aircraft can be classified as *public* or *civil*. A public aircraft is defined by the FAA as either “... *an aircraft used only for the United States Government, ... [or] owned and operated by the government of a State, the District of Columbia, or a territory or possession of the United States or a political subdivision of one of these governments*” or “... *an aircraft owned or operated by the armed forces or chartered to provide transportation to the armed forces. ...*,” provided some conditions are met. Civil aircraft are all aircraft that are not public [Title 14 CFR §1.1]. To *operate* an aircraft means “*use, cause to use or authorize to use aircraft, for the purpose of air navigation including the piloting of aircraft, with or without the right of legal control (as owner, lessee, or otherwise)*” [Title 14 CFR §1.1].

Aircraft often operate within *controlled airspace*, that is, “*an airspace of defined dimensions within which air traffic control service is provided to IFR flights and to VFR flights in accordance with the airspace classification*” [Title 14 CFR §1.1]. The *air traffic control service (ATS)* is a “*a service operated by appropriate authority to promote the safe, orderly, and expeditious flow of air traffic*” [Title 14 CFR §1.1]. Another relevant term is *air traffic management (ATM)* which refers to “*the dynamic, integrated management of air traffic and airspace – safely, economically and efficiently – through the provision of facilities and seamless services in collaboration with all involved parties*” (Air Transport Association (ATA) 2008).

The terms *accident*, *incident*, *damage*, and *hazard* are of particular importance when specifying safety requirements. One can find varying definitions for the term *accident*. One of the most succinct used by the FAA is “*An unplanned event or series of events that results in damages.*” (Federal Aviation Administration 2000). The ICAO defines the same term as “*An occurrence associated with the operation of an aircraft which takes place between the time any person boards the aircraft with the intention of flight until such time as all such persons have disembarked, in which (a) a person is fatally or seriously injured as a result of: being in the aircraft; or direct contact with any part of the aircraft, including parts which have become detached from the aircraft; or direct exposure to jet blast (except when the injuries are from natural causes, self inflicted or inflicted by other persons, or when*

the injuries are to stowaways hiding outside the areas normally available to the passengers or crew); or (b) the aircraft sustains damage or structural failure which: adversely affects the structural strength, performance or flight characteristics of the aircraft and would normally require major repair or replacement of the affected component (except for engine failure or damage, when the damage is limited to the engine, its cowlings or accessories; or for damage limited to propellers, wing tips, antennas, tires, brakes, fairings, small dents or puncture holes in the aircraft skin); or (c) the aircraft is missing or is completely inaccessible” [ICAO Annex 13]. On the other hand, incident is defined as “An occurrence, other than an accident, associated with the operation of an aircraft which affects or could affect the safety of operation” [ICAO Annex 13].

The definition of accidents makes use of the term *damage*, which in turn is defined as “... *the severity of injury, and/or the physical, and/or functional, and/or monetary loss that could result if hazard control is less than adequate*” (Federal Aviation Administration 2000).

The term *hazard* refers to conditions that may lead to accidents, regardless of their probability or the severity of the outcome. The FAA differentiates between primary hazards and contributory hazards: “A *primary hazard is one that can directly and immediately results in: loss, consequence, adverse outcome, damage, fatality, system loss, degradation, loss of function, injury, etc. The primary hazard is also referred to as: catastrophe, catastrophic event, critical event, marginal event, and negligible event*” (Federal Aviation Administration 2000). Initiating hazards include events and conditions that start an adverse chain of events that can lead to an accident. Primary hazards are events that directly and immediately cause an accident. Contributory hazards are any kind of conditions that can lead to an accident, including primary hazards.

87.3 Airworthiness Certification

In order for any aircraft to fly legally in the United States, it must carry an airworthiness certificate issued by the FAA [Title 14 CFR §91.203]. Airworthiness certification covers a wide spectrum of areas related to aspects of the aircraft design, construction, and operation. Presented below are some of these areas along with a selection of the various aspects investigated during certification:

Flight: Performance, flight characteristics, controllability, maneuverability, and stability

Structure: Loads, control surfaces, stabilizing and balancing surfaces, and fatigue evaluation

Design and construction: Wings, control surfaces, control systems, landing gear, and pressurization

Powerplant: Fuel system, oil system, cooling system, induction system, exhaust, and control

Equipment: Instruments' installation, electrical systems, lights, and safety equipment

In addition to aircraft, Airworthiness Directives (AD) exist for aircraft engines and propellers.

According to FAA Order 8130.2F, there are two conditions that need be met in order for an aircraft to be considered airworthy; it must conform to its type certificate including any supplemental certificates, and it must be in a condition that ensures safe operation Federal Aviation Administration (2004). For aircraft that are not type certified, compliance with the second condition is adequate. Besides standard certification, Special Airworthiness Certificates (SAC) are also available, usually for experimental or special purpose aircraft.

It should be noted that the Title 14 CFR allows the FAA administrator to prescribe additional requirements and special conditions for aircraft, aircraft engines, or propellers when due to a novel or unusual feature, current airworthiness regulations are inadequate or inappropriate [Title 14 CFR §21.16].

87.3.1 Type Certificate

A type certificate is a collection of documents, drawings, specifications, datasheets and any related information needed to demonstrate compliance with the applicable paragraphs of the Title 14 CFR [Title 14 CFR §21.41]. These may also include inspection and preventive maintenance programs and instructions for continued airworthiness [Title 14 CFR §21.31]. During the application for type certificate, the FAA administrator may require an inspection and test of the aircraft [Title 14 CFR §21.33], which may also include flight tests [Title 14 CFR §21.35].

Once a type certificate has been issued, it is in effect until surrendered, suspended, or revoked [Title 14 CFR §21.51]. Nevertheless, after modifications to an aircraft, a new certificate may be required. When the extend of the changes is not significant, the type certificate can be amended [Title 14 CFR §21.91] or a supplemental certificate will be issued [Title 14 CFR §21.113].

87.3.2 Standard Certificates

Standard airworthiness certificates are given to aircraft that are type certified in any of the categories defined in [Title 14 CFR §21.175], including:

- Normal, utility, acrobatic, and commuter aircraft (Title 14 CFR Part 23)
- Transport aircraft (Title 14 CFR Part 25)
- Normal rotorcraft (Title 14 CFR Part 27)
- Transport rotorcraft (Title 14 CFR Part 29)
- Manned free balloons (Title 14 CFR Part 31)

In addition to the above categories, type certification is available for primary [Title 14 CFR §21.24], restricted [Title 14 CFR §21.25], U.S. Army surplus [Title 14 CFR §21.27] and imported [Title 14 CFR §21.29] aircraft, as well. An overview of the applicability requirements for each of the aforementioned categories is given in Table 87.1.

Table 87.1 Aircraft types with standard airworthiness certificates along with occupancy, weight, and other restrictions (compiled from information in the Title 14 CFR)

| Category | Max. seats | MTOW (kg) | Notes |
|-----------------------------------|------------------|-----------|---|
| Normal | ≤9 ^a | ≤5,670 | Non-acrobatic operations |
| Utility | ≤9 ^a | ≤5,670 | Limited acrobatic operations |
| Acrobatic | ≤9 ^a | ≤5,670 | No restrictions |
| Commuter | ≤19 ^a | ≤8,600 | Non-acrobatic operations |
| Transport | N/A | N/A | |
| Primary | ≤4 ^b | ≤1,225 | Limited power/unpressurized cabin |
| Restricted | N/A | N/A | Special purpose operations ^c |
| Normal rotorcraft | ≤9 ^a | ≤3,175 | |
| Transport rotorcraft ^d | ≤9 ^a | ≤9,070 | |
| Manned free balloons | N/A | N/A | |

^aExcluding pilot seats^bIncludes the pilot^cIncludes agricultural, forest and wildlife conservation, aerial surveying, patrolling, weather control, and aerial advertising operations^dTransport rotorcraft are type certificated in two categories (A and B). Rotorcraft that meet the above restriction may be certificated in the B category, while those with higher seating capacity must be certificated in the A category

87.3.3 Special Certificates

In FAA Order 8130.2F and for aircraft that do not meet requirements for a standard certificate but are still capable of safe flight, SAC are available. More specifically special certificates can be given in the primary [Title 14 CFR §21.184], restricted [Title 14 CFR §21.185] and limited [Title 14 CFR §21.189] categories, for aircraft type certificated under these categories. In addition to that, SACs are available for aircraft belonging to the light-sport category and for experimental aircraft. Finally special flight permits are also available.

87.3.3.1 Light-Sport

FAA Order 8130.2F also provides rules for certification of light-sport Aircraft (LSA). This category is for aircraft other than helicopters that do not exceed 600–650 kg, having a maximum speed of not more than 120 knots and a capacity of not more than two persons. Additional requirements are made based on the presence of certain equipment.

A SAC is issued for aircraft of this type after successful inspection of the aircraft and its documentation. The latter includes operating instructions and maintenance procedures and a statement from the manufacturer that the aircraft complies with the provisions of the appropriate consensus standards [Title 14 CFR §21.190]. Upon successful completion of the inspection, the FAA may amend the certificate with operational restrictions, if deemed necessary.

87.3.3.2 Experimental

Experimental certificates are given for a variety of purposes [Title 14 CFR §21.191]:

- Research and development of equipment, operating techniques, or new aircraft designs.
- Showing aircraft compliance with a type certificate or a supplemental certificate after major changes.
- Crew training.
- Exhibitions at air shows or movies. This includes required pilot training and flight from and to the exhibition area.
- Air racing, including practicing and flight from/to the area.
- Market surveys, sales training and customer flight crew training.
- Operating of amateur-built aircraft.
- Operating of primary kit-built aircraft that have not been assembled under the supervision and control of a production certificated entity.
- Operation of certain types of light-sport aircraft.

Before a special certificate in this category is issued, the applicant must submit appropriate documentation. In the case of aircraft used for research and development purposes, this documentation includes the purpose of the experiment along with the number of flights, the location, and drawings/photographs of the aircraft [Title 14 CFR §21.193]. FAA Order 8130.2F makes provisions for several operational restrictions for experimental aircraft depending on their characteristics. The duration of experimental certificates is 1 year or less except for kit-built aircraft that typically do not expire [Title 14 CFR 21.181].

87.3.3.3 Special Flight Permits

These permits are given to aircraft that would not qualify for other airworthiness certificates but are capable of safe flight [Title 14 CFR §21.197]. The purpose of these permits is to allow the aircraft to fly to a different location for storage, repairs, and maintenance or to avoid areas of impending danger. The permit is issued after an application where the purpose and characteristics of the flight are detailed, and it may include limitations or special instructions from the FAA [Title 14 CFR §21.199].

Special flight permits may also be given for airworthy aircraft, to allow them to fly with excess fuel weight, beyond their certificated capacity or when flying over areas where refueling is not possible [Title 14 CFR §21.197].

87.4 Special Aircraft Categories

Although normally all aircraft need either a standard or a SAC to fly, there is a category of aircraft (classified as vehicles in the Title 14 CFR) for which this requirement is waived. The other special category concerns remote-controlled (R/C) model aircraft that also operates under few restrictions. Although not mentioned in the

Title 14 CFR, R/C aircraft are of interest since they present the basis of many UAS designs. It should be stressed however that R/C models are allowed to operate only for recreational purposes and that in the Federal Aviation Administration (2007) it was made clear that UAS operations cannot be based on R/C model procedures.

87.4.1 Vehicles

This category of aircraft includes moored balloons, unmanned balloons, unmanned rockets defined in Title 14 CFR Part 101, and ultralights defined in Title 14 CFR Part 103. Ultralights are single-occupant, manned aircraft used for recreation or sport purposes only, with a maximum empty weight of 70 kg for unpowered and 115 kg for powered vehicles [Title 14 CFR §103.1].

Many of the requirements regarding pilot certification, operating and flight rules, vehicle registration and marking, and maintenance certification, including the requirement to carry an airworthiness certificate that are normally applicable to aircraft, are waived for this category. Nevertheless, operational restrictions may be in place.

For example, the following pertain to the operation of ultralight vehicles:

- Daylight operations only [Title 14 CFR §103.11].
- Yield the right-of-way to all aircraft [Title 14 CFR §103.13].
- No operations allowed over congested areas [Title 14 CFR §103.15].
- No operations allowed in Class A, B, C, and D airspace. For operations in Class E near airports, ATC authorization is required first [Title 14 CFR §103.17].
- Pilot must operate by visual reference with the surface [Title 14 CFR §103.21].

87.4.2 R/C Models

Model airplanes are regulated on a voluntary basis, based on AC91-57 with few operational restrictions. In addition to that an independent organization, the Academy of Model Aeronautics (AMA) issues normal or restricted flight permits after inspection of the model, provides insurance for its members, and organizes areas to safely practice aeromodeling. It is noteworthy that in Academy of Model Aeronautics (2007), restrictions additional to the ones in FAA AC91-57 are imposed, both in design (e.g., the weight of the models and their propulsion methods) as well as in operation.

87.5 Pilot Certification

Title 14 CFR Part 61 is involved with the requirements for issuing pilot, flight instructor, and ground instructor certificates, ratings, and authorizations [Title 14 CFR §61.1]. An appropriate pilot certificate is required for a person to assume the role of pilot in command or of required crew member [Title 14 CFR §61.3]. Some

Table 87.2 Pilot certificates summarized from [Title 14 CFR §61.5]

| Category | Class |
|-------------------------------|------------------------------------|
| Airplane | Single-engine land |
| | Multiengine land |
| | Single-engine sea |
| | Multiengine sea |
| Rotorcraft | Helicopter |
| | Gyroplane |
| Lighter-than-air | Airship |
| | Balloon |
| Weight-shift-control aircraft | Weight-shift-control aircraft land |
| | Weight-shift-control aircraft sea |
| Powered lift | N/A |
| Powered parachute | Powered parachute land |
| | Powered parachute sea |

operators are also required to possess a current medical certificate issued based on procedures described in Title 14 CFR Part 67.

There are several types of pilot certificates with different training and certification requirements and with different privileges for their holders [Title 14 CFR §61.5]:

- 1. Student pilot
- 2. Sport pilot
- 3. Recreational pilot
- 4. Private pilot
- 5. Commercial pilot
- 6. Airline transport pilot certificate

Each pilot certificate (with the exception of a student certificate) comes with ratings for aircraft categories, classes, and types the holder may operate as well as the instrument rating for private and commercial pilots. Table 87.2 summarizes the aircraft category and class ratings. There are also instrument ratings for airplanes, helicopters, and powered lifts [Title 14 CFR §61.5].

Similar ratings are placed on flight instructor and ground instructor certificates when all the training and certification requirements are met.

Title 14 CFR Part 61 also includes the level of knowledge, training, operations proficiency, and experience a pilot must possess before being issued a certificate. This includes training and testing procedures. Title 14 CFR Part 63 is involved with certification of crew members other than pilots and Title 14 CFR Part 65 with airmen certification.

87.6 Operation Rules

Operational rules for manned aircraft operating in the U.S. NAS are prescribed in Title 14 CFR Part 91, which applies to all aircraft with the exception of moored balloons, kites, unmanned rockets, unmanned free balloons, and ultralights

[Title 14 CFR §91.1]. Part 91 also establishes the responsibility for aircraft operators to support the continued airworthiness of each airplane [Title 14 CFR §91.1].

The person ultimately responsible for the operation of the aircraft is the pilot in command [Title 14 CFR §91.3]. The pilot is also responsible for evaluating the airworthiness of the aircraft and determining if it is in a condition safe to fly [Title 14 CFR §91.7]. After the aircraft has been deemed safe to fly and before takeoff, the pilot needs to be familiar with any information concerning the flight, such as weather reports, fuel requirements, airport characteristics, and aircraft performance characteristics [Title 14 CFR §91.103].

To minimize the risk of collisions, no person is allowed to operate an aircraft in close proximity to another [Title 14 CFR §91.111], and when the weather conditions permit, the pilot should be alert in order to *see and avoid* other aircraft [Title 14 CFR §91.113]. Additionally right-of-way rules are established [Title 14 CFR §91.113]. With the exception of water operations, typically the aircraft with less maneuverability has the right-of-way. This rule is superseded when an aircraft is in distress, at which time it has the right-of-way with respect to all other air traffic. In general during emergencies pilots are allowed to deviate from the requirements of Part 91, even contrary to ATC instruction, provided that Air traffic control (ATC) is notified of this deviation as soon as possible [Title 14 CFR §91.3, §91.123]. In any other situation, no one is allowed to deviate from ATC clearance and instructions [Title 14 CFR §91.123].

Additional safety regulations do not permit pilots to fly below 10,000 ft or in proximity of Class B, C, and D airspace at speeds exceeding 250 and 200 knots, respectively [Title 14 CFR §91.117]. Similarly, minimum safe altitudes are established so that upon catastrophic failures, an emergency landing can take place without undue risk to people or property [Title 14 CFR §91.119].

87.6.1 Flight Rules

Title 14 CFR Part 91 defines two types of flight rules: visual flight rules (VFR) and instrument flight rules (IFR). In addition to the normal operations, Title 14 CFR Part 91 includes guidelines for emergencies as well as special operations like aerobatics, towing, and parachuting.

87.6.1.1 Visual Flight Rules

Under VFR the pilot is expected to control the aircraft's trajectory and avoid other aircraft based on visual cues, although separation instruction may be provided by ATC when flying in certain classes of controlled airspace.

A prerequisite to flying under VFR is the presence of enough fuel onboard, so that the aircraft can reach its first landing destination and fly for 30 min or 45 min after that during the day or night, respectively [Title 14 CFR §91.151]. Similar requirements exist on the flight altitude and weather conditions [Title 14 CFR §91.155]. The minimum weather conditions for VFR operations are summarized in Table 87.3.

Table 87.3 Weather minimums for VFR operations (Source: [Title 14 CFR §91.155])

| Airspace | Visibility (statute miles) | Distance from clouds | | |
|---|-------------------------------|----------------------|----------|--------------|
| | | Above | Below | Horizontal |
| Class A | N/A | N/A | | |
| Class B | 3 | Clear of clouds | | |
| Class C | 3 | 1,000 ft | 500 ft | 2,000 ft |
| Class D | 3 | 1,000 ft | 500 ft | 2,000 ft |
| Class E (<10,000 ft) | 3 | 1,000 ft | 500 ft | 2,000 ft |
| Class E (≥10,000 ft) | 5 | 1,000 ft | 1,000 ft | 1 stat. mile |
| Class G (day, ≤1,200 ft) | 1 | Clear of clouds | | |
| Class G (day, >1,200 ft and <10,000 ft) | 1 | 1,000 ft | 500 ft | 2,000 ft |
| Class G (night, <10,000 ft) | 3 | 1,000 ft | 500 ft | 2,000 ft |
| Class G (≥10,000 ft) | 5 | 1,000 ft | 1,000 ft | 1 stat. mile |

The information required when filing a VFR flight plan includes, besides the aircraft and pilot identification data, the point and time of departure, the route, altitude and airspeed, the amount of fuel onboard, and the point and time of arrival [Title 14 CFR §91.153].

87.6.1.2 Instrument Flight Rules

Flight under IFR is subject to similar restrictions in terms of fuel availability [Title 14 CFR §91.167], but in addition to that the presence of an operational and properly maintained VHF omnidirectional range (VOR) radio navigation system is also required [Title 14 CFR §91.171].

In order to operate IFR in controlled airspace, the pilot must first submit an IFR flight plan and receive appropriate ATC clearance [Title 14 CFR §91.173]. Once in controlled airspace, the pilot is required to monitor the appropriate communication frequency and report to ATC when she/he reaches predetermined points, encounters unforecast weather conditions, or other problems that may affect flight safety [Title 14 CFR §91.183].

87.6.2 Emergency Rules

In the case of disasters, the FAA administrator can issue a Notice to Airmen (NOTAM) designating an area within which temporary flight restrictions apply [Title 14 CFR §91.137]. In this case, no aircraft can enter the designated area except for aircraft participating in hazard relief, carrying law enforcement officials or any other category specified in [Title 14 CFR §91.137].

The FAA can also use the NOTAM system to issue new emergency traffic rules or regulations, whenever conditions for safe operations under normal rules are or will not be sufficient [Title 14 CFR §91.139].

87.6.3 Maintenance Requirements

The person responsible for the maintenance of the aircraft is either the owner or the operator who may not operate the aircraft unless the inspection and replacement intervals as well as any other prescribed maintenance procedures have been complied with [Title 14 CFR §91.403]. More specifically an annual inspection is required along with an inspection for the issuance of an airworthiness certificate [Title 14 CFR §91.409]. Additional inspections may be required for other types of aircraft.

After any maintenance procedure, the aircraft must be approved to return to service by an authorized person and the maintenance record has been updated [Title 14 CFR §91.407]. In some cases, an operational inspection by a pilot with adequate rating may be required [Title 14 CFR §91.407].

In addition to that, Title 14 CFR Part 91 requires operators to “support the continued airworthiness” of each airplane, by revising the inspection schedule, incorporating any design changes and revisions to the instructions for continued airworthiness [Title 14 CFR §91.1501].

87.7 Airspace Classes

Depending on the altitude and proximity to airports, the NAS is segregated into several classes as shown in Fig. 87.1. For each airspace class, different operating rules may be in effect, based on the stipulations of Title 14 CFR Part 91.

Classes A through E, ordered from most restrictive to less restrictive, correspond to controlled airspace. Wherever different airspace classes overlap, the most restrictive designation applies [Title 14 CFR §71.9].

Airspace between 18,000 ft above mean sea level (MSL) to about 60,000 ft belongs to the Class A airspace [Title 14 CFR §71.33]. Class A airspace is reserved for IFR traffic and an aircraft needs to receive ATC clearance before entering [Title 14 CFR §91.133]. There are also requirements for communication and transponder equipment.

Classes B [Title 14 CFR §71.51] and D [Title 14 CFR §71.61] include the airspace above and around airports of different sizes. They are designed to include traffic from/to the airport and ensure appropriate separation. In order to operate an aircraft in Class B airspace, ATC clearance is required from the facility controlling that area, and the pilot must possess at least a private pilot certificate although exceptions exist for other pilot certificates when specific requirements are met [Title 14 CFR §91.131]. In addition to the above, the aircraft needs to be equipped with appropriate communications and navigation equipment as well as an operating transponder [Title 14 CFR §91.131].

Requirements for operations in Class C and D airspace are similar, although more relaxed. In general, an aircraft entering these classes of airspace must establish

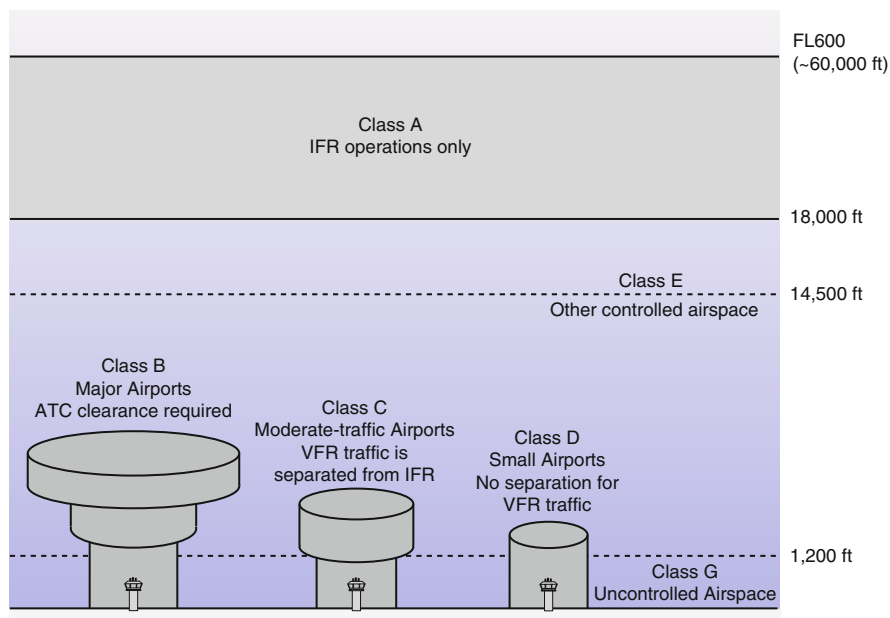


Fig. 87.1 Overview of the NAS classes. This figure depicts only a general view of the airspace classes. For the accurate extends of each airspace class, the reader is referred to the latest FAA Order 7400.9

and maintain communications with ATC while within that airspace, regardless of whether it is arriving or being only through traffic [Title 14 CFR §91.129, §91.130].

Finally, Class E corresponds to the rest of the controlled airspace. It includes the airspace from 14,500 ft up to the Class A boundary, as well as the airspace above 60,000 ft [Title 14 CFR §71.71]. In the vicinity of airports, class E airspace may extend down to the surface. Similarly in the proximity of federal airways, Class E airspace extends upwards from 700 or 1,200 ft above ground level. In the vicinity of an airport in Class E airspace, the pilot must establish two-way radio communication with ATC, although in the case of radio failure landing is still allowed, provided that VFR conditions exist, visual contact with the tower is maintained, and a clearance to land has been received [Title 14 CFR §91.127].

The last class is Class G airspace, which normally includes the space up to 1,200 ft above the ground but can be up to 14,500 ft. Although Class G airspace is also known as uncontrolled airspace, operating rules do apply. Helicopters and aircraft flying below 1,200 ft should typically operate clear of clouds and at speeds that allow the pilot to see and avoid other traffic as well as any obstructions on the ground [Title 14 CFR §91.155]. In the vicinity of airports, two-way radio communication with ATC must be established, and in addition to that, all turns must be made to the left, unless the airport indicates the opposite [Title 14 CFR §91.126].

Other restrictions may also be in effect depending on the type of the aircraft, such as avoiding crowded areas and noise limits.

Every year the FAA publishes a revised Order 7400.9, which includes the current airspace designations for the U.S. NAS.

87.8 Regulation Development Models

The traditional model of regulation development has been based on sufficiently mature technologies for which standards had been developed and possibly implemented. The regulatory body, in this case the FAA, undertakes the task of assessing the technology and standards available and develops appropriate regulations. Because of the aforementioned requirements, this process is slow, costly, and in some cases counterproductive since developed technology and standards are not necessarily adopted.

In March of 1996, the U.S. Congress recognizing the need to advance cooperation between industry and the Federal Government, signed Public Law 104–113, also known as the National Technology Transfer and Advancement Act of 1995. Section 12(d)(1) reads:

Except as provided in paragraph (3) of this subsection, all Federal agencies and departments shall use technical standards that are developed or adopted by voluntary consensus standards bodies, using such technical standards as a means to carry out policy objectives or activities determined by the agencies and departments.

This “industry consensus” model was recently used for the regulation of the LSA category. In this case, the FAA participated actively in the development of standards, and as a result, these standards were immediately incorporated into the regulatory framework upon publication. This approach is faster and more cost-effective, since the burden of drafting the standards is mostly with the industry. In addition to that, conformance with the standard is self-regulated and FAA involvement is limited to oversight and penalizing nonconformance, thus further reducing the cost to the Federal Government. More information about this model can be found in Anand (2007).

Regardless of the regulation model, another key characteristic of regulation development is the basis of the airworthiness certification. As discussed by Haddon and Whittaker (2002), military systems have been traditionally evaluated under a safety target approach. In this case, the aircraft is designed and operated with a particular role and operating environment in mind, and the airworthiness certification includes appropriate operational restrictions to ensure that an adequate level of safety is achieved.

According to manned aviation regulations, airworthiness is based on compliance with a code of requirements. This approach has the advantage of no interlinking between airworthiness and operation, facilitating a variety of applications as well as interoperability. Nevertheless, special safety targets are also included to evaluate new technologies or designs, not covered by existing code. These can be found in

paragraph 1309 of current Certification Specifications (CS) or the corresponding Acceptable Means of Compliance (AMC) sections.

References

- Academy of Model Aeronautics, 2008 Official national model aircraft safety code. Effective 1 Jan 2008 (2007)
- Air Transport Association (ATA), Learning center (2008), <http://learningcenter.airlines.org/>
- S. Anand, Domestic use of unmanned aircraft systems: evaluation of policy constraints and the role of industry consensus standards. *J. Eng. Public Policy* **11** (2007)
- ASTM International, Standard practices for unmanned aircraft system airworthiness. Standard F 2501-06 (2004)
- Federal Aviation Administration, Appendix A, in *FAA System Safety Handbook* (Washington, DC, USA, FAA, 2000)
- Federal Aviation Administration, Airworthiness certification of aircraft and related products. Order 8130.2F (2004)
- Federal Aviation Administration, Unmanned aircraft operations in the national airspace system. Docket No. FAA-2006-25714 (2007)
- D.R. Haddon, C.J. Whittaker, *Aircraft Airworthiness Certification Standards for Civil UAVs* (UK Civil Aviation Authority, London, 2002)

Human Factors of Unmanned Aircraft System Integration in the National Airspace System

88

Bill Kaliardos and Beth Lyall

Contents

88.1 Introduction 2136

88.2 Pilot Operations at the Control Station 2138

 88.2.1 Gap: Loss of Natural Sensing 2138

 88.2.2 Gap: Control Loop Degradation: Data Dropouts 2140

 88.2.3 Gap: Control Loop Degradation – Latency 2142

 88.2.4 Gap: Link Monitoring 2143

 88.2.5 Gap: Automation and Automated Systems 2144

 88.2.6 Gap: Information Formats 2146

88.3 Air Traffic Control 2148

 88.3.1 Gap: Different Aircraft Performance and Missions 2149

 88.3.2 Gap: Visual Instructions 2150

 88.3.3 Gap: Lost-Link Predictability and Procedures 2151

 88.3.4 Gap: Communications 2152

 88.3.5 Gap: Workstation Information 2153

 88.3.6 Gap: Detection of Small-Size UAs 2154

88.4 Conclusions 2154

References 2157

Abstract

The purpose of this chapter is to identify human factors challenges to integrating Unmanned Aircraft Systems (UASs) in the National Airspace System (NAS), for both pilots and air traffic controllers. The method for identifying these challenges

B. Kaliardos (✉)
Federal Aviation Administration, Washington, DC, USA
e-mail: Bill.Kaliardos@faa.gov

B. Lyall
Research Integrations, Incorporated, Beaverton, OR, USA
e-mail: beth.lyall@researchintegrations.com

was primarily based on the differences or “gaps” between manned aircraft in the NAS today and the unique aspects of UASs. The goal is not to generate a comprehensive list of human factors issues, but to focus on those that are traceable to fundamental characteristics of UASs (primarily due to the UAS pilot being physically remote from the aircraft) and that are also considered challenging with respect to the current NAS and its regulatory framework.

Current UAS designs do not generally emphasize NAS integration considerations, and approvals for airworthiness and operations are based on special operational limitations determined on a case-by-case basis. This chapter assumes that, for integration into the NAS, future UASs will be much more standardized in many respects than in the past – closer to that of today’s civil manned aircraft, which are subject to rigorous design, operational, manufacturing, and training approvals by the Federal Aviation Administration (FAA). Under such assumptions, human factors is not only central to the very definition of UAS but is also central to some of the most important challenges facing the integration of UAS in the NAS.

The human factors gaps are discussed in two parts. The chapter first addresses human factors for pilot operations at the control station covering the loss of natural sensing, control loop degradation, link monitoring, automation, and information formats. The chapter then addresses human factors for air traffic controllers, which covers aircraft performance and missions, visual instructions, lost-link predictability and procedures, communications, workstation information, and detection of small-size aircraft.

88.1 Introduction

Human factors (HF) entails a multidisciplinary effort to generate and compile information about human capabilities and limitations and apply that information to equipment, system facilities, procedures, jobs, environments, training, staffing, and personnel management for safe, comfortable and effective human performance (Federal Aviation Administration, Order 9550.8). When the human is a functional element within a larger system such as the National Airspace System (NAS), human factors is important in overall system performance such as safety and efficiency of aircraft operations. In this chapter, “human-system integration” (HSI) is used interchangeably with “human factors,” but the discussion here is primarily limited to equipment and procedures.

For Unmanned Aircraft Systems (UASs), the question often arises: “How can there be human factors for an aircraft that does not have a human onboard?” While it is true that the human is not physically integrated with the aircraft, the human is functionally integrated with the aircraft and, in fact, physically integrated with the unmanned aircraft *system*. That system typically comprises of the unmanned aircraft (UA), control station, and control/communication (C2) links. Therefore,

human factors is an important part of UAS design and operation, particularly when considering integration with the NAS.

In this chapter, “integration” in human-system integration applies to two conditions of humans (H) and systems (S):

- When the S in HSI refers to UAS and the H refers to the pilots
- When the S in HSI refers to NAS and the H refers primarily to the pilots and/or air traffic controllers (ATC)

UAS in the NAS can therefore be considered as a system of systems, in which multiple humans are functional elements.

The perspective presented here focuses on civil UAS with minimum operational restrictions, unlike the current environment. Today, public operators (such as the military services) versus civil operators dominate UAS operations in the NAS with an emphasis on mission effectiveness in segregated airspace. Furthermore, much of the near-term growth of civil UAS is expected for UAs that are smaller than manned aircraft and with strict operational limits such as line-of-sight control. However, it is assumed here that the true integration of UASs in the NAS will be based on nonsegregated airspace with minimum operational restrictions. This would present unique human factors challenges that are driven primarily by safety of the flying and nonflying public and workload for air traffic controllers, but constrained by manned aircraft operations, air traffic services, certification services, etc. FAA policy (e.g., Notices 8900.227, 7210.846), Civil UAS Integration Roadmap, and Concept of Operations (draft) describe similar assumptions, scope, and other relevant background information in more detail.

The “lessons learned” from UAS so far (e.g., accident statistics, air traffic control experience) are relevant, but need to be carefully treated because they do not necessarily represent tomorrow’s UAS. It is expected that future UAS integration into the NAS will be accomplished not on a case-by-case basis (such as with today’s Certificate of Authorizations or Special Airworthiness Certificate – Experimental Category), but likely through the development and application of standardization. That is, future UASs are expected to be much more standardized in many respects than in the past – closer to that of today’s civil manned aircraft, which are subject to rigorous design, operational, manufacturing, and training approvals by the FAA and based upon a combination of federal regulations, policy, guidance material, and industry standards.

Nevertheless, UAS approvals will still be different in many respects than manned aircraft due to fundamentally unique characteristics of UAS design and architecture. The fundamental differences – the “gaps” – between UAS and manned aircraft provide a basis for understanding the human factors challenges for UAS integration in the NAS.

There is much to understand to effectively integrate UASs into the NAS, and a good starting point is to describe the potential gaps between what is known about manned aircraft and operations and what is expected for UASs. While similar UAS human factors analyses have been done in the past (e.g., McCarley and Wickens 2005), the gaps presented here are primarily from the FAA regulatory guidance, and

policy perspective. The remainder of this chapter is organized around these gaps and presented in two parts: pilot operations at the control station and air traffic control.

88.2 Pilot Operations at the Control Station

The first category of UAS human factors considerations pertains to pilot operations at the control station. For current approvals, FAA policy (e.g., Interim Operational Approval Guidance 2008, Notices 8900.227 and 7210.846) specifies a “pilot” or “pilot in command” that is responsible for the UAS flight operation, which is referred to here for brevity as the UAS “pilot.” The control station provides the interface between the pilot and aircraft, as well as between the pilot and other UAS and NAS elements.

88.2.1 Gap: Loss of Natural Sensing

Pilots of unmanned aircraft possess an awareness of the aircraft and operation that is qualitatively different from manned aircraft, and a key reason for this is because pilots are not colocated with the UA and therefore not able to naturally sense the aircraft and its local environment. To naturally sense means to use physiological senses (visual, aural, tactile, proprioceptive, vestibular, etc.) to perceive information directly, without the aid of technology such as artificial sensors and displays. Information is sensed “locally” in that pilots unaided by technology only see and hear, for example, within a limited volume around them. For manned aircraft pilots, examples of naturally sensed information include aircraft orientation and turbulence through “seat of the pants” flying, engine power through aural and tactile sensing, and traffic and terrain through viewing out the windshield. For unmanned aircraft pilots, locally sensed information (near the control station) is of limited value, even with direct line-of-sight of the UA. Although technology can mitigate and sometimes enhance certain information, the fact remains that unmanned aircraft pilots simply do not use their natural senses in the same way as manned aircraft pilots.

88.2.1.1 Loss of Natural Vision

Natural vision may be the most critical difference between manned and unmanned aircraft pilots. For example, for manned aircraft, the aircraft windshield is critical for natural vision. Title 14 of the Code of Federal Regulations (14 CFR) 25.773 states, “Each pilot compartment must be arranged to give the pilots a sufficiently extensive, clear, and undistorted view, to enable them to safely perform any maneuvers within the operating limitations of the airplane, including taxiing takeoff, approach, and landing” (see also 14 CFR 23.773). Natural vision can be used in a variety of important pilot tasks, such as to detect and assess environment states (e.g., traffic, terrain and obstacles, foreign objects, weather) and aircraft states (e.g., orientation,

speed, icing, engine flameouts). Guidance on windshield criteria for visibility such as field of view and optical properties are provided in FAA Advisory Circular (AC) 25-773-1, for example. UAS compensatory strategies such as aircraft cameras (visible spectrum) might provide some level of equivalency to meet the applicable equipment and operational pilot visibility requirements, but so far such equivalency, means of compliance, or new regulations remain to be determined. The problem is complicated by data link bandwidth limits, which can constrain field of view (instantaneous and overall), resolution, latency, frame rate, etc. In short, in the absence of natural vision of an onboard pilot, UAS currently cannot comply with Visual Flight Rules (VFR) or comply with any clearance that includes a visual component.

Even in instrument flight rules (IFR) there are many visual operations. Manned aircraft pilots are required to see certain features when operating on or near the surface. For example, 14 CFR 91.175 is an operational regulation, which includes requiring that visual references such as the approach light system, runway, and threshold are distinctly visible and identifiable to the pilot when operating below specified altitudes. Pilots of UAs obviously cannot look directly through the windshield for visual identification, but technologies may be able to provide compensatory alternatives for safe operation in which pilots rely solely on artificial sensing and displays. In fact, manned aircraft today make use of enhanced flight vision systems (EFVS) to provide supplemental information to receive operational approvals for landing in lower visibility conditions, but only with displays such as HUDs that *complement* natural vision with the pilot seated in a normal position and line of vision and looking forward along the flight path. (Under 14 CFR 91.175(l), a pilot may use an approved EFVS to descend below decision altitude (DA) or minimum descent altitude (MDA) to 100 feet above the touchdown zone (TDZ) elevation (TDZE) from a straight-in instrument approach procedure (IAP) other than category II (CAT II) or category III (CAT III).) An EFVS uses imaging sensing technologies, such as in the long-wavelength spectrum (e.g., infrared, microwave) to provide a real-time enhanced image of the external scene topography to the pilot. For manned aircraft, EFVS operations are proof that electronic imaging can be used beyond “situation awareness” (which is what would be needed for UAS), but only in the context of clearly defined procedures that are based on natural vision.

Similarly, some manned aircraft also have installed synthetic vision systems (SVS), which utilize a terrain and obstacle database that is accessed and displayed according to real-time aircraft position and attitude data. Although SVS has not yet enabled operational credit, such concepts continue to be discussed. As with EFVS, critical to these discussions is also how SVS is used by pilots that can also view the external scene through a windshield.

Both EFVS and SVS are relatively new technologies for manned aircraft, and they may offer lessons for compensatory information for UA operation. However, the use of real or synthetic images as a replacement versus a complement to natural vision presents many challenges for integrating UAS in the NAS. Many other related NAS integration challenges will likely be encountered, such as those mentioned for standard non-enhanced visible spectrum imaging (e.g., field of view, camera and

display resolution, image latency), as well as database integrity, and procedures for normal and non-normal conditions. For more information on EFVS and SVS, see AC 90-106 and AC 20-167.

88.2.1.2 Loss of Nonvisual Sensing

Another consequence of the UA pilot's inability to use natural senses involves the nonvisual senses. While visual modalities often involve focused attention, such as in a pilot's sequential instrument scan, other senses offer the advantage of acquiring information without such directed attention. For example, in manned aircraft, engines create background noise that is often simple to detect when it changes, as in a loss of power. Likewise, turbulence moves and orients the (aircraft and) pilot, with naturally salient cues to their vestibular, haptic, and proprioceptive senses to detect orientation, force, and position. Vibration and buffeting can adversely affect pilot fatigue and control but can also provide valuable flight envelope cues (e.g., as described in 14 CFR 23.251). These examples illustrate how natural sensing can be effective for easily detecting certain manned aircraft conditions, which, for UAs, is practically impossible without artificial means. Compensatory technologies include the reproduction of, say, aural cues, motion/vibration, and control force feedback at the control station. The same information can also be represented visually on a display, but such displays can add to visual overload, and the effectiveness of reproducing non-visual aircraft cues visually at the control station remains to be determined. These issues need further investigation to determine the nature and quality of information required at the UA control station.

The consequences of UA pilots not being able to directly sense the aircraft and its environment depend on the type of operations and the compensatory technologies used at the control station and the aircraft (Williams 2008). Flying manned aircraft in the NAS currently is based on regulations, guidance, and standards that were developed with the assumption that pilots are able to obtain important information, visual and otherwise, through natural sensing of the aircraft and its local environment. A challenge for integrating UAS in the NAS is to develop a new or modified basis to account for the loss of this naturally sensed information.

88.2.2 Gap: Control Loop Degradation: Data Dropouts

In the feedback control of aircraft, pilots provide a variety of system inputs and receive feedback on their effects. The focus in this section is on flight control. However, the human factors effects of control loop degradation apply to other systems as well.

The UAS control function through the wireless data link is inherently less reliable than the direct hardwired links used in manned aircraft. Link degradation can occur in both paths of the feedback control loop: (a) downlinked "feedback" information to the control station displays and (b) uplinked "control" information to the aircraft. While both information components often degrade simultaneously, each may also degrade separately. Furthermore, degradation is often in the form of relatively short

intermittent periods or “dropouts,” which is the focus of this section. Link failures of relatively long duration, in which the aircraft is in autonomous “lost-link” mode, are discussed in other sections of this chapter, such as Sects. 88.2.5 and 88.3.3.

88.2.2.1 Feedback, Downlinked Portion

The extent of control loop degradation is a unique issue with UAS. For comparison, it is worth mentioning that degradation does occur to some degree in the manned aircraft equivalent. In particular, for the feedback “information” portion of the control loop, even hardwired display systems can potentially suffer from data dropouts, whether due to the sensor, data bus, processing, or display systems.

However, *wireless* feedback control loops introduce UAS flight control issues never before experienced with manned aircraft. Consider the downlinked or telemetry portion, in which sensor and other data collected at the UA are transmitted to the control station, for processing by computers or humans. In today’s UAS, it is common for short dropouts to occur (Williams 2008). To address this, current UAS architectures and operations reduce the likelihood of intermittent dropouts to noncritical flight phases, such that the highest reliability is achieved during critical phases such as takeoff and landing. However, operation-specific system behavior such as primary flight displays that depend on the phase of flight might be unacceptable, based on, for example, 14 CFR 23.1301 and 23.1309. It remains to be determined what will be acceptable for UAS.

Existing guidance that could possibly apply to UAS data dropouts in the feedback portion of the control loop include AC 25-11A, which provides display-related system safety guidelines:

There should be a means to detect the loss or erroneous primary flight information, either as a result of a display system failure or the failure of an associated sensor. When loss or malfunction of primary flight information is detected, the means used to indicate the lost or erroneous information should ensure that the erroneous information will not be used by the flight crew (for example, removal of the information from the display or placement of an ‘X’ through the failed display).

The above guidance was likely written under the assumptions of infrequent occurrence of data dropouts and with backup standby displays and complementary views out the window. For UAS, the occurrence of dropouts, their indication, and the procedures to accommodate them will need to be developed, factoring in the likely reduced availability of backup or secondary sources to provide similar information (which might use the same link). Additionally, transitions between good data and stale data at the control station (e.g., “freezing and unfreezing” the displays) will add more challenges to the effects of degradation of the feedback portion of the control loop.

88.2.2.2 Control, Uplinked Portion

For the uplinked portion of the control loop, manned aircraft operation does not provide a useful analogy from direct control links, whether mechanical, hydraulic, or fly-by-wire. All manned aircraft applicants complying with 14 CFR 2x.1309 (part 23, 25, 27, 29) must show that “the occurrence of any failure condition which

would prevent the continued safe flight and landing of the airplane is extremely improbable.” It is unclear how the UAS control uplink will be treated from a system safety perspective relative to manned aircraft control systems. UAS will likely require entirely new policy to accommodate control uplink dropouts, covering a wide range of human factors areas such as:

- Prediction of control uplink dropouts
- Indication and alerting of control uplink dropouts
- Distinguishing feedback (downlink) dropouts from control uplink dropouts
- Control uplink dropout transitions between fully functional and degraded link
- Aircraft automation behavior during control uplink dropouts

Aircraft automation behavior is particularly critical, not only for appropriate behavior during fully automated periods, but also during intermittent dropout periods. Aircraft automation behavior also affects air traffic controllers (see Sect. 88.3.3 in this chapter).

88.2.3 Gap: Control Loop Degradation – Latency

Latency in the control loop can be problematic in both manned aircraft and UAS, but for different reasons. In the manned aircraft control loop, for example, latencies did not present challenges until the incorporation of automation and computers in control systems. These introduced small signal delays in various ways, including filtering, processing, and networking. Manned aircraft simulators were especially prone to time delays in the presentation of electronic visual imagery. From the human factors perspective, time delays can lead to increased workload, larger flight technical errors, and control instability due to what is commonly known as pilot-induced oscillation (PIO) or aircraft-pilot coupling (APC). Generally, time delay effects in manned aircraft control are well-understood problems in the context of defined physical attributes (aerodynamics, mass distribution), control tasks (e.g., landing), control system design (e.g., gain), and disturbances (e.g., crosswinds). Therefore, regulations such as 14 CFR 2x.143 and flight test procedures such as those described in AC 23-8C and AC 25-7B address overall aircraft controllability and stability, but not the specific role of latency effects.

Unmanned aircraft introduce a new regime of control loop latency due to the wireless link. Two classes of wireless links can be considered for simplicity, each at opposite latency extremes: line-of-sight and satellite. Line-of-sight links produce the smallest time delays (usually less than a few tenths of a second) with signal transport delays accounting for a negligible contribution to the total delay. However, line-of-sight operations are limited in the distance between control station and aircraft. For beyond-line-of-sight, satellite links produce the largest time delays (often on the order of a second), in which signal transport delays are a non-negligible portion of the total latency. Latency is not necessarily due to degradation, but degradation can exacerbate latency effects.

The consequences of control loop latency depend on many factors, such as the pilot task and aircraft dynamics. For a given time delay, adverse effects in flight

control tend to increase with dynamic tasks such as landing, and aircraft that are highly responsive. In contrast, sluggish aircraft and steady flight without rapid maneuvers are more tolerant to time delays. This is a key reason why today's UASs often use satellite-based control links for the relatively static maneuvers and flight phases (e.g., enroute), whereas line-of-sight links are often used for takeoff and landing.

Because of the interdependencies of many factors, tomorrow's UAS might need to address time delay effects through combinations of aircraft design (responsiveness and stability), control link architecture and performance, control station feedback, control input devices, maneuver limitations and pilot training, and automated "inner-loop" aircraft control. In addition, novel techniques such as model-based feed-forward control may provide further mitigation of time delays when pilots are in the control loop. A more extreme mitigation, out of the scope of this section, is to remove the human from certain control tasks altogether, thereby circumventing the latency effects of a remote operator (see also Sect. 88.2.5 in this chapter). But for UA control tasks in which the human is in the loop, a key challenge for NAS integration will be to develop regulations and standards to address control loop time delays, despite the many variables that contribute to these delays and their effects.

88.2.4 Gap: Link Monitoring

A wireless data link for aircraft control is unique to unmanned aircraft and therefore not addressed in today's manned aircraft regulations and standards. Because the data link is critical for safe operation, yet is subject to degradation and failure modes very different from control links in manned aircraft (e.g., cables, fly-by-wire), link monitoring might be needed by the pilot or other flight crew members. However, the scope of this section is limited to link monitoring by the UAS pilot using minimal automation because it is a new and significant task relative to manned aircraft pilot duties. The term "link monitoring" is used in this chapter to include tasks that maintain link health as well.

The task of link monitoring requires both system knowledge and human interface equipment at the control station. System knowledge can include an understanding of the available architectures (e.g., satellite, line-of-sight, ground based), signal-in-space behavior, aircraft orientation and range effects, weather and ground effects, data degradation observability, and random dropout behavior. In today's UAS, dedicated link specialists are often responsible for link system knowledge, coordinating with the pilot either remotely or near the control station. In other UASs, pilots – not dedicated link specialists – accommodate the additional tasks associated with link monitoring. It has not yet been determined whether these additional tasks impose unacceptable workload on the pilot. For example, 14 CFR 2x.1523 for manned aircraft requires a determination of the required minimum crew to operate the aircraft. Applicants complying with these regulations must consider individual pilot workload, accessibility and ease of operation of control by the appropriate

crew member (including the possibility of one being incapacitated), and certain operations (see also ACs 23.1523 and 25.1523-1). When adapting these regulations to UAS, analyses should include the added workload of link monitoring and control access in order to establish minimum crew requirements for safe operation.

The link-monitoring task will also require equipment in the control station that involves a human interface. Manned aircraft, even with advanced fly-by-wire systems, have no equivalent to this, outside of extremely improbable system failures of the primary flight controls and their annunciations. Yet, the control station might have to include displays and controls for control link monitoring, in addition to equipment for typical aviation functions such as communications, navigation, and surveillance. Standards will need to be developed to address, at a minimum, data link display location and information and controls requirements, with considerations for the context of the control station (see ACs 25-11A, 23-1311C, 20-175 for flight deck displays and controls and their integration in the flight deck).

88.2.5 Gap: Automation and Automated Systems

UAS designs require the use of many types of automated systems, some different than those used in manned aircraft. The term “automation” is used frequently in UAS discussions, but can mean different things to different people. An automated system or function can generally be defined as one that is performed by a system, which would otherwise be performed by a human operator. Although, in the context of UASs, the broad terms, “automation” or “automated,” are often used to refer to the aircraft as a whole, it is more useful to refer to the specific functions or systems, such as the control of navigation and energy states (e.g., speed, position, flight path). It is also useful to distinguish along other dimensions such as “information automation” versus “control automation.” Often information automation is used to describe systems or functions that process and organize information that would otherwise need to be processed or organized by the human in the system. Likewise control automation is used to describe those functions or systems that are used to control the aircraft or other elements of the UAS (e.g., autoflight or autothrust). The functions can fall under classic categories such as “aviate,” “navigate,” and “communicate,” as well as “hazard avoidance,” “systems management,” “preflight planning,” and other areas. For each of the systems, a design decision must be made about how and when to automate each function – whether it will be accomplished exclusively by a system, exclusively by the pilot (or other human operator), by some combination – or how and when to transition from one to the other. The approach to these design decisions should be consistent for all systems in the UAS; the description of this is often referred to as the automation design philosophy.

The incorporation of advanced automated systems in manned aircraft is a useful analogy for unmanned aircraft. While it is often thought that UAS is unique for its extended use of automated systems, it is important to understand and appreciate the highly sophisticated automated systems that are common to many types of manned aircraft and learn from lessons acquired through the many years of design and implementation of those systems. Flight management computers and integrated

modular avionics are examples of highly complex hardware and software for both information and control automation, including self-monitoring for detection and isolation of faults that would otherwise ripple through other systems.

Most of the same issues with automated system use, risks, and benefits found in manned aircraft also apply to UAS, but UASs present additional and unique challenges. Many of these unique automation challenges can be notionally classified as either pertaining to intelligence limits that drive the “maximum” levels of automation, or latency effects that drive the “minimum” levels of automation.

88.2.5.1 Intelligent Automation (Maximum Levels of Automation)

UAS has automation needs that push the limits of system performance toward more intelligence and sophistication, beyond what is currently used today in manned aircraft. The driver for this is primarily the potential for lost control link (“lost link”). Under lost link, humans are no longer functionally part of the decision process and control loop, and the UA’s onboard automated systems function completely without humans during these non-normal conditions. This is not to say that automated systems are going to “replace” humans in function and performance, but that they must be designed to be able to operate at times without the human complement. This is a critical difference. In today’s manned aircraft, automated systems are regulated and approved within the paradigm of colocated human interaction. An important part of safety analysis, regulatory compliance, and design is the underlying assumption that pilots are able to mitigate certain equipment failures and errors from interaction with automation (14 CFR 2x.1309, 14 CFR 25.1302). For example, pilots monitor autoland systems during automated landings, with the ability to immediately intervene as necessary if a system failure occurs or in any situation that can compromise continued safe operation. Another example is alerting systems, which typically are designed with an acceptable proportion of false/nuisance alerts and to which pilots apply additional information, context, and judgment when determining the appropriate response. When automated systems are designed to operate “autonomously” – without the option of human intervention – there are a number of decision-making consequences to performance and safety. These decision-making consequences can include the loss of:

- Judgment that comes with pilot expertise and survival instinct
- Responsibility and accountability
- Access to additional information sources
- Context
- Understanding and prioritization of goals
- Adaptability to unforeseen events
- Visual pattern recognition for identifying features
- Air traffic communication using natural language

A critical regulatory challenge for UAS might be not only to adapt the current design regulations from manned to unmanned use but also perhaps to use an *operational* requirement, such as 14 CFR 91.111 (a) – “No person may operate an aircraft so close to another aircraft as to create a collision hazard” – to create an equivalent *equipment* requirement. This would ultimately call for deriving

well-defined functional and performance criteria for equipment, from operational requirements that were intended to be complied with by human operators and not systems.

While there might not be hard theoretical limits to automation achieving certain functions or behaviors, there are “soft” limits due to practical factors such as design knowledge, assumptions of future environments, and software validation and verification. These practical factors establish, for the time being, what is essentially an approximate upper limit or the “maximum automation” level in NAS-integrated UAS.

88.2.5.2 Latency Effects (Minimum Levels of Automation)

In contrast to the upper limits in the incorporation of UAS automation due to the lost value of the human complement, there might also be lower limits of automation due to the remote pilot’s limited ability to perform inner loop control. One reason is latency effects from the human being remotely located; some UAS links introduce delays in the control or communications loop that limit the degrees to which humans can be effectively involved (see also Sect. 88.2.3 in this chapter). The lowest level of automation that is often referred to as “manual” control, which is standard in nearly every manned aircraft, might not even be an option in unmanned aircraft due to the time delays associated with the control link. For many control tasks, automated systems could be needed to augment human control, such as through inner-loop automation with human supervisory control, through manual control with feed-forward compensation, or other means. The elimination of manual control, which is one example of raising the “minimum automation” threshold, will likely affect regulations that govern primary flight controls, such as 14 CFR 23.143 and 23.155. The elimination or limitation of manual control will also affect policy that govern pilot responsibilities in rapid response operations, such as maneuvering due to traffic, terrain, windshear, unusual attitude, stall, aircraft system failures, and perhaps even compliance to air traffic control. Time delays in the control link therefore have significant effects in human-automation interaction, and could drive the establishment of a new minimum automation level.

Extensive use of aircraft system automation is not unique to UAS. However, for UASs that are integrated in the NAS, both the minimum level and maximum levels may necessarily be shifted upward, toward higher levels of automation, when compared to manned aircraft. This in turn has human factors implications for pilots at the control station because of human-automation interaction effects. It also has human factors implications for air traffic controllers (see Sect. 88.3.3 in this chapter).

88.2.6 Gap: Information Formats

The manner in which information is presented to pilots, and input from pilots, may differ between manned and unmanned aircraft. While information requirements dictate “what” is needed on a display, information formats dictate “how” that

information is represented. Hence, the same information may be needed by pilots of both manned and unmanned aircraft, but the unique characteristics of UAS may call for different formats.

88.2.6.1 Example: Aircraft Attitude Format

One example is the reference frame for aircraft attitude. Attitude indicators show aircraft orientation relative to the earth or, specifically, the roll and pitch angles of the aircraft relative to the horizon. The two primary components of an attitude indicator are the aircraft symbol, or “miniature wing,” and the horizon bar. In manned aircraft, it is most common to fix the aircraft or “miniature wing” symbol to the airframe or body axis, such that it maintains the same orientation relative to the actual aircraft and normally seated pilot. A key reason for this is to match what the pilot sees out the window. That is, the horizon as seen out the window moves similarly to the horizon bar, from the pilot’s perspective. This “egocentric” format is the common format for attitude indicators (some Russian aircraft, for example, are notable exceptions).

The attitude indicator in a control station is not necessarily in the context of an aircraft-centric, out-the-window view, as in manned aircraft, and therefore could warrant a different attitude display format. Although much less common in (western) manned aircraft, exocentric format attitude indicators are allowable. For example, AC 25-11A states that both fixed airplane reference and fixed earth reference bank pointers are acceptable for primary attitude information, although “a mix of these types in the same flight deck is not recommended.” UAS pilots, being physically remote from the aircraft, do not use natural vision through a windshield to sense the environment local to the aircraft, although this view can be replicated with the appropriate sensors and displays (see Sect. 88.2.1 in this chapter). It may be more appropriate to display unmanned aircraft attitude with an exocentric view, in which the horizon bar is fixed, and aircraft symbol appears to move relative to that (and relative to the pilot). Exocentric formats inherently reflect the nonmoving and remote nature of the UAS pilot, which is a key difference between manned and unmanned aircraft operation. However, it is not clear which format is the most appropriate, and other factors such as negative transfer of learning must also be considered.

88.2.6.2 Environmental Information Format

Beyond attitude indicators, other format examples to consider include displays that represent the external environment, such as traffic (FAA Technical Standard Order (TSO)-C147, -C118, -C119c, -C195a), weather (TSO-C63d, -C105), and terrain (TSO-C151b). With manned aircraft, such displayed information is often comprehended in the context of natural vision out the windshield, as in the case with aircraft attitude. Traditionally, it has been common to provide this information on exocentric views (e.g., planview moving maps). Recent trends in manned aircraft to display environmental information on forward-looking egocentric-format displays might not be as appropriate for UAS.

88.2.6.3 Other Displayed Information Formats

It is likely that much of the displayed information at a control station can benefit from the application of manned aircraft display standards, most of which derive from highly evolved designs based on sound design principles and best practices, and years of operations. For example, altitude and speed dials, tapes, and numeric indicators likely do not appear to present the need for UAS-specific changes. A similar argument might also apply to the “basic T” arrangement, the standard primary instrument display arrangement for manned aircraft, (as required in 14 CFR 23.1321, 25.1321, 27.1321), particularly when considering negative transfer of learning effects on UAS pilots with manned piloting experience (assuming such information is required for pilot control). The determination of new information formats for UAS control station displays should be derived, such as from the fundamental differences between manned and unmanned aircraft.

88.2.6.4 Control (vs. Display) Formats

Lastly, information formatting regarding *controls* (i.e., input device) versus displays may also differ between manned and unmanned aircraft. It is first helpful to distinguish aircraft *systems* controls (such as for transponder settings) from *flight* controls. While there are unique considerations for UAS systems controls such as the lack of vibration, maneuver g-forces, and ambient noise that can affect pilot performance, most aircraft systems controls such as buttons, knobs, touch screens, and cursor control devices (AC 20-175) do not appear to be fundamentally affected by the remote nature of the UAS control station. UAS primary and secondary flight controls might benefit from compensation for the loss of natural sensing, such as force feedback to comply with 14 CFR 23.255, but such techniques are already employed in manned aircraft, as in fly-by-wire control systems. New considerations for UAS flight controls include control link latency compensation, and effects of control device misalignment with the vehicle axis. The level of automation in UAS flight guidance and control is often cited as a reason for new flight control formats (e.g., supervisory control via a cursor control device), but manned aircraft offer multiple automation levels as well. Rather, a case for new UAS control formats might be based more on the lowest level of control automation, not the highest (see Sect. 88.2.5 in this chapter). For example, manned aircraft flight control systems are highly automated yet still provide the options for manual or quasi-manual control through control formats offered by yokes – a feature that might not exist with UAS. Therefore, an important consideration for integrating UAS in the NAS is for aircraft that no longer have traditional control formats (e.g., yokes) for primary flight control due to automated inner loop control systems.

88.3 Air Traffic Control

The second category of UAS NAS integration human factors considerations pertains to air traffic control. “Controller” is used generically here to include the many types

of air navigation service providers in various facilities, to whom FAA Order JO 7110.65 applies.

88.3.1 Gap: Different Aircraft Performance and Missions

The flight performance of UAs, such as speed, climb and descent rates, and altitude envelope, can be significantly different from manned aircraft. Air traffic controllers expect certain aircraft performance levels in the NAS in order to orchestrate traffic safely, efficiently, and under acceptable workload. Although FAA Order JO 7110.65, Appendices A–C, list performance values for a large variety of aircraft, controllers typically deal with an expected and well-known variety of performance groups. Aircraft that cannot meet the expected performance, or are significantly different, are potentially disruptive to controllers as well as pilots of nearby manned aircraft.

UAs therefore present NAS integration challenges because of performance differences from manned aircraft, both directly and indirectly. Although unique aircraft performance is not an inherent aspect of UAs, the lack of onboard pilots and cockpit equipment allows for comparatively small aircraft and power plants relative to manned aircraft, each with performance consequences. This has the potential to significantly increase the range of aircraft performance values that will be present in the NAS. For example, the lightest manned airplane weight category used in wake turbulence separation is currently “small” ($\leq 41,000$ lb, FAA JO 7110.65; FAA Aeronautical Information Manual) and “light” ($\leq 15,000$ lb, ICAO) maximum certified takeoff weight, yet UAs can easily weigh below 1,000 or 100 lb. (A special FAA rulemaking committee has defined “small” in another context, but “small-UASs” (FAA Order 1110.150) are not considered here to be NAS integrated and not in the scope of this chapter.) Such small weights, odd sizes, and other physical attributes can lead to a variety of aircraft performance not seen before, most notably in lowering the range of aircraft air speed and change rates (vertical rates). As an example consequence, a slow aircraft can require increased controller monitoring and workload, with necessary clearance amendments issued to move traffic around it (increasing fuel burn) or to decelerate traffic (leading to delays). As another example consequence, sector controllers that need to issue an altitude restriction prior to delivery to an adjacent sector will issue descents accordingly; UAs or any aircraft that are performance outliers can make it difficult for controllers to predict trajectories and separation and can furthermore result in additional workload and coordination if an altitude restriction cannot be met. Expected responses to controller instructions (e.g., “climb and maintain one four thousand”) can also be affected because UAs may not be able to comply in a timely manner due to aircraft performance and other factors. Other consequences include new considerations for wake vortex separation (JO 7110.65, paragraph 5-5-4) and visual/radar detection (see also Sect. 88.3.6 of this chapter).

The absence of onboard pilots not only allows for UAs that have unique physical and performance attributes but also allows for unique operations or missions, such as multiday endurance or high altitude. In these cases, typical manned

aircraft performance may not be achievable without considerable design trade-offs that compromise mission-specific performance. Furthermore, UA missions (e.g., surveillance, inspection) often call for atypical flight profiles such as circles and spirals or unpredictable paths. As with UA performance, UA flight profiles are not necessarily specific to UAS versus manned aircraft, but the missions allowed by remote pilots and unique aircraft performance enable and foster these profiles, possibly to the extent that the aircraft cannot be exclusively contained in restricted airspace. In comparison to typical point-to-point operations, an increase in atypical missions and flight path geometries can be overall disruptive to controllers and other traffic.

88.3.2 Gap: Visual Instructions

UASs currently cannot accept visual instructions from air traffic controllers. An important part of controller communication with manned aircraft is that a pilot can visually acquire an object or feature that is verbally described to them by the controller. For example, JO 7110.65, Chap. 3-1-6, states that controllers describe “vehicles, equipment, or personnel on or near the movement area in a manner which will assist pilots in recognizing them.” More importantly, visual separation can be used to temporarily separate aircraft, in which the controller receives positive acknowledgement that one pilot has the other aircraft in sight (JO 7110.65, Chap. 7-2-1). Although a number of criteria must be met (e.g., other approved separation is assured before and after the application of visual separation; direct communication is maintained with one of the aircraft involved), visual separation operations can be routinely applied to an extended sequence of aircraft, which is particularly helpful to the controller in high-workload situations.

Because UAs do not have a pilot onboard, the operational paradigm of visual instructions for UAS has to be reconsidered, which presents many challenges for UAS integration in the NAS. One solution might be to keep the UA pilot in the loop by providing natural vision “equivalent” information at the control station, such as through UA cameras. However, the use of electronic imaging technology to replace versus complement pilot natural vision has not yet been achieved (see Sect. 88.2.1 in this chapter). Another possible solution is to represent appropriately equipped traffic with symbology on suitable control station displays, as with ADS-B (automatic dependent surveillance-broadcast) applications and CDTIs (cockpit display of traffic information). Current approved ADS-B applications (RTCA DO-317A) designed for use in the visual separation regime have so far used CDTIs only *with* visual acquisition. However, future applications may allow for IFR use without visual acquisition requirements, for specific, limited operations. Such CDTI-based operations would not provide “equivalence” to visual operations, but could offer similar operational advantages through specific procedures that leverage aircraft equipment and pilot qualification. These kinds of CDTI-based operations would also introduce a number of human performance and system safety concerns, in comparison to natural vision. Another potential and longer-term solution for UAS

is for aircraft automation to comply with visual instructions without involving the pilot, using appropriate sensing technology, but this has automation challenges as well (see Sect. 88.2.5 in this chapter). The inability of UAs to “see” like manned aircraft pilots presents many challenges for UAS integration in the NAS, not only for UA/pilot decision-making and control but for air traffic controllers as well.

88.3.3 Gap: Lost-Link Predictability and Procedures

“Lost link,” defined here for brevity to mean the lost control uplink, is one of the key hurdles with UAS integration in the NAS. As mentioned in Sect. 88.2.2.2 of this chapter, there is not a direct analogy to manned aircraft, and the event is unique to UAS. During lost link the aircraft is autonomous, even though it is not by choice of the designer or pilot.

A lost-link aircraft can be a burden to controllers and a safety hazard to the NAS if controllers are not able to 1) issue its instructions or clearances, or 2) readily predict its next steps.

88.3.3.1 Loss of ATC Instructions

First, the inability of ATC to issue instructions is a fundamental risk, even when the lost-link behavior is readily predictable. This is because a controller’s primary responsibility of aircraft separation must be commensurate with the ability to “control” aircraft through instructions to the pilot, who actually controls the aircraft during nominal operation. Controller instructions are not rare events that reflect poorly defined flight plans, but, rather, are ubiquitous in the complex and unpredictable details of controlled airspace and through all flight phases. Therefore, it is not unlikely that a lost-link period will overlap with a controller’s need to issue a clearance amendment, for example, yet such amendments cannot be executed during lost link. A similar situation exists with observed clearance deviations. JO 7110.65, paragraph 2-1-26, states, “When it appears that the actions of a pilot constitute a pilot deviation, notify the pilot, workload permitting.” But “notify the pilot” implies a pilot’s ability to comply with a controller instruction through flight controls. Furthermore, the loss of voice communications often is associated with the loss of UA control so that without alternate means of communication (e.g., ground network), pilot notification and communication might not be possible, and emergency determination as described in JO 7110.65, Chap. 10-1-1, may be left to the judgment of the controller. Any resulting traffic restrictions (JO 7110.65, Chap. 10-4-1) are disruptive to controllers and other aircraft in the area and can pose a significant safety risk to other aircraft as well.

88.3.3.2 Loss of Predictability

A further problematic aspect of lost-link events is that, at some point, the UA automation will typically discontinue the original trajectory and control the aircraft on a new one. This is somewhat analogous to lost communications in manned aircraft, in which controllers must still predict what the aircraft will do (JO 7110.65,

paragraph 10-4-4, notes that “when an IFR aircraft experiences two-way radio communications failure, air traffic control is based on anticipated pilot actions.” See also FAA Aeronautical Information Manual, paragraph 6-4-1). However, lost-link behavior for any particular UA might not be easily predicted because the procedure the UA automation will initiate in that situation could be based on factors such as its specific communication architecture, systems that have failed, operational priorities, and the contingency plans incorporated within the UAS automation. The most intelligent behavior may therefore be too complex to sufficiently precoordinate with ATC. Hence, it may be necessary for NAS-integrated UAS to ensure UA predictability through an understandable set of predefined contingency information that is pre-coordinated with the appropriate ATC facility (current FAA policy requires coordination of lost link route of flight, transponder use, lost link orbit points, communications procedures, and pre-planned flight termination points in the event recovery of the UAS is not feasible). The degree of pre-coordination needed to ensure predictability might decrease as controllers are provided with improved tools to obtain information during lost-link events.

Without a functional pilot in the control loop, the most appropriate UA behavior, even from the ATC perspective, is not likely to be one-size-fits-all, but will be specific to the airspace, traffic, and other factors. On one hand, the UA often needs to maneuver and reestablish the link in order to be controllable, and on the other hand, the UA behavior needs to be appropriate to the airspace, yet predictable. A key challenge for UAS integration in the NAS is to find a solution within that trade-space that yields appropriate, predictable, and ultimately ATC-controllable UA behavior.

88.3.4 Gap: Communications

Voice communications between controllers and manned aircraft pilots yield human-system performance that may be difficult for UAS to meet. Communications with manned aircraft is based in part on equipment standards, but the human element of the overall performance is not well defined. Nevertheless, overall human-system communications performance is based on expectations of latencies and rhythms that can be sensitive to the changes from UAS.

Notionally, UASs introduce at least a third basic node in the communications architecture. The three nodes are ATC, the UA, and the control station. It is assumed here that the control station is very remote from both ATC and the UA, possibly precluding the use of standard aircraft-ATC communications equipment at the control station.

In the simplest case, the three-node network presents two basic options for linking controller with pilot. The first is to use the UA as the middle node and relay. This relay method has the potential benefit of transparency to ATC since human-to-human communications are preserved with no new equipment or procedures. It also preserves “party line” communication, which has well-recognized benefits. However, the relay method adds communications complexity that, compared to standard pilot-controller communication, can lead to higher potential for dropouts

and latency – both of which can introduce step-ons and errors, delay aircraft compliance, and increase controller’s workload. The communications link also is typically affected by the same factors as the control/data link, such that lost link and lost communications can occur simultaneously and exacerbate their individual effects.

Another option for linking controller with pilot is to not involve the UA at all, and instead rely on a separate infrastructure for alternate communications. This may be a more reliable method in the future since it can involve multiple wired and wireless networks and achieve acceptable communications performance that is not as dependent on the UA position or maneuvers. However, it might require new equipment and/or procedures (such as for sector handoffs) for both controllers and pilots and can therefore be considered challenging to the concept of NAS integration. A separate communications infrastructure might also adversely affect airspace awareness if party line information is not maintained, as compared to standard radio communications.

Additional communications options include manned aircraft NextGen technologies such as data communications (“Data Comm”), which use alphanumeric displayed messages instead of voice, using protocols that accommodate asynchronous communication. Although Data Comm offers many advantages over voice, it can inherently introduce time delay (through asynchronous dialog) and the loss of party line awareness, both of which are already attributes of the UA relay architecture. Hence, the suitability of Data Comm for UAS compared to manned aircraft remains to be determined. A challenge for UAS integration in the NAS is to leverage UAS’s unique communications infrastructure to maintain an acceptable level of communications performance, including human performance, in both normal and non-normal conditions.

88.3.5 Gap: Workstation Information

Controllers need to know specific flight data about the traffic that they monitor and control, and UAS may introduce further information requirements. For manned aircraft, controllers may have access to information such as aircraft ID, aircraft type, and certain optional onboard equipment. The available information depends primarily on whether the aircraft has filed an IFR flight plan.

It is unclear what additional information controllers will need for UAS operations in the NAS. On one hand, early standardization discussions emphasized transparency so that controllers would not necessarily know if an aircraft is manned or unmanned. On the other hand, it became clear that there are unique aspects of UAS, and controllers may need certain UAS-specific information to maintain safety, efficiency, and acceptable workload. Example UAS-specific air traffic controller information could include but is not limited to:

- Unique code for identifying aircraft as unmanned
- Indication of when UA loses control link with control station
- Flight-specific procedures for lost control link and lost communications (as discussed in Sect. 88.3.3.2 in this chapter)

Unfortunately, planned display changes for air traffic control of manned aircraft are already presenting information management challenges to designers of controller workstations, as more and more technologies and information are made available. UAS-specific information will need justification to earn its way in a datablock or elsewhere, particularly when considering the addition of new training and procedures onto an existing complex set of tasks.

88.3.6 Gap: Detection of Small-Size UAs

UAs can be much smaller than manned aircraft, which not only leads to aircraft performance differences (see Sect. 88.3.1 in this chapter) but also difficulties in detection by ATC and other traffic. Detection difficulties can be both visual and radar based. Manned aircraft have evolved an approximate lower bound in size and radar cross section. For example, controllers in towers therefore expect to visually detect aircraft a certain distance away through the tower window, and pilots of other aircraft are also expected to visually acquire aircraft through the aircraft windshield and sometimes follow them. They not only *detect* aircraft visually but also estimate other states such as range and speed by means of its subtended angle and movement. Radar-based equipment complements or replaces visual detection and state estimation (or vice versa), but this equipment relies on sufficiently large aircraft cross sections (for primary radar) or transponder equipage (for secondary radar).

When a UA is of sufficiently small size, there is risk that its small cross section can cause visual and radar detection difficulties and other state estimation difficulties. For example, a tower controller that expects to see any aircraft on a runway in a given visibility condition might mistakenly assume that a runway is clear, even when it is actually occupied by an unexpectedly small UA. This may compromise a controller's ability to, for example, visually scan a runway for traffic (JO 7110.65, paragraph 3-1-12). Visual state estimation can also be compromised, such as when a manned trailing aircraft is attempting to maintain visual spacing on a leading UA based on erroneous assumptions of its size (see Sect. 88.3.2 in this chapter). Similar concerns arise also with a pilot's ability to maintain a safe distance and see and avoid other aircraft, per 14 CFR 91.111 and 91.113.

Although special operating limitations will address the smallest sized UAs (FAA Order 1110.150), many UAs outside of that category could still be orders of magnitude smaller than manned aircraft. A key challenge for UAS integration in the NAS is to understand the effects that small-size UAs have on the airspace system, such as human-system performance in aircraft detection and other state estimation.

88.4 Conclusions

The purpose of this chapter was to identify human factors challenges to integrating UASs in the NAS, for both pilots and air traffic controllers. The method for identifying these challenges was primarily based on the differences or "gaps"

between manned aircraft in the NAS today and the unique aspects of UAs. The goal was not to generate a comprehensive list of human factors issues, but to focus on gaps that are traceable to fundamental characteristics of UAs (primarily due to the UA pilot being physically remote from the aircraft) and that are also considered challenging from the perspective of the current NAS and its regulatory and policy framework.

The human factors NAS integration gaps discussed in this chapter were:

Pilot operations at the control station

- Loss of natural sensing
- Control loop degradation – data dropouts
- Control loop degradation – latency
- Link monitoring
- Automation and automated systems
- Information formats

Air traffic control

- Different aircraft performance and missions
- Visual instructions
- Lost-link predictability and procedures
- Communications
- Workstation information
- Detection of small-size UAs

There are many other gaps as well. Some (e.g., “sense and avoid”) were not discussed earlier because their gaps can be comprised of elements from the above list. Some common UAS human factors issues were not addressed here because they are not fundamental gaps, but rather the result of design choices that do not follow aviation standards versus the result of the fundamental attributes of UAS.

Notable human factors gaps worth briefly describing here, but not already covered in this chapter, include:

- Sterility and security of the control station. Manned aircraft cockpits are inherently isolated by means of their location within the aircraft and location of the operating aircraft itself (e.g., airborne). UAS control stations do not have these inherent characteristics, and so new considerations are needed to achieve the levels of security and sterility needed for appropriate UAS flight crew performance.
- Pilot complacency and the loss of shared fate. UAS pilots are physically isolated from the aircraft and its local environment, which can exacerbate pilot complacency, boredom, loss of attentiveness, and fatigue. Furthermore, concerns have been raised about safety because aircraft accidents or flight termination do not pose a direct health risk to the pilot, which can be a perceived safety risk when compared to manned aircraft pilots, who are assumed to use survival instincts when necessary. More research is needed to understand these differences.
- Pilot control handoffs. Crew resource management (CRM) is well established with manned aircraft and more straightforward than with UAS due to the colocation of crew members, all of whom use one physical flight deck interface. However, a UAS flight crew can be physically distributed among multiple control stations, in which aircraft control is handed off during operation. This is a new

aviation paradigm that will likely require new design features and flight crew procedures, with added considerations for air traffic controller handoffs between multiple sectors.

- Pilot training and qualification. FAA Notice 8900.227 includes policy on UAS personnel qualifications, including pilot training. However, ultimately, it is conceivable to require some pilots to be trained under an FAA approved training program to operate UASs in the NAS, depending on the type of operation and regulatory requirements developed. For comparison of required training programs, it may be more appropriate to use training for 14 CFR part 135 pilots instead of the more formal training required by 14 CFR part 121. For both types of training, the baseline definition of training requirements is done by a Flight Standardization Board (FSB) convened and led by the FAA Aircraft Evaluation Group (AEG). The FSB produces an FSB report to specify FAA training, checking, and currency requirements and to assist FAA Principal Operations Inspectors (POI) in the administration of approved training, checking, and currency programs for operators of the specified airplanes. An identical or similar process might be used to determine minimum training requirements for UAS pilots. In addition to the FSB report, FAA Order 8900.1 provides guidance on training program details. These documents can point to a likely structure that may be used for UAS pilot training; however, many details are still to be determined about the specific requirements and process that will be implemented. It should also be noted that all manned aircraft pilots are required to be qualified under 14 CFR Parts 61 and 91 with the minimum number of flight hours and pass the appropriate tests to meet the level of operation for which they are being qualified.
- Flight planning. Flight planning is important for any flight, but when the pilot is onboard the aircraft, many contingency decisions can be deferred until operation. With UAS, the potential for lost link places more demands on flight planning (prior to actual operation) in order to account for the remote location and possible functional absence of humans *during* operation (see Sect. 88.2.5 in this chapter). For example, when a manned aircraft engine fails, the pilot might be able to simply look out the window to determine a suitable emergency landing location. With UAS, such real-time decision-making may be precluded, depending on the sensing, automation, and level of pilot involvement at the control station. Therefore, UAS flight planning might need to be much more detailed and rigorous when compared to manned aircraft flight planning.

The challenges presented in this chapter will collectively require significant time and effort by the UAS and the larger aviation community to overcome. Progress can be facilitated by moving toward UASs that are designed specifically to include integration in the NAS, similar to how manned aircraft are designed. This would include human factors considerations for pilots, controllers, and other actors not discussed in this chapter such as observers, airport personnel, and maintenance personnel.

If the significance of human factors is not understood, the NAS integration gaps identified in this chapter may merely be addressed with operational limitations (such

as line-of-sight control) that fall short of true NAS integration when compared to manned aircraft. Human factors is not only central to the very definition of UAS but is also central to some of the most important challenges facing the integration of UAS in the NAS.

References

- Federal Aviation Administration, Order 9550.8, Human factors policy, Oct 1993
- Federal Aviation Administration, Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap, First Edition-2013
- Federal Aviation Administration, Notice 8900.227, Unmanned Aircraft Systems (UAS) Operational Approval, Jul 2013
- Federal Aviation Administration, Notice 7210.846, Unmanned Aircraft Operations in the National Airspace System (NAS), Jul 2013
- Federal Aviation Administration, Advisory circular 25.773-1, Pilot compartment view design considerations, Jan 1993
- Federal Aviation Administration, Unmanned Aircraft Program office, Interim operational approval guidance 8-01, Unmanned Aircraft Systems Operations in the U.S. National Airspace System, Mar 2008
- Federal Aviation Administration, Advisory circular (AC) 20-167, Airworthiness Approval of Enhanced Vision System, Synthetic Vision System, Combined Vision System, and Enhanced Flight Vision System Equipment, June 2010
- Federal Aviation Administration, Advisory circular (AC) 90-106, Enhanced flight vision systems, June 2010
- Federal Aviation Administration, Advisory circular (AC) 25-11A, Electronic flight deck displays
- Federal Aviation Administration, Advisory circular (AC) 23-8C, Flight test guide for certification of part 23 airplanes, Nov 2011
- Federal Aviation Administration, Advisory circular (AC) 25-7B, Flight test guide for certification of transport category airplanes, Mar 2011
- Federal Aviation Administration, Advisory circular (AC) 23-1523, Minimum flight crew, Jan 2005
- Federal Aviation Administration, Advisory circular (AC) 25.1523-1, Minimum flightcrew, Feb 1993
- Federal Aviation Administration, Advisory circular (AC) 23.1311-1C, Installation of electronic display in part 23 airplanes, Nov 2011
- Federal Aviation Administration, Advisory circular (AC) 20-175 controls for flight deck systems, Dec 2011
- Federal Aviation Administration, Technical standard order, TSO-C147, Traffic advisory system (TAS) airborne equipment, Apr 1998
- Federal Aviation Administration, Technical standard order, TSO-C118, Traffic alert and collision avoidance system (TCAS) airborne equipment, TCAS I, Aug 1988
- Federal Aviation Administration, Technical standard order, TSO-C119c, Traffic alert and collision avoidance system (TCAS) airborne equipment, TCAS II with optional hybrid surveillance, Apr 2009
- Federal Aviation Administration, Technical Standard order, TSO-C195a, Avionics supporting automatic dependent surveillance–broadcast (ADS-B) aircraft surveillance applications (ASA), Feb 2012
- Federal Aviation Administration, Technical standard Order TSO-C63d, Airborne weather radar equipment, Feb 2012
- Federal Aviation Administration, Technical standard order TSO-C105, Optional display equipment for weather and ground mapping radar indicators, June 1984

- Federal Aviation Administration, Technical standard order TSO-C151b, Terrain awareness and warning system, Dec 2002
- Federal Aviation Administration, Air traffic control, Order JO 7110.65U, Feb 2012
- Federal Aviation Administration, Aeronautical Information Manual, Official guide to basic flight information and ATC procedures, Feb 2012
- Federal Aviation Administration, Order 1110.150, Small unmanned aircraft system aviation rulemaking committee, Apr 2008
- Federal Aviation Administration, Order 8900.1, Flight standards information management system (FSIMS), Sept 2007
- International Civil Aviation Organization (ICAO), Doc 8643, Aircraft type designators, 39th edn., Apr 2011
- J. McCarley, C. Wickens, Human factors implications of UAVs in the national airspace. Technical report, AHFD-05-05/FAA-05-01, University of Illinois at Urbana-Champaign, Apr 2005
- RTCA DO-317A, Minimum operational performance standards (MOPS) for aircraft surveillance applications system (ASAS), Dec 2011
- Title 14 of the Code of Federal Regulations, 25.773, Pilot compartment view
- Title 14 of the Code of Federal Regulations, 23.773, Pilot compartment view
- Title 14 of the Code of Federal Regulations, 91.175, Takeoff and landing under IFR
- Title 14 of the Code of Federal Regulations, 23.251, Vibration and buffeting
- Title 14 of the Code of Federal Regulations, 23.1301, Functions and installations
- Title 14 of the Code of Federal Regulations, 2x.1309, Equipment, systems, and installations
- Title 14 of the Code of Federal Regulations, 23.143 and 25.143 general
- Title 14 of the Code of Federal Regulations, 27.143 and 29.143 controllability and maneuverability
- Title 14 of the Code of Federal Regulations, 2x.1523, Minimum flight crew
- Title 14 of the Code of Federal Regulations, 91.111, Operating near other aircraft
- Title 14 of the Code of Federal Regulations, 23.155 Elevator control force in maneuvers
- Title 14 of the Code of Federal Regulations, 23.1321/25.1321/27.1321, Arrangement and visibility
- Title 14 of the Code of Federal Regulations, 23.255, Out of trim characteristics
- Title 14 of the Code of Federal Regulations, 91.113, Right-of-way rules: except water operations
- Title 14 of the Code of Federal Regulations, 25.1302, Installed systems and equipment for use by the flightcrew
- K. Williams, Documentation of sensory information in the operation of unmanned aircraft systems, Federal aviation administration, DOT/FAA/AM-08/23, Oct 2008

Douglas Marshall

Contents

89.1 Introduction 2160

89.2 A Brief History of Regulation 2160

89.3 Are Contracting States Bound by ICAO Definitions of Aircraft and Aeroplanes? 2163

89.4 Where to Find the “Rules of the Road” In Domestic and International Airspace 2164

89.5 Can Unmanned Aircraft Comply with ICAO Rules of the Air?..... 2166

89.6 Understanding the Regulatory Structure: What to Look For? 2166

89.7 What Applies and What Does Not?..... 2168

89.8 Role and Development of Standards..... 2170

89.9 A Suggested Method for Compliance 2171

89.10 Is Full Global Harmonization Realistic and Achievable?..... 2174

References 2175

Abstract

The rapidly evolving technology of unmanned systems has outpaced the ability of aviation regulators and policy makers around the world to develop reasonable design, production, airworthiness, maintenance, licensing, registration, operational, and crewmember certification regulations and standards. Efforts are underway in several countries, as well as internationally, to develop these regulations and harmonize them across borders. The multiple uses of civil UAS technology compel an approach to regulatory oversight that acknowledges that each system has unique capabilities and characteristics, and that the creation of rigid categories for requirements may inhibit development and innovation. On the other hand, the potential for harm is such that regulations and standards are unquestionably needed. For nations with comprehensive and deeply embedded aviation regulations such as the United States, the beginning point for compliance

D. Marshall
Physical Science Laboratory, New Mexico State University, Las Cruces, NM, USA
e-mail: dmmarshall@psl.nmsu.edu; marshall@gra.midco.net

is to understand the regulatory scheme and how the FAA applies the existing rules to UAS operations. This chapter proposes a methodology for analysis of current regulations for applicability to UAS activities. Where the rules are in development, or merely being contemplated, a structure for harmonization with international standards is proposed. The article focuses on the USA as the nation that has the most comprehensive aviation regulations, while also supporting major efforts to promulgate regulations and standards to supplement those regulations with UAS-specific requirements and criteria. To provide global context, the International Civil Aviation Organization's structure and procedures are reviewed in detail, and contrasted with State-sponsored regulatory processes, with an analysis of the applicability of ICAO's standards to UAS operations.

89.1 Introduction

Every civilized nation has some form of rules, regulations, or laws that regulate economic activities and social behavior. The scope and process for the creation of these rules varies widely among nations, depending upon many factors that derive from the form of government that produces those rules. As a society grows more open and more complex, more rules and regulations may become necessary to maintain order and protect people from harm that may be caused by the unrestricted activities of others. Those harms can be physical, health related, economic, environmental, or any number of potentially damaging outcomes from a governmental entity, an organization, or an individual doing or failing to do something that threatens the well-being of others. As the technology of unmanned aerial systems evolves and more potential users seek to deploy them for commercial, scientific or public safety purposes, some governments have attempted to meet the demand and the potential for harm by developing regulations that address the many risk management and policy challenges presented by these devices. Since there is no overarching international law, treaty or set of standards that governs unmanned aerial systems, the developer, manufacturer, and end user must be wary of the potential for inadvertent violations of existing law, or of seeing a formerly legal activity become illegal or proscribed as a result of a change in the rules. Methodologies for understanding the existing rules and participating in the process of developing new or modified rules should be key elements in the business plan for any individual or entity that seeks to participate in the unmanned systems arena.

89.2 A Brief History of Regulation

Regulations, and the supporting standards that help people and organizations comply with them, have been a core element of the legal system in the United States for nearly 150 years. Every Federal regulatory agency promulgates rules that are intended to carry out the agency's Congressional mandates. State and local

regulatory agencies act in much the same way. Congress has neither the time nor the resources to define and monitor every element of the particular industry or activity that it undertakes to regulate through the enactment of federal laws, so the agencies that Congress creates and funds are delegated the legislative powers that Congress has been granted by Article, Sect. 89.1 of the U.S. Constitution. Those delegated powers are implemented through the rule making process. The earliest Federal regulatory agency that still exists is the Office of the Comptroller of the Currency, which was created in 1863 to charter and regulate the nation's banks. The modern era of Federal regulation really began with the creation of the Interstate Commerce Commission (ICC) in 1887, which was directed to protect the public from excessive and discriminatory railroad rates. The regulation was economic in nature, by the setting of rates and by regulating how railroad services were to be provided. The ICC was dissolved in 1995, and its remaining powers were reassigned to the Surface Transportation Board. The ICC's administrative model was that of an independent, bipartisan commission that used an adjudicatory approach to arrive at decisions on contested matters. This structure was adopted by several agencies subsequently created, such as the Federal Trade Commission (FTC), the Water Power Commission (later the Federal Power Commission), and the Federal Radio Commission (which became the Federal Communications Commission). Congress created several other agencies early in the twentieth-century to regulate commercial and financial systems – including the Federal Reserve Board, the Tariff Commission, the Packers and Stockyards Administration, and the Commodities Exchange Authority, all established before 1922. The Food and Drug Administration was created in 1931 to ensure that certain foods and drugs were fit for human consumption.

Many other Federal regulatory agencies were created in the 1930s as part of President Franklin D. Roosevelt's "New Deal" programs. These included the Federal Home Loan Bank Board (1932), the Federal Deposit Insurance Corporation (FDIC) (1933), the Commodity Credit Corporation (1933), the Farm Credit Administration (1933), the Securities and Exchange Commission (SEC) (1934), and the National Labor Relations Board (1935). The jurisdictions of both the Federal Communications Commission (FCC) and the Interstate Commerce Commission were also expanded to regulate other methods of communications (e.g., telephone and telegraph) as well as other transport modes (such as trucking).

The United States' first attempt to regulate commercial aviation arose out of the U.S. Post Office's commencement of airmail operations in 1918. The first published aviation regulations were released in 1926 by the Aeronautics Branch of the Department of Commerce and were known as "Air Commerce Regulations." They consisted of six chapters spanning 45 pages of text, and covered the areas of Licensing, Marking and Operation of Aircraft, Licensing of Pilots and Mechanics, Air Traffic Rules, and rounded out by a miscellaneous section. Aviation is indeed one of the oldest regulated industries or activities in the USA. Presently the Federal Aviation Regulations ("FARs") fill four volumes of the Code of Federal Regulations, consisting of over 460 sections filling more than 3,600 pages, and totaling over 8,000 regulations, counting major subparts and sections.

Outside of the USA, aviation regulations vary from country to country, as can be expected. Some countries have aviation rules and regulations similar to the United States, but none are as comprehensive. In Europe, for example, each individual country ("Member State") has its own set of regulations, developed and enforced by Civil Aviation Authorities ("CAAs") in each country. And there is an additional layer of safety rules that fall under the regulatory and administrative jurisdiction of the European Aviation Safety Agency ("EASA"), an agency of the European Union ("EU"). EASA's role is to provide advice to the European Union for drafting new legislation, implementing and monitoring safety rules, including inspections in the Member States, type-certification of aircraft and components, as well as the approval of organizations involved in the design, manufacture and maintenance of aeronautical products, authorization of third-country (non-EU) operators, safety analysis, and research. The European Organization for the Safety of Air Navigation, known as "EUROCONTROL," is the entity that provides harmonized air navigation services across European skies and is an organization separate from EASA. This structure is to be contrasted with the United States', where all aviation regulations and air navigations services fall under the umbrella of one agency, the Federal Aviation Administration ("FAA"), although investigation of accidents is conducted by the National Transportation Safety Board ("NTSB"), which also provides an administrative, quasi-judicial review of FAA enforcement actions.

Virtually every member of the United Nations (currently numbering 191 countries) has its own version of domestic aviation regulations, which for the most part are patterned after the 1944 Chicago Convention on International Civil Aviation, which established the International Civil Aviation Organization ("ICAO") as a means to secure international cooperation and the highest possible degree of uniformity in regulations and standards, procedures, and organization regarding civil aviation matters. The Chicago Conference laid the foundation for a set of rules and regulations regarding air navigation as a whole, which was intended to enhance safety in flying and set the groundwork for the application of a common air navigation system throughout the world.

ICAO works in close cooperation with other members of the United Nations family such as the World Meteorological Organization, the International Telecommunication Union, the Universal Postal Union, the World Health Organization, the International Maritime Organization, and the Arctic Council. Aviation-related non-governmental organizations ("NGOs") also participating in ICAO's efforts include the International Air Transport Association, the Airports Council International, the International Federation of Air Line Pilots' Associations, and the International Council of Aircraft Owner and Pilot Associations.

A comprehensive analysis of the aviation regulatory schemes in each and every country that publishes some form of regulation is beyond the scope of this chapter. However, since ICAO Member States contract to follow ICAO's rules and supplementary material and to publish for all to see any exceptions taken to any ICAO Article, Annex, Regional Supplementary Procedure ("SUPPS") or Procedure for Air Navigations Services ("PANS"), a beginning point must be the aforementioned ICAO rules and procedures themselves.

A discussion of methods to ensure compliance with applicable aviation regulations by those involved with any aspect of unmanned aerial systems necessarily begins with determining what an “aircraft” is. ICAO’s definitions state that an aircraft is “(a)ny machine that can derive support in the atmosphere from the reactions of the air other than the reactions of the air against the earth’s surface [5]. An “aeroplane” is “a power-driven heavier-than-air aircraft, deriving its lift in flight chiefly from aerodynamic reactions to surfaces which remain fixed under given conditions of flight” [16]. A radio-controlled (“RC”) model aircraft (fixed-wing or helicopter) purchased off-the-shelf from the local hobby shop or airport terminal would be included under either of these definitions. There is no definition of an unmanned aircraft in the Convention or the Annexes to the Convention. However, Article 8 of the Convention states that:

No aircraft capable of being flown without a pilot shall be flown without a pilot over the territory of a contracting state without special authorization by that State and in accordance with the terms of such authorization. Each contracting state undertakes to insure that the flight of such aircraft without a pilot in regions open to civil aircraft shall be so controlled as to obviate danger to civil aircraft [11].

This provision only applies to pilotless [22] aircraft being flown over the territory of a contracting state without the State’s permission, and each contracting state agrees that pilotless aircraft will not be flown in a manner that endangers civil aircraft.

What type of aircraft does the Convention intend to regulate? Article 3 of the Convention states that:

- (a) This Convention shall be applicable only to civil aircraft, and shall not be applicable to state aircraft.
- (b) Aircraft used in military, customs and police services shall be deemed to be state aircraft.
- (c) No state aircraft of a contracting state shall fly over the territory of another State or land thereon without authorization by special agreement or otherwise, and in accordance with the terms thereof [12].

It is thus clear that the ICAO definitions of aircraft that are subject to its Articles, Annexes, and Supplementary Agreements include any man-made contrivance that is capable of sustained flight above the immediate surface level of the Earth. An “aeroplane” is defined as a powered aircraft. No minimum size is described, so even the previously mentioned radio-controlled model aircraft could be covered under a literal definition. Thus, in the ICAO regulatory scheme, no distinction is made between manned and unmanned aircraft.

89.3 Are Contracting States Bound by ICAO Definitions of Aircraft and Aeroplanes?

Another way of stating this question is to inquire if contracting states may create their own definitions of airplanes or aircraft and, if so, whether they can enforce those definitions and any rules or regulations that may apply on operations in ICAO-defined international airspace in which the contracting state provides air navigation

services. Similarly, may a State impose its own regulations on aviation activities engaged in by its own citizens outside of their domestic airspace jurisdiction? In answer to those questions, the ICAO Council adopted a resolution in 1948 alerting contracting states to the desirability of their adopting, in their own national regulations, as far as practicable, the precise language of those ICAO Standards that are of a regulatory character and also of indicating departures from the Standards, including any additional national regulations that were important for the safety or regularity of air navigation. The resolution also noted that, wherever possible, the provisions of Annex 2 were written in such a way that they could be incorporated, without major textual changes, into national legislation [23].

Later, in 1972, when adopting Amendment 14 to Annex 2 relating to authority over aircraft operating over the high seas, the Council declared that the Amendment was intended solely to improve safety of flight and to ensure adequate provision of air traffic services over the high seas. The Council emphasized that the amendment was in no way intended to impact the legal jurisdiction of States of Registry over their aircraft or the responsibility of contracting states under Article 12 of the Convention for enforcing the Rules of the Air [1]. Thus, contracting states are free to create their own definitions and categories of aircraft, and to the extent that those States retain jurisdiction over aircraft registered in their State even if they are operating in international airspace, the States may apply their own laws and regulations.

The issue then becomes whether the aviation laws, rules, and regulations of a contracting state apply to operations in international airspace for which the contracting state provides flight information or air traffic control services. And for operators of unmanned systems, the challenge becomes where to look first for guidance on the rules applying to those systems, that is, locally or globally?

89.4 Where to Find the “Rules of the Road” In Domestic and International Airspace

The Articles in chapter of the Chicago Convention (similar to the Articles in the U.S. Constitution) describe the framework and establish the parameters for the regulatory scheme. Article 1 declares that “*The Contracting States recognize that every State has a complete and exclusive sovereignty over the airspace above its territory.*” This is important to note, as the rules described in the Convention do not apply to the airspace in or over a contracting state’s territory. Article 2 goes on to provide that “*For the purposes of this Convention the territory of a State shall be deemed to be the land areas and territorial waters adjacent thereto under the sovereignty, suzerainty, protection or mandate of such State.*” All other airspace not defined as falling within the sovereign protection of a state (and not necessarily a contracting state) is common, or international airspace.

The Forward to Annex 2 to the Convention states that the Standards in the document, together with the Standards and Recommended Practices of Annex 11, govern the application of the Procedures for Air Navigation Services – Air Traffic Management (PANS-ATM, Doc 4444) and the Regional Supplementary

Procedures – Rules of the Air, and Air Traffic Services, which are to be found in ICAO Doc 7030 [13].

Chapter 2 of Annex 2 defines the territorial application of the Rules of the Air as follows:

- 2.1.1 The rules of the air shall apply to aircraft bearing the nationality and registration marks of a contracting state, wherever they may be, to the extent that they do not conflict with the rules published by the State having jurisdiction over the territory overflown.

Note. – The Council of the International Civil Aviation Organization resolved, in adopting Annex 2 in April 1948 and Amendment 1 to the said Annex in November 1951, that the Annex constitutes Rules relating to the flight and manoeuvre of aircraft within the meaning of Article 12 of the Convention. Over the high seas, therefore, these rules apply without exception.

- 2.1.2 If, and so long as, a contracting state has not notified the International Civil Aviation Organization to the contrary, it shall be deemed, as regards aircraft of its registration, to have agreed as follows: For purposes of flight over those parts of the high seas where a contracting state has accepted, pursuant to a regional air navigation agreement, the responsibility of providing air traffic services, the “appropriate ATS authority” referred to in this Annex is the relevant authority designated by the State responsible for providing those services.

Note. – The phrase “regional air navigation agreement” refers to an agreement approved by the Council of ICAO normally on the advice of a Regional Air Navigation Meeting [10].

In summary, the “Rules of the Air” as defined by ICAO, which consist of general rules, visual flight rules and instrument flight rules, apply to all aircraft bearing registration marks of a contracting state, regardless of where the aircraft is flying, and apply without exception over the high seas, and over national territories to the extent that they do not conflict with the rules of the State being overflown. *The pilot-in-command of an aircraft is responsible for compliance with the rules of the air.* Regardless of the type of flight plan, the pilots are responsible for avoiding collisions when in visual flight conditions, in accordance with the principle of see-and-avoid. Flights operating under instrument flight rules are either kept separated by air traffic control units or provided with collision hazard information by the appropriate air traffic service (ATS) authority.

The world’s airspace is divided into a series of contiguous flight information regions (FIRs) within which air traffic services are provided. In some cases, the flight information regions cover large oceanic areas with relatively low air traffic density, within which only flight information and alerting services are provided. In other flight information regions, large portions of the airspace are controlled airspace within which air traffic control service is provided in addition to flight information and alerting services. Flight information service is provided to aircraft operating in controlled airspace and to others known to the air traffic services units. The prime objective of air traffic services, as defined in Annex 11, is to prevent collisions between aircraft. This Annex also describes ways to expedite and maintain an orderly flow of air traffic and to provide advice and information

for the safe and efficient conduct of flights and alerting service for aircraft in distress. To meet these objectives, ICAO provisions call for the establishment of flight information centers and air traffic control units [20].

Most of the airspace in Oceanic FIRs/CTAs (Control Areas) is high seas airspace within which the ICAO Council has resolved that rules relating to flight and operations of aircraft apply without exception. The majority of the airspace is also controlled airspace, and instrument flight rules (“IFR”) apply to all flights in oceanic airspace when at or above FL060 (flight level 6,000 ft) or 2,000 ft. (600 m) above ground level (AGL), whichever is higher, even when not operating in instrument meteorological conditions (“IMC”).

89.5 Can Unmanned Aircraft Comply with ICAO Rules of the Air?

Unmanned aircraft must be able to comply with the rules of the air set forth in Annex 2 to the Convention before they can be operated in international airspace. ICAO Annex 2 requires that those rules apply to aircraft bearing the nationality and registration marks of a contracting state. What standards then apply if the contracting state that provides flight information, alert, or air traffic control services in the international airspace sector of a Flight Information Region has no specific rules or regulations that address the unique characteristics of unmanned aircraft?

Regardless of whether the flight in international airspace is being conducted under visual or instrument flight plans, the pilot in command is responsible for avoiding collisions when in visual flight conditions, in accordance with the principle of see-and-avoid [21]. Flights operating under instrument flight rules are either kept separated by air traffic control units or provided with collision hazard information. An aircraft shall not be operated in a negligent or reckless manner so as to endanger life or property of others [2]. An aircraft shall not be operated in such proximity to other aircraft as to create a collision hazard [3]. The implication of a literal reading of these rules is that, unless a properly certificated onboard see-and-avoid system is present and operating on the UAS, the aircraft probably cannot comply with the Annex 2 requirements. That may be one of the most challenging ICAO standards to meet, but there are others dealing with airworthiness, navigation equipment, and reliability that would prove equally challenging unless the operator was to secure special permission from the controlling CAA to occupy the airspace.

89.6 Understanding the Regulatory Structure: What to Look For?

In the USA, the Federal Aviation Regulations (FARs) contain two sections that address the basic see-and-avoid obligation:

§ 91.111 Operating near other aircraft.

- (a) No person may operate an aircraft so close to another aircraft as to create a collision hazard.
- (b) No person may operate an aircraft in formation flight except by arrangement with the pilot in command of each aircraft in the formation.
- (c) No person may operate an aircraft, carrying passengers for hire, in formation flight [6].

§ 91.113 Right-of-way rules: Except water operations.

- (a) *Inapplicability*. This section does not apply to the operation of an aircraft on water.
- (b) *General*. When weather conditions permit, regardless of whether an operation is conducted under instrument flight rules or visual flight rules, vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft. When a rule of this section gives another aircraft the right-of-way, the pilot shall give way to that aircraft and may not pass over, under, or ahead of it unless well clear.
- (c) *In distress*. An aircraft in distress has the right-of-way over all other air traffic.
- (d) *Converging*. When aircraft of the same category are converging at approximately the same altitude (except head-on, or nearly so), the aircraft to the other's right has the right-of-way. If the aircraft are of different categories –
 - (1) A balloon has the right-of-way over any other category of aircraft;
 - (2) A glider has the right-of-way over an airship, powered parachute, weight-shift-control aircraft, airplane, or rotorcraft.
 - (3) An airship has the right-of-way over a powered parachute, weight-shift-control aircraft, airplane, or rotorcraft.

However, an aircraft towing or refueling other aircraft has the right-of-way over all other engine-driven aircraft.

- (e) *Approaching head-on*. When aircraft are approaching each other head-on, or nearly so, each pilot of each aircraft shall alter course to the right.
- (f) *Overtaking*. Each aircraft that is being overtaken has the right-of-way and each pilot of an overtaking aircraft shall alter course to the right to pass well clear.
- (g) *Landing*. Aircraft, while on final approach to land or while landing, have the right-of-way over other aircraft in flight or operating on the surface, except that they shall not take advantage of this rule to force an aircraft off the runway surface which has already landed and is attempting to make way for an aircraft on final approach. When two or more aircraft are approaching an airport for the purpose of landing, the aircraft at the lower altitude has the right-of-way, but it shall not take advantage of this rule to cut in front of another which is on final approach to land or to overtake that aircraft [7].

It should be noted that Title 14 CFR, Part 91.113 excludes operations over water, and international airspace, by definition, is over the high seas, or over water, so the section would not apply.

Part 91.1(b) requires that civil aircraft must comply with ICAO Annex 2 when operating over the high seas [8]. Annex 2 requires that "Aircraft shall be equipped with suitable instruments and with navigation equipment appropriate to the route being flown." In addition, Annex 6, Part II, requires that an aircraft operated in international airspace have operating navigation equipment that will enable it to

proceed in accordance with the filed flight plan and with any requirements imposed by air traffic services [4].

Accordingly, any flight operated in international oceanic airspace on an IFR flight plan, a VFR controlled flight plan, or at night, if continued beyond the published range of normal airways navigation facilities (NDB, VOR/DME) is considered to be a long-range Class II navigation operation. Long-range Class II navigation in ICAO Controlled Airspace requires the aircraft to be navigated within the degree of accuracy that is required for air traffic control (the aircraft must follow the centerline of the assigned route, maintain assigned altitude and the speed filed or assigned).

All requirements of Annex 2 (as supplemented by Regional Supplementary Procedures, Document 7030 and Annex 6) are incorporated in 14 CFR 91.1 for those aircraft operating under U.S. civil certifications in international oceanic airspace [15]. In other words, even if a U.S.-registered aircraft leaves U.S. territorial airspace and enters international airspace, it must still be operated under the applicable Federal Aviation Regulations and related orders.

ICAO publishes the SUPPS, which constitute the procedural portion of the Air Navigation Plans developed by Regional Air Navigation (“RAN”) meetings to meet the needs of specific areas that are not covered in the worldwide provisions. The SUPPS complement the statement of requirements for facilities and services contained in the Air Navigation Plan publications. Procedures that apply worldwide are included either in the Annexes to the Convention on International Civil Aviation as Standards or Recommended Practices (“SARPs”), or in the published PANS.

The prospective UAS operator is cautioned that the SUPPS do not have the same status as SARPS. The PANS are recommended to contracting states for worldwide use, while the SUPPS are recommended to contracting states for application in the groups of flight information regions to which they are relevant. PANS originally were developed from common recommendations of regional meetings and were given worldwide application by the ICAO Council after action thereon by ICAO Divisions. Subsequently, there has been a gradual evolution of procedures from the regional to the worldwide category as ICAO Divisions have adapted regionally developed procedures to broader worldwide requirements. Concurrently, some of the worldwide procedures have been found suitable for classification as SARPS and therefore are gradually being incorporated into the Annexes to the Convention [18].

89.7 What Applies and What Does Not?

In the international arena, contracting states that provide air traffic control or information services in international airspace Flight Information Regions can, through the mechanism of supplementary agreements, establish additional rules or procedures for aircraft entering and transiting the airspace in which the contracting states have agreed to provide services. The answer to the question of whether those States can impose their own domestic aviation regulations on aircraft and pilots operating in those FIRs depends upon whether those local regulations conflict with ICAO’s Rules of the Air and other Annexes to the Convention. Examples of

potentially inconsistent application or interpretation of ICAO rules can be found in Canada, Denmark, and Iceland, three nations that control or provide services in North Atlantic, North American, and Arctic airspace. All three States require that pilots and aircraft be IFR (instrument flight rules) rated for trans-oceanic flight, regardless of the altitude to be flown, making no distinction between higher-flying commercial airliners and lower-flying general aviation aircraft. However, other North Atlantic States allow Visual Flight Rules (“VFR”) flight at or below FL055 (5,500 ft above the surface), which means that no services such as navigation vectors or separation between aircraft are provided [14].

The “see-and-avoid” requirement that is both implied and explicitly stated in the Annexes to the Convention presents a challenge and possibly a threshold barrier to anyone wishing to operate unmanned aircraft in international airspace. A remotely piloted aircraft by definition has no human onboard to provide the see-and-void capability that a pilot brings to manned aviation, that is, the ability and the legal requirement to look for and see another aircraft, mentally process the information, and then initiate the necessary steps to avoid a collision. Although many unmanned aircraft are equipped with a camera or visualization device, the equipment is generally used for surveillance or observing whatever the aircraft has been deployed to observe. Others carry remote sensing instruments that may not include a camera and are navigated through the air semi-autonomously on preprogrammed flight plans that rely upon Global Positioning System technology and other sensors that monitor the aerodynamic performance and system health of the aircraft. Remotely situated pilots operating a combination of computerized navigation systems, synthetic vision, and onboard forward-looking cameras operate a variety of military grade UASs, but typically not in nonsegregated civil airspace (there may be exceptions in some European countries). To date, an airborne “see-and-avoid” system that has been certified by any civil aviation authority as being capable of replacing the ability of a human pilot onboard the aircraft to provide the see-and-avoid capability that is required for flight in international airspace has yet to emerge.

As discussed above, contracting states retain the right to publish exceptions to ICAO standards, recommended practices, and procedures as set forth in the Annexes and Supplementary documents [9]. This does not mean that ICAO’s regulations do not apply without exception to international airspace, but rather that they apply in territorial airspace only to the extent that they do not conflict with the regulations of the contracting state. Each contracting state thus retains sovereignty over its own airspace and aviation regulations. In the USA, the FAA publishes and keeps current an extensive list of exceptions to ICAO’s Annexes. Currently no exceptions published by the FAA address the operation of unmanned aircraft in international airspace or airspace in which the USA provides air traffic services. The Regional Supplementary Procedures document published by ICAO sets forth all procedures that have been developed by each contracting state for the Flight Information Regions or Control Areas for which their Air Traffic Service units provide service. As of this writing, none of the regional agreements address flights of unmanned aircraft in their control or information areas.

As ICAO is the standardizing body for international air transportation, its primary goal is to develop the core principles and safe procedures of air navigation and to promote continued cooperation among nations to further evolve those standards as technology changes and improves. The top-level priority is to ensure the safe and orderly growth of civil aviation worldwide, and that now includes unmanned systems.

Ongoing efforts in Europe and several other ICAO states (United States, Canada, Australia, Great Britain, Norway, Greenland/Denmark, and Sweden, to name a few) have focused on developing specific UAS regulations, and in some cases to harmonize them with the ICAO UAS Study Group's recommendations. ICAO released UAS Circular 328 in March of 2011; a 46-page "must read" document that is ICAO's first step towards harmonizing UAS regulations and standards throughout the world. As stated in the forward, "(T)he goal of ICAO in addressing unmanned aviation is to provide the fundamental international regulatory framework through Standards and Recommended Practices, with supporting Procedures for Air Navigation Services and guidance material, to underpin routine operation of UAS throughout the world in a safe, harmonized and seamless manner comparable to that of manned operations. This document is the first step in reaching that goal" [17].

Proposed amendments to Annex 2 (Rules of the Air) and Annex 7 (International Standards Aircraft Nationality And Registration Marks) addressing some UAS-related issues are in circulation, but nothing has been approved, so the diligent operator will want to continue to monitor the ICAO effort, as the outcome will impact development of UAS regulations in the contracting states.

89.8 Role and Development of Standards

As ICAO is a standards making organization, so is the FAA, but in a different sense. ICAO publishes its rules for air navigation in international airspace, and all contracting states agree to abide by them. The contracting states participate in the standards development process and have an equal say in the final product. In the USA, the FAA makes rules (FARs) through a complex and time consuming procedure that allows for broad participation by all citizens, and is guided by the Administrative Procedures Act [24]. To support the regulations, the FAA also engages the services of groups of experts, working through the protocols of organizations such as ASTM, RTCA, and SAE, to create standards that describe methods to comply with the requirements of particular provisions of the FARs. These standards are then published as Advisory Circulars ("ACs"). The ACs usually have some language to the effect that the circular is not mandatory and is not a regulation, but merely describes an acceptable means, but not the only means, of compliance with the subject regulation. It may also state that if the described means of compliance is utilized, it must be followed entirely.

Whether developed at a national level, as in the USA, or internationally, the standards processes are typically consensus based, relying upon the expertise of people who are engaged in the particular field or industry and who have special

knowledge about the subject that they are willing to share. The published standards are especially useful to the proponent or applicant for permissions to operate an unmanned system in nonsegregated airspace, as a tool to convince the controlling civil aviation authority that they can safely operate their system in a manner that does not interfere with navigation or present an unreasonable risk of harm to persons or property in the air or on the ground.

The FAA's general guidelines for UAS standards development are focused on the agency's safety assessment of the potential risks presented by the introduction of unmanned aircraft into the National Airspace System ("NAS"). Standard bodies are to associate standards with hazard causal events (after a hazard has been identified, assess what causes that hazard). For example, a hazard might be identified as a loss of control of the aircraft, causing it to divert from its intended flight path and "fly away" to an unpredictable termination of flight. The causes of such an event could be many; among them a failure of the power source that drives the telemetry or data link that connects the aircraft with the ground control station. That failure in turn could be due to the exhaustion of battery power, a structural battery failure, or a wiring disconnect between the battery and the onboard navigation package, and so on. The regulatory agency or body may impose a general requirement that such an event or series of events be "highly improbable" or "extremely unlikely." Or the requirement could be more specific, such as that the battery must have a life cycle of so many hours of use. If the requirement is general, a supporting standard could be that the data link architecture be tested to achieve some mathematical equation or number that is equivalent to "highly improbable." If the requirement is more specific, the standard might suggest a method to test the life cycle of the battery or put it under some form of stress environment until it fails.

A critical element of a useful standard is that it must be understood. If standards are too abstract, or cannot be tested using conventional scientific methods, there is no way to demonstrate that "acceptable" has been achieved. Thus, for each standard, a number of questions must be answered (a "standard for a standard," so to speak). First, what does the standard aim to achieve? Is it a general standard, or specific to a particular hazard? Is it directed towards design, production, operations, or some other requirement? Second, how is the standard relevant to UAS safety? If the standard is not safety related, but only addresses commercial viability, it may not be an appropriate aviation standard. Third, which hazard causal events are addressed? Fourth, is the standard clear? Can it be reasonably achieved? Is the standard complete? And lastly, does the standard have sufficient content and structure to be usable? The bottom line is that the system or component must be both safe and reliable, and the standards should be constructed to assist the public in achieving both requirements.

89.9 A Suggested Method for Compliance

For the uninitiated or unwary, aviation rules, regulations, laws, and standards present a minefield of possible violations or dead-end paths.

To determine the potential applicability of current Federal Aviation Regulations to all aspects of unmanned aircraft systems, operations and manufacturing, a top-level review of each section of Title 14 of the CFRs was conducted to determine which, if any, of the FARs could be directly applied to UAS. The review further separated these regulatory sections and subsections into four categories: *Clearly Applies*, *May Apply by Interpretation*, *Does Not Apply*, and *Could Apply with Revision*. The total number of sections, subsections, and parts that were so categorized exceeded 8,400.

Follow-on studies of Parts 43, 61, and 91 of the FARs developed and further explored the rationale behind each designation, taking into consideration the intent of the entire Part, so far as the intent is known from Federal Register publications, the Preambles to the published Parts, and other historical or archival documents. The analysis approached the applicability of each section of the relative Part to UAS operations and airworthiness requirements from the perspective of the original intent and safety implications of the relevant sections and endeavored to clarify technical issues raised by the application of known UAS technology to existing regulations.

For example, the 132 subsections of Title 14 CFR Part 61 (Certification of Pilots, Flight Instructors and Ground Instructors) each have multiple subsections, so to the extent that a “scientific method” can be applied to the analysis that underlies this study, each major section was examined holistically, acknowledging that the system as a whole determines in an important way how the parts should be interpreted (“The whole is more than the sum of its parts” – with apologies to Aristotle). However, “reductionism” (holism’s opposite) in science holds that a complex system can be explained by reduction to its fundamental parts. The approach of the study was to assess each major section initially in the context of the overarching purpose of the CFRs (aviation safety), to look behind the section by examining the intent of the rule makers in adopting the language that they did (if evidence of that intent is available), and then to reduce the section and its subsections in the manner that might be employed by a court in interpreting the language of the regulation. Although it would be extremely time consuming and potentially confusing to attempt to identify every possible or reasonably foreseeable scenario that could invoke an interpretation of the regulation, an attempt was made to analyze each section from the perspective of known and contemplated unmanned aircraft applications, drawing upon the materials presented at UAS-focused conferences, symposiums, workshops, and committee meetings as well the multitude of white papers, treatises, thesis, promotional materials, and briefings that are commonly available on public domain websites.

A comprehensive and sustainable analysis of a complex body of regulations necessarily requires much more than just looking at the language in each regulation. In the U.S. system of aviation regulations, some sections cross-reference to other sections or subsections within the Part or to other parts of 14 CFR. For example, Subpart K, Parts 61.401–431 refers to Flight Instructors with a Sport Pilot Rating, but the Subpart does not define a “Sport Pilot” or a “Light Sport Aircraft.” To determine whether any or all of this Subpart applies to unmanned aviation, one must refer to a number of other Subparts to obtain the necessary definitions, but must also

look at the entire history of Light Sport Aircraft regulation to determine relevance to the issues proposed in the study.

Other sections and Subparts require extensive historical analysis that can only be developed through archive searches or anecdotal sources. The overarching consideration is how specific regulatory language might be interpreted by the FAA's (or any CAA's) enforcement organization, the user community that desires UAS access to a nation's airspace and the adjudicators who may be called upon to arbitrate or decide disputes arising from those interpretations. This approach, to be comprehensive and useful, requires the evaluator to wear three hats: The FAA/CAA enforcement staff and legal counsel (if any), the pilot, operator or instructor who may be most directly affected by an interpretation of a rule or regulation, and the Administrative Law Judge, NTSB Member or appellate court justice (in the American system) who may ultimately be called upon to decide the issue in an adversarial proceeding.

The sources for interpretation of U.S. Federal Aviation Regulations can be identified in 11 categories:

- The language of the regulation itself
- Explanations of the meaning of the regulation that may be contained within the regulation's language
- Guidance offered in the Preambles, if any
- Cross-references to other regulations and their respective guidance
- Advisory Circulars that represent the FAA's interpretation of regulatory language, often with reference to a particular incident or situation, and contain explanations of regulations, other guidance material, best practices, or information useful to the aviation community (as noted above, Advisory Circulars are not regulatory, do not create or change a regulatory requirement, and are not binding on the public)
- FAA Notices and Orders (published)
- FAA Policy Statements and Guidance Documents (published)
- Legal interpretations contained in NTSB case reports or in published opinions from the courts of general jurisdiction
- Language in the Aeronautical Information Manual ("AIM")
- Other agency interpretations (often published, but not binding on the public)
- FAA Chief Counsel opinion letters

The available resources in other countries may not be as extensive as in the USA, and the best resource for guidance for UAS activities will most likely be the language of the regulation itself, perhaps informally supplemented by a conversation with a knowledgeable representative of the CAA.

The requirement for this type of regulation study is to provide an analytical tool, employing system safety analysis methodology, as well as sound legal interpretive principles, that will enable the interested party to identify and appreciate the gaps, overlaps, consistencies, and inconsistencies in the current regulatory scheme that serves as the framework for oversight of involved in unmanned aviation. Evaluating a body of regulations that is comprised of over 8,000 sections that is supplemented by thousands of pages of Orders, Advisor Circulars, guidance documents, and is

continually in the process of revision, is a daunting task. And that only describes the challenge in the United States. For a UAS manufacturer, developer, or operator to compete their product or service in a global market, the challenge may be to perform or engage the performance of similar top-down studies in every potential arena of use. Because there is little or no global harmonization of UAS rules other than those that ICAO is developing for service providers in international airspace, with potential application or adoption in territorial airspace, a methodical approach facilitated by a strong analytical device becomes the tool of choice for the diligent UAS proponent.

89.10 Is Full Global Harmonization Realistic and Achievable?

A wildly optimistic answer would of course be “Yes.” There are no fewer than a dozen UAS harmonization and standards efforts underway as this book goes to press, both on an international level and localized within individual countries. In the USA, the President signed into law legislation in February of 2012 that specifically mentions unmanned aircraft systems and directs the FAA to perform certain tasks and achieve specific benchmarks (including full integration of UAS into the national airspace) before September 30, 2015 [19]. In 2008 the FAA chartered a sUAS (small UAS) Aviation Rulemaking Committee (“ARC”) to recommend a set of rules to govern smaller categories of unmanned systems. Simultaneously, a safety risk assessment was performed, and the ASTM organization was later engaged to develop a set of standards that would provide guidance on means of compliance with the new rules when they are published in the NPRM (Notice of Proposed Rulemaking), which is a legal requirement that the FAA must meet before the rules can be finalized.

Elsewhere, the work goes on in many countries that have responded to the demands of the commercial and scientific UAS community to promulgate rules and standards that will allow them to carry out a wide variety of useful purposes for UAS, such as climate and atmospheric research, marine and terrestrial mammal observation, fisheries monitoring, migratory bird surveys, sea ice analysis, pipeline and power line inspection, agricultural remote sensing, wildlife and forest monitoring, law enforcement, etc., and the list goes on. The work undertaken by the ICAO UAS Study Group, as seen in ICAO Document 328, may provide a starting point for nations that have not begun to address the UAS issue, a global context for those that have so begun, and a roadmap for compliance for nations such as the United States that have progressed the farthest in developing a comprehensive framework for regulatory oversight over this rapidly evolving technology.

Conclusion

For both the novice and the seasoned professional, the task of finding, understanding, and incorporating into UAS development and operations all of the applicable rules, regulations, standards, and guidance can be somewhat

overwhelming. Some aviation regulations are voluminous (the United States), some short and sweet (e.g., Iceland), and others quite incomprehensible (Russian Federation, among others). The methodology suggested in this chapter works especially well for complex regulatory environments, but can be useful as well for those that are less cumbersome. The important element to remember is that all rules and regulations are subject to interpretation, and the person or entity that chooses to engage in UAS activities without a thorough understanding of the rules and requirements does so at their peril. In the USA, the peril can mean significant monetary penalties, revocation of certificates and licenses, and even jail time. One bad event or accident involving a UAS operating outside of agreed upon rules and standards could set the industry back several years, in the opinion of many experts in the field. Flying first and begging forgiveness later if something goes wrong may work in limited circumstances, but as a long-term business practice, it is a risk that may not be worth taking.

Harmonization on a global scale is indeed achievable, and the processes tend to be relatively transparent and friendly to all comers who have an interest in the issues. The best way to ensure that your voice is heard is to participate in the rulemaking at some level, and to bring your experiences and knowledge to the table for the benefit of all.

References

1. Amendment 14 to Annex 2 of the Convention on International Civil Aviation
2. Annex 2 to the Convention, Chapter 3, section 3.1.1
3. Annex 2 to the Convention, Chapter 3, section 3.2.1
4. Annex 6 to the Convention, Part III
5. Annex 2 to the Convention, International Standards, Chapter 1, Definitions
6. 14 CFR Part 91.111
7. 14 CFR Part 91.113
8. 14 CFR Part 91.1 (a) Except as provided in paragraphs (b) and (c) of this section and §§91.701 and 91.703, this part prescribes rules governing the operation of aircraft (other than moored balloons, kites, unmanned rockets, and unmanned free balloons, which are governed by part 101 of this chapter, and ultralight vehicles operated in accordance with part 103 of this chapter) within the United States, including the waters within 3 nautical miles of the U.S. coast; (b) Each person operating an aircraft in the airspace overlying the waters between 3 and 12 nautical miles from the coast of the United States must comply with §§91.1 through 91.21; §§91.101 through 91.143; §§91.151 through 91.159; §§91.167 through 91.193; §91.203; §91.205; §§91.209 through 91.217; §91.221; §§91.303 through 91.319; §§91.323 through 91.327; §91.605; §91.609; §§91.703 through 91.715; and §91.903; (c) This part applies to each person onboard an aircraft being operated under this part, unless otherwise specified
9. Chapter 1, Articles 1 and 2 of the Convention, *Supra*.
10. Chapter 2, Annex 2 to the Convention on International Civil Aviation
11. Convention on International Civil Aviation, Article 8
12. Convention on International Civil Aviation, Article 3
13. Forward to Annex 2 to the Convention on International Civil Aviation
14. *Ibid*.
15. *Ibid.*; 14 CFR Part 91.1
16. *Id*.

17. ICAO Circular 328 AN/190 “Unmanned Aircraft Systems Circular” ISBN 978-92-9231-751-5
18. ICAO Regional Supplementary Procedures (Doc 7030)
19. Public Law 112-95, The “FAA Modernization and Reform Act of 2012”
20. The Convention on International Civil Aviation, Annexes 1 to 18 (ICAO Annexes Booklet)
21. “The pilot-in-command of an aircraft shall, whether manipulating the controls or not, be responsible for the operation of the aircraft in accordance with the rules of the air, except that the pilot-in-command may depart from these rules in circumstances that render such departure absolutely necessary in the interests of safety.” Annex 2 to the Convention, Chapter 3, section 3.2
22. The term “pilotless” is a misnomer, in that even RC aircraft are “piloted,” just not by a person on board the aircraft
23. <http://www.icao.int/>
24. 5 USC Chapter 5

Certification Strategy for Small Unmanned Aircraft Performing Nomadic Missions in the U.S. National Airspace System

90

Maciej Stachura, Jack Elston, Brian Argrow, Eric W. Frew, and
Cory Dixon

Contents

90.1 Introduction 2178

90.2 Motivation and Application Strategy 2180

90.3 COA Application..... 2182

 90.3.1 COA Area Selection 2182

 90.3.2 Airworthiness Statement 2185

 90.3.3 Lost Communications and Emergency Procedures 2187

90.4 Using COAs and FAA Interaction..... 2191

 90.4.1 COA Provisions 2191

 90.4.2 Activating COA Areas 2193

 90.4.3 Reporting 2193

90.5 Case Study/Lessons Learned 2194

References 2197

M. Stachura (✉) • J. Elston • E.W. Frew
Department of Aerospace Engineering Sciences, University of Colorado, Boulder,
CO, USA
e-mail: stachura@colorado.edu; elstonj@colorado.edu; eric.frew@colorado.edu

B. Argrow
Director Research and Engineering Center for Unmanned Vehicles, University of Colorado,
Boulder, CO, USA
e-mail: brian.argrow@colorado.edu

C. Dixon
College of Engineering and Applied Science, University of Colorado at Boulder, Boulder,
CO, USA

Abstract

This chapter discusses the specifics of the Certificates of Authorization (COA) that were obtained for the second Verification of the Origin of Rotation in Tornadoes Experiment (VORTEX2) project and how the operations were conducted to satisfy the COA requirements. A strategy is outlined for operating these nomadic missions with small UAS within the confines of FAA regulations. This includes information on getting FAA COAs for a large area, specifically focusing on area selection, airworthiness, and emergency procedures, which are the keys to these applications. Interacting with the FAA once COAs have been granted is very important to the success of such a mission and is included in this chapter along with some lessons learned to improve future projects with similar goals.

90.1 Introduction

On 6 May 2010, on the plains of west Kansas, the Tempest unmanned aircraft system (UAS) made the first ever intercept of a supercell thunderstorm as part of the VORTEX2 field campaign (Elston et al. 2011b). This was surpassed on 10 June 2010, on the plains of eastern Colorado, with the first ever UAS intercept of a tornadic supercell (Elston et al. 2011b). With more than 100 scientists and engineers deploying more than 40 instrument platforms, VORTEX2 was the largest effort in history dedicated to the study of tornadoes (Oceanic and Association 2010). Two field campaigns were conducted 1 May–13 June 2009 and 1 May–15 June 2010; the Tempest UAS was deployed for the 2010 field campaign. The VORTEX2 participants maintained a completely nomadic program, roaming the Great Plains to track supercell thunderstorms. Supercells are the most violent form of severe convective storms, often producing the most violent and damaging tornadoes. The purpose of VORTEX2 is to study these storms for a better understanding of the ingredients that create supercells and to try to answer the question of why some supercells produce tornadoes while others do not (VORTEX2 SPO).

The development of the Tempest unmanned aircraft system (Fig. 90.1) addressed a variety of scientific, technical, logistical, and regulatory issues. Science drivers for the Tempest UAS focused on the need to sample pressure, temperature, humidity, and wind velocity in the rear-flank downdraft of supercell thunderstorms. Current models suggest that this area plays a causative role in tornadogenesis, and these models cannot be evaluated without in situ data. In turn, the science drivers lead to the requirement to sample the storm in areas where little is known about the flight conditions that will be encountered. As a result, the Tempest aircraft was developed for robust flight across a range of expected flight conditions. The dynamic nature of supercell thunderstorms necessitated a mobile concept of operations such that the unmanned aircraft could be stowed and transported by ground and then launched as quickly as possible. Finally, additional limitations were imposed by the Federal Aviation Administration (FAA) in order to satisfy regulatory requirements for the operation of unmanned aircraft systems.

Fig. 90.1 The Tempest unmanned aircraft (note the funnel cloud on the horizon, aligned with the midpoint of the aircraft)



The operations of unmanned aircraft systems in the U.S. National Airspace System (NAS) are governed by the Federal Aviation Administration, with guidelines established in *Interim Operational Approval Guidance 08–01, Unmanned Aircraft Systems Operations in the U.S. National Airspace System* (Policy 08–01) (Davis 2008). This document was created to define requirements and procedures for the operation of unmanned aircraft systems and it specifies that special approvals are required for the operation of UAS in the NAS (outside of active restricted, prohibited, or warning areas) because unmanned aircraft systems, are not compliant to many sections of Title 14 of the Code of Federal Regulations (14 CFR) (<http://www.gpoaccess.gov/cfr/>) and thus require an alternate means of compliance. This compliance is demonstrated by receiving a certificate of waiver or authorization (COA) or by obtaining a special airworthiness certificate—experimental category, the only two available methods of approval.

For public institutions, defined as *one that is intrinsically government in nature*, approval for operations of UAS is obtained through the COA process (Davis 2008). A COA defines operational requirements, emergency procedures, airworthiness requirements, an area of operations, and ground crew proficiency requirements for the operation of a single class of UAS. The overarching goal of the COA application is to demonstrate that the operations of the UAS have an equivalent level of safety to manned operations. In contrast to public institutions, private institutions and commercial companies (Ballinger and Bossert 2007) must also obtain a special airworthiness certificate, typically an experimental airworthiness certificate, for legal operations of their unmanned aircraft systems.

A recent FAA Aviation Rulemaking Committee (Tarbert and Wierzbanski 2009) reported “a comprehensive set of recommendations for small UAS regulatory development.” These regulatory recommendations have yet to be adopted, even in part. The regulations that must be satisfied for the operation of UAS remain those developed for manned aircraft resulting in a COA application process for UAS that is not straightforward. Prior to the 2006 “grounding” of UAS for those operating without a COA or an experimental certificate, many

small UAS operators claimed permission to operate according to FAA Advisory Circular AC 91–57 (Van Vuren 1981), which provides guidelines for the operation of hobbyist model aircraft according to the rules of the Academy of Model Aeronautics (AMA). With the creation of the Unmanned Aircraft Program Office (UAPO) in 2006 came enforcement of the COA and special airworthiness certificate requirements for public and civil UAS, respectively. The UAPO and the FAA Air Traffic Organization (ATO) were unprepared for the widespread misunderstanding of the COA application process, the subsequent onslaught of COA applications, and an almost complete misunderstanding of the requirements for civil entities to obtain UAS certification (Murphy and Argrow 2009).

Although the UAPO has worked to clarify regulations and submission requirements, rather than define a complete set of sufficient conditions for receiving a COA, the UAPO approach is to evaluate submitted UAS, operations, and procedures individually. As a result, information provided from the online application portal (http://www.faa.gov/about/office_org/headquarters_offices/avs/offices/afs/afs400/afs407/) can be vague, and determining the appropriate level of detail and requirements for each section of the application often requires several submissions, rejections, and subsequent iterations of the application.

In the interest of illuminating this process, this chapter describes the construction and subsequent use of 59 COA applications for the Tempest UAS used during the VORTEX2 field experiment. This chapter presents the information used to complete the COA application process and identifies decision and methods employed for a successful application, including the specification of operations area, how the airworthiness of the system was derived, the emergency procedures that were used, and the evolution of the actual operations and reporting requirements under the COA. The guidance given in this document follows that of Policy 08–01 (Davis 2008). It must be stressed that the information described here only pertains to the Tempest UAS, and the specific details included in our applications were based on the judgements of the authors and do not reflect definitive standards for a successful application.

90.2 Motivation and Application Strategy

The primary engineering objective for the Tempest UAS deployment was to develop a small UAS networked into a mobile command, control, and communications (C3) infrastructure that could meet the requirements for supercell penetration, specifically in the rear-flank downdraft (RFD), a region considered critical to the understanding of tornado formation. The UAS would also have to meet requirements for portability, with the mobility to target and track supercell thunderstorms, then enable rapid launch, storm penetration, and recovery of the unmanned aircraft.

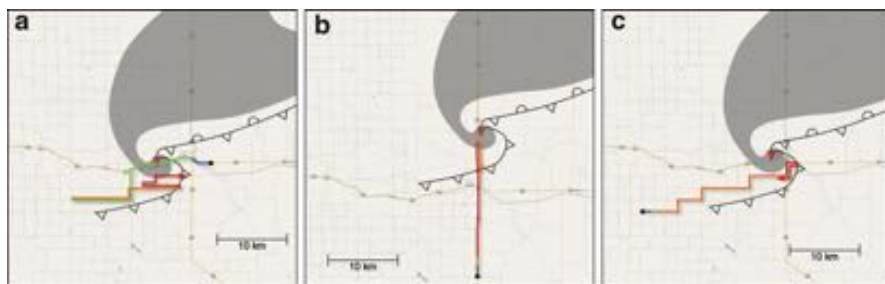


Fig. 90.2 Three sampling scenarios designed for use in the 2010 VORTEX2 campaign (Elston et al. 2011b) (a) S1 = standard inflow launch parallel to storm motion (b) S2 = inflow launch perpendicular to storm motion (c) S3 = outflow launch parallel to storm motion

Although the engineering requirements for the Tempest UAS were ultimately driven by the meteorological science objectives, for VORTEX2 the purpose was to demonstrate the feasibility of a small UAS for supercell sampling, so the science objectives were secondary to the engineering objectives. The primary science objective was to collect in situ measurements of pressure, temperature, and humidity, targeted in the rear-flank downdraft (RFD) and its outflow (VORTEX2 SPO). The RFD generally forms on the southwest portion of the storm, and when the downdraft approaches the ground, it spreads horizontally to form the RFD gust front. The focus of the Tempest UAS science mission is to fly into the RFD from the east or the south, crossing the gust front during the ingress.

Figure 90.2 from Elston et al. (2011b) illustrates three sampling scenarios. In S1 the UA is launched from the east of the supercell, then flies beneath the hook of the mesocyclone, near the tornado indicated by the small red triangle, into the RFD. The UA then proceeds to fly multiple transects, where the different colors of the trajectories indicated transects at different altitudes. For S2, the UA approaches the mesocyclone hook from the south, crossing the RFD gust front with its horizontal outflow before the UA makes contact with the main downdraft of the RFD and the tornado. S3 is the most difficult of the three scenarios, with an approach from the southwest which means that the UA is chasing supercells that typically move on northeast to easterly tracks.

Development of the concept of operations (CONOPS) for the Tempest UAS, and subsequent COA applications, focused on safe, assured operations by maintaining situational awareness of the UAS and airspace at all times. The main barrier to safe operation that satisfies FAA regulations is the ability to perform “sense and avoid” whereby the operational airspace is continually monitored and deconflicted, i.e., the UAS is kept clear of other airborne traffic. Though the FAA allows different options for providing sense and avoid, (e.g., stationary visual observers on the ground, visual observers in a chase aircraft, ground-based radar, radio transponders, and direction from air traffic control), stationary ground observers were used as the only solution during VORTEX2.

A second major factor in the COA application process was the dynamic, mobile pace of operations required to sample supercell thunderstorms. The conditions for supercell thunderstorm formation can become evident several days in advance. However, pinpointing the location and trajectory of a particular storm is difficult. Further, the onset of tornado formation cannot be predicted reliably in advance (the whole point of the VORTEX2 mission). Tornadogenesis within a supercell thunderstorm has been observed to occur in as little as 13 min from the first manifestation of potential tornadic activity (Erickson and Brooks 2006). As a result, successful operation during VORTEX2 required the ability to establish flight profiles with minimal advance notification. The standard provisions for most COA's require at least 48 h notification to activate a COA area through Notice to Airman (NOTAMS) along with contacting other groups such as ATC or military operations groups depending on the area. This provision would make these types of operations impossible.

Many of the FAA requirements that needed to be satisfied affected directly the design of the Tempest system. These design decisions were a direct result of a previous project, the Collaborative Colorado-Nebraska Unmanned Aircraft System Experiment (CoCoNUE), that acted as a precursor to the VORTEX2 campaign with the goal of using a UAS to sample across an air mass boundary. One of the important lessons learned during these experiments is that maintaining eyes on the aircraft from a chase car (an FAA requirement to ensure airspace deconfliction) is only feasible if the aircraft is tasked to orbit the chase car. This coupled with the requirement of a stationary ground station led to the use of a tracker vehicle which needed to maintain a data link with the aircraft to share its GPS position and led to the use of an ad hoc network to allow the UA to simultaneously communicate with both the ground station and the tracker vehicle; see Elston (2011) for more information on the communication subsystem.

Some subsystems had to be added to the Tempest UAS to either satisfy FAA requirements or to increase safety to expedite the COA process including a 900 MHz tracking antenna with Yagi for up to 20 miles of range to ensure communication over the maximum range of COA area from stationary ground station, a 2.4 GHz data link with ad hoc communication and multi-hop routing protocol for communication from the aircraft to both the tracker vehicle and the stationary ground station, a COTS autopilot and airframe for self-certification, a tracker vehicle to stay with the UA, and a scout vehicle to check the roads ahead for the tracker vehicle.

90.3 COA Application

90.3.1 COA Area Selection

Certificates of authorization are issued for specific areas of operation. The FAA requires that the UA position be known with enough accuracy that air traffic controllers can inform nearby aircraft. To satisfy this requirement, UAS operators

are generally required to file a Notice to Airmen (NOTAM) 72–48 h in advance of operations. The NOTAM specifies the location and time of the flight within the area specified by the COA. Several institutions have obtained COAs to perform prolonged flights over large distances by filing specific flight plans 72–48 h before the intended operation (Ambrosia and Hinkley 2008). In contrast, the work motivating this work required flight within 3,000 ft of the ground with much shorter notification constraints. Therefore, the focus of COA area selection presented here is on flights with small UAS through class G and E airspace without a predetermined flight plan.

Since NOTAMs could not be issued far in advance, COA application size was kept small enough to insure that the position of the UA was known relatively well and to minimize the workload on air traffic controllers who must notify pilots approaching the active COA airspace. Given the context of the VORTEX2 mission (Elston et al. 2011b), it was desirable to operate the UAS in an area over 24,000 square miles (62,000 sq km) in size (the green box in Fig. 90.3). Based on informal discussion with the FAA Unmanned Program Office, it was determined that this area was too large for a single COA.

To obtain permission for flying in this large area, the authors followed the example of the University of North Dakota's successful COA applications in multiple locations across their state (Douglas Marshall and Benjamin Trapnell, personal communication, 17 Dec 2008.), where they broke an area into several COA areas made up of approximately 10×10 mile (16×16 km) polygons. In the case of the Tempest COAs, the domain of operation was split into 59 areas approximately 20×20 miles (32×32 km) in an attempt to balance the size of the areas with the number of COA applications that would be needed to cover the complete mission domain (Fig. 90.3). In a debrief to the Denver Air Route Traffic Control Center (ARTCC) following the VORTEX2 operations, air traffic controllers indicated that circular areas are the preferred shape to issue a conventional NOTAM, specified by a radial distance from a geographic point. (Denver ARTCC, personal communication, 14 Oct 2010.) Future applications and renewals will follow this guidance.

After selecting the maximum size of the bounding polygon, the following additional items were considered to determine its shape: population density, major roads, airports, and airspace (Fig. 90.4). These were used to restrict the polygons to regions that could be shown to have low potential for property damage or human injury in the case of a failure of the UAS. The specific values were chosen heuristically with the expectation that they would satisfy the FAA UAPO. The minimum limits are 3 miles (5 km) from any town of significant size (population >1,000), 3 miles (5 km) from any airport, 1 mile from any major road (such as an interstate highway), and within class E or G airspace.

The FAA determines the maximum altitude for operations based on an estimate of the ability of an observer with a class 2 medical certification to see the UA and be able to ensure airspace deconfliction with other aircraft. The Tempest COA applications requested a flight ceiling of 3,000 ft above ground level (AGL). However, with its 10.5-ft (3-m) wingspan and slender fuselage, the FAA imposed a



Fig. 90.3 Vortex 2 operations area (*red*), desired UAS operations area (*green box*), actual COA areas (*blue polygons*)



Fig. 90.4 Considerations for the sizing, shape and location of each *polygon*, with green arrows indicating minimum distances and obstacles including 5 miles or more from both airports and built-up areas, 1 mile from major highways, and consideration for Victor airways, which the FAA does not allow loitering in

ceiling of 1,000 ft (300 m) AGL for operation of the Tempest UAS, with some areas limited to a 400-ft (120-m) ceiling based on proximity to specific airport approach airspace. Victor airways must also be considered, and in some cases impact the decisions to provide permission for flights over 400 ft (120 m) AGL.

90.3.2 Airworthiness Statement

Policy 08–01 (Davis 2008) outlines the procedure to create an airworthiness statement and lists several standards that have acceptable criteria for the self-certification of a public aircraft. The Tempest airframe airworthiness statements are based upon the Department of Defense (DoD) handbook MIL-HDBK-516A ([http://www.everyspec.com/MIL-HDBK/MIL-HDBK+\(0500+-+0599\)/MIL-HDBK-516A-2069/](http://www.everyspec.com/MIL-HDBK/MIL-HDBK+(0500+-+0599)/MIL-HDBK-516A-2069/)), which is superseded by MIL-HDBK-516B ([http://www.everyspec.com/MIL-HDBK/MIL-HDBK+\(0500+-+0599\)/MIL-HDBK-516B-10216/](http://www.everyspec.com/MIL-HDBK/MIL-HDBK+(0500+-+0599)/MIL-HDBK-516B-10216/)). Although the following discussion refers to MIL-HDBK-516A, the reader should refer to the more recent MIL-HDBK-516B when preparing a new COA application.

These handbooks were created for the certification of military aircraft, including manned and unmanned aircraft that carry ordnance, so the user must determine the criteria that are relevant for civilian applications. Sections 4–19 of MIL-HDBK-516A contain criteria specific to the different systems and operational procedures that must be addressed for airworthiness, though it is clear that Sections 9, 17, and 18 of MIL-HDBK-516A do not apply to unmanned aircraft.

The Tempest airworthiness document submitted to the FAA contained the heading of each subsection (e.g., 12.x) followed by statements explaining how each criterion is addressed to guarantee airworthiness. In many cases the subsection was not pertinent to the Tempest UAS, and it was sufficient to include a statement indicating this fact. Examples of criterion in MIL-HDBK-516A that were not necessary to satisfy for the Tempest UAS are structural fatigue, flight envelope, aircraft stability, and avionics architecture. It is, however, necessary to outline the procedures and analyses that are used to guarantee these criteria or steps used to mitigate risks from possible failures.

Airworthiness of the Tempest UAS was demonstrated based on three main factors. First, the airframe was developed in collaboration with a commercial manufacturer with experience designing and constructing competition radio-controlled sailplanes. In particular, Skip Miller Models (Skip Miller) modified an existing design based on specifications for the VORTEX2 mission. Successful demonstration in remote control dynamic soaring (RCS [2011](#)), where aircraft routinely obtains high air speeds and accelerations, validates the ability of the construction techniques used in the Tempest airframe to provide sturdy and durable aircraft. Second, the commercial Piccolo SL autopilot (Cloudcap) used for the Tempest UAS has an established record of success, both in military systems and other unmanned aircraft that have obtained COAs, including other UAS operated by the authors. Third, the complete Tempest UAS avionics system, which includes redundant wireless communication channels, onboard supervisory computer, ground control station, and operator interface, has been demonstrated through flight operations of other aircraft (Brown et al. [2007](#); Frew et al. [2008](#); Houston et al. [2012](#)). An appendix with examples of checklists, flight logs, maintenance logs, and operational procedures was included in the COA application to document those items used to support safe operations and maintenance. General guidelines for preparing an airworthiness statement based on lessons learned from VORTEX2 and other flight operations are given in Elston et al. ([2011a](#)). Excerpts from the Tempest UAS airworthiness statement are given here to show the level of detail required. Heading titles correspond to the sections of MIL-HDBK-516A.

6. Flight Technology

6.1 Stability and Control

The airframe used for the Tempest UA is the Tempest glider, commercially available from Skip Miller Models. The Piccolo Light autopilot control system ensures stable flight characteristics when coupled with a stable aircraft such as the Tempest UA as

required in Sect. 6.1.2.3. The envelopes, as outlined in 6.1.6, will be safe because the Tempest airframe is a commercially available glider.

6.2 Vehicle Control Functions (VCF)

The physical components of the VCF (i.e., the servos, control links, control surfaces, and electrical connections) employ high-grade commercial components sized for the maximum GTOW. All control surfaces are fully articulated during preflight checks to visually verify clearances. The control link to the Piccolo Plus on the Tempest UA is a single point of failure, but contingencies are in place in case of failure as outlined in the Lost Communication section. Preflight checklists for the Tempest UA are used to minimize the risk of failure as per 6.2.2.36. The emergency procedures, as outlined in the Emergency Procedures section, are appropriate and address the full range of possible emergencies as required in 6.2.55.

The Tempest subsystems are powered by a lithium-ion battery packs. Power is monitored from the UAS ground station to indicate primary power loss. The contingency is to immediately power-down any payload and return the Tempest UA for immediate landing. In the unlikely event of complete power loss, the UA servos reset the control surfaces to a default setting for spiral glide to the ground to prevent an uncontrolled excursion outside the primary area of operations. The VCF software is provided commercially by Cloudcap for the Piccolo Plus autopilot. This software is extensively tested and is safe in all normal flight conditions. For emergency situations, the procedures used are outline in the Emergency Procedures document.

9. Crew Systems

The Tempest is unmanned; therefore, there are no crew systems.

90.3.3 Lost Communications and Emergency Procedures

Lost communications and emergency procedures are specified as part of the COA application. The contents of these sections of the COA application pertain to the operations during abnormal and emergency situations. It is impossible to develop guidelines and procedures to deal with all situations, so the Tempest UAS application enumerated them as best as possible and stated that the judgement, skill, and training of all persons involved in flight operations were necessary to bring an abnormal or emergency situation to a safe conclusion.

The policies and procedures used for the Tempest UAS were developed and refined through previous flight experiences with other aircraft (Brown et al. 2007; Frew et al. 2008; Houston et al. 2012). In general, the response to an in-flight emergency or severe change in weather is to bring the UA back to the main landing site and to begin landing procedures, using either manual or automatic landing. The COA applications stated that all incidents and accidents would follow reporting and notification processes and requirements as laid out in FAA Orders 8020.11,

7210.56, and in NTSB 830. During the VORTEX2 mission, there were no incidents that required reporting under these rules.

The Tempest UAS COA application was written specifically for a three-person team piloting an unmanned aircraft system using a Piccolo autopilot system ([Cloudcap](#)). FAA regulations require a two-person team consisting of a pilot in-command (PIC) and a trained, medically certified observer (Davis 2008). Typical Tempest UAS operations consist of the PIC acting as a mission commander and two pilots at control (PACs): the PAC-M with manual flight control through a handheld console and the PAC-O who monitors and controls the UA when it is in semiautonomous mode. Only one of these copilots will be the acting PAC at a given time; therefore, the PIC can perform one of the roles. For Tempest UAS flights during VORTEX2, the PIC always served as the PAC-M.

90.3.3.1 Emergency Procedures

Enumerating all the emergency procedures contained in the Tempest COA application is prohibitive here. The emergency procedures for the Tempest UAS COA application follow the general guidelines (Elston et al. 2011a) derived from experience with the Tempest and other UAS. Overall emergency procedures were designed to maintain safe operations when possible with aerodynamic termination utilized if the aircraft becomes uncontrollable.

The COA application states that a handheld VHF radio will be used to broadcast a distress message in some emergency situations to aid in maintaining a safe airspace when traffic is present in the COA airspace. Distress messages will be broadcast on the specified CTAF/UNICOM frequency of the airport listed in the operational summary of the COA application. The distress message will contain the following information and will be given in the order listed below. Once broadcasted, the message will be repeated every 5 min, upon changes in condition or known location or upon any update requests received. Once the urgent condition is over, a final message will be broadcast to notify any traffic that the emergency condition is over.

1. PAN-PAN, PAN-PAN, PAN-PAN
2. "Local traffic"
3. "This is unmanned aircraft Tempest"
4. The nature and type of the distress
5. Pilot's intention request
6. Last known position and heading
7. Altitude
8. Hours and minutes of propulsion battery remaining
9. Weather
10. Other useful information such as visible landmarks, aircraft color, and that no people are onboard

The emergency procedures outlined in the Tempest UAS COA application are broken down into categories. *Operation emergency procedures* describe how the UAS team responds to incidents involving personnel of the general operation of the aircraft. For example, this section outlines how to respond if the pilot is unable to perform duties or if the aircraft leaves the COA boundary. *Ground control station emergency* describes procedures for responding to ground control station failures,

such as loss of power in the various components. Finally, *Tempest UA* describes responses to emergencies derived from the aircraft, such as loss of power or failure of the navigation solution. The following sections provide an overview of some of the main procedures included in the Emergency Procedures.

Operational Emergency Procedures

- Pilot/Operator/Observer Is Unable to Perform Duties

If a crew member becomes incapacitated during a UA flight, then the duties of that person will be transferred to another certified crew member depending upon the status of the current flight and availability of crew members.

- Loss of Visual Line of Sight by Observer

If the observer loses line of sight (LOS) to the aircraft, then another team member will act as the observer if they have LOS. If no team member has LOS, then the PIC will initiate landing procedures if the airspace is deemed safe or the PIC will command aerodynamic termination if the airspace is not deemed safe.

- Loss of Communication Between PIC and Observer(s)

If there is a loss of communication between the PIC and observer with handheld radios, a backup method such as cell phones shall be used.

- Loss of Communication with ATC

Operations of the *Tempest UAS* will be conducted in class E or G airspace under VFR conditions. Due to these conditions, it is assumed that communications with the local ARTCC or Terminal Radar Approach Control (TRACON) will not be required, but a handheld VHF radio will be used to maintain situational awareness and direct communication to local air traffic if necessary.

- UA Leaves COA Boundary Uncontrollable

In this situation, the first priority of the PIC is to maintain as safe an airspace as possible surrounding the UA. Procedures described here should include broadcasting a PAN-PAN message and using all reasonable effort to maintain VLOS and to terminate the flight.

Ground Control Station Emergency

- Operator Interface Software/Computer Failure

Failure of the operator interface software or the computer running it means that the PAC-O no longer has situational awareness of the UA and can no longer command or control the UA. Procedures for this emergency consider both the case where the UA is in VLOS of the PAC-M and beyond. Communication procedures to the team members are outlined, along with appropriate steps to safely terminating the mission, including automated return to the location of the ground control station, automated landing, remote landing, and finally aerodynamic termination.

- Ground Station Failure

Procedures for ground station failure consider the devices connected (e.g., pilot console, ground station GPS) and the possibility of device disconnection or failure. They also address the possibility of total ground station failure. Appropriate actions include switching of flight modes, turning off DGPS, or terminating the mission.

Unmanned Aircraft Emergency

- Engine-Out

In the case of engine failure, if the aircraft altitude is high enough, the standard landing procedure is initiated. If the altitude is too low, the aircraft is brought down quickly and safely and with the observer helping locate landing areas that consider cases when property, people, or other manned aircraft are in the vicinity.

- Autopilot/Servo Power Loss

Complete loss of power to the autopilot (or servos if on a separate power source) is a catastrophic failure, resulting in the UA being uncontrollable and unpredictable in its flight path. In this situation, the first priority of the PIC is to maintain as safe an airspace surrounding the UA as possible and issuing the appropriate distress messages.

- Degrading Performance of the Autopilot

Degrading or poor performance of the autopilot can be caused by bad or failed sensors, actuators, or by improperly set or adjusted feedback gains. It is expected that most failures that show as degraded autopilot performance are primarily due to the failure of onboard sensors such as the rate gyros, accelerometers, a poor or bad GPS solution, circuit board temperature failure, or failure of the air data system (dynamic and static pressure). Procedures include assessing the failure and determining the appropriate action in a per-sensor manner.

90.3.3.2 Lost Communication and Lost Link Procedures

The Tempest UAS uses a primary communication channel for command and control uplink as well as data downlink. The primary communication link is provided by a 1-W 900 MHz spread-spectrum modem manufactured by Microhard Corp. and is supplied with the Piccolo SL autopilot system as original equipment. The same modem is used in the autopilot and in the Ground Control System (GCS). A lost link on the 900-MHz primary link is considered a lost communication event. Depending upon the operational mode of the UA, the Piccolo autopilot detects two types of lost communications: one is when the aircraft is under manually piloted mode and the other is when the Piccolo is operating under semiautonomous control. In either case, in the event of a lost communication emergency, the UA is no longer under control of the PAC and collision avoidance capability has been compromised. In this situation, the first priority of the PIC is to maintain as safe of an airspace surrounding the UA as possible. This is done by broadcasting a PAN-PAN message to any nearby traffic as described in the Emergency Procedures document.

When the autopilot is under manual pilot (or manual-assisted) control, a lost link is determined by the UA as not receiving any decodable manual pilot control packets for 2 s, referred to as the *pilot time out*. Once this condition is met, the autopilot will automatically switch to autopilot mode and enter into the closest waypoint plan at the time of the pilot time out.

If communications fail entirely for 10 s (defined as the *communication time out*), either under manual or semi-autonomous modes, the autopilot will take automatic action depending upon the status of the GPS time out. If the GPS time out has

been asserted prior to the communication time out, then when the communication time out occurs the autopilot will issue an aerodynamic termination. If the GPS time out has not occurred, then the autopilot will switch from the current flight plan to the emergency waypoint plan, defined by the lost communication entry point. If communication is reacquired during the orbiting phase of the emergency flight plan then the PAC can initiate landing procedures. If communication is not reacquired after 2 min of orbiting at the lost communication waypoint, the autopilot will automatically switch to the autoland segment of the emergency flight plan, and will begin an autonomous landing.

Finally, since the operation of the Tempest UAS will be well within the communication range of the 900 MHz link, failure of the communication link will tend to be hardware based. Therefore, after a lost communication event is detected, a crew member who is not currently tasked as the observer or PIC will be directed by the PIC to inspect the communication hardware in the GCS to make sure there is no visible problems with the 900 MHz antenna, its location, or in the cabling.

90.4 Using COAs and FAA Interaction

This section describes additional provisions included in the actual COAs issued for the Tempest UAS and notification procedures prior to flight. In general, there is little interaction between the FAA Unmanned Program Office and the applicant once the COA application was submitted. As a result the final approved Certificate of Authorization could contain additional provisions or changes from the application. Further, the COA describes procedures for interacting with air route traffic control centers (ARTCCs) prior to flight.

90.4.1 COA Provisions

Fifty-nine distinct Certificates of Authorization were issued for the Tempest UAS participation during VORTEX2. The COAs included some changes from the original application and additional provisions that were not stated in the application. The original Tempest UAS COAs required notification 72–48 h in advance of flight operations. However, during the VORTEX2 campaign, the uncertainty of forecasting the time and location of target supercell thunderstorms required shortening the advance notice window. After submitting evidence to the UAPO and the ARTCCs in the COA areas, “pen-and-ink” (A pen-and-ink change is terminology used by FAA to indicate a minor change to a previously issued COA document.) changes were made to the Tempest UAS COAs to shorten the notification requirements to 2 h.

Table 90.1 describes the main provisions directly stated in the COA document. These include weather conditions suitable for flight operations, documentation and additional clearances required by the UAS operations team, and operational requirements. The main significant difference between the final COA and the

Table 90.1 Summary of Main FAA Requirements for Tempest UAS Operations

| |
|---|
| <i>Weather minimums</i> |
| Visibility of 3 statute miles |
| UA must maintain 500-ft below and 2,000-ft lateral separation from clouds |
| Daytime operations only: 1 h before sunrise until 1 h after sunset |
| <i>Required documentation</i> |
| Vehicle airworthiness document |
| Class 2 medicals for both the PIC and observer |
| PIC knowledge of FAR Part 91 (class G airspace) |
| Current private pilot certificate for the PIC (class E airspace) |
| Hard copy of COA document |
| <i>Operational requirements</i> |
| PIC must have 3 qualified events in the last 90 days |
| Single UA operation only |
| No dropping anything from the UA |
| No loitering in Victor airways |
| No operation in GPS test area or degraded RAIM |
| PIC and observer must be in constant two-way communication |
| UA cannot exceed 1,000 ft vertical and 1/2 or 1 mile horizontal from observer |
| NOTAM must be issued 2 h prior to flight operations |
| Incidents, accidents, and COA boundary deviations must be reported 24 h prior to any additional flights |
| Monthly reports of operations must be submitted to the FAA |

application was a reduction in allowable flight altitude (reduced from 3,000 to 1,000 ft) and allowable lateral separation (reduced from 1 to 1/2 mile). Through personal communications, the FAA UAPO indicated that these limits represented their judgement of the ranges at which the aircraft could be seen visually by the ground observer.

The specific requirements and procedures for UAS operations are contained in the COA document. These are the same procedures outlined in Policy 08–01 (Davis 2008) applied to the specific UAS and COA area that include specific statements of requirements for crew proficiency (e.g., PIC, observer, medicals, knowledge of FAR 61.105, 91.(3,7,13,17,111,113), limits, and currency and instructions for contacting an ARTCC or TRACON. The COAs required the pilot in command to have a current private pilot certificate. This requirement was in place to insure that the PIC understood the operational procedures of aircraft in the U.S. National Airspace System, not because the Tempest UAS handled like a manned aircraft. The PIC and the observer, who monitored the aircraft at all times, were also required to pass class 2 medical certification.

90.4.2 Activating COA Areas

Several steps are required prior to flight operation in order to notify air route traffic control centers and local airfields of pending UAS flights. The first step in activating a COA is to issue a Notice to Airmen (NOTAM) (<https://pilotweb.nas.faa.gov/PilotWeb/>) describing the planned activity. For the VORTEX2 mission, NOTAMs could be issued for up to four different COA areas with 2-h advance notice. Because the ARTCCs notify all pilots of NOTAMs in a given area, it was not feasible to issue NOTAMs for all 59 COA areas. The 2-h advance notice was at the limits of the meteorologists' abilities to predict thunderstorm evolution. As a result, NOTAMs could be rescinded and issued for different areas as needed, resetting the 2-h wait time before flight operation could commence.

Although the COA areas in the applications were specified by the coordinates of the bounding polygon, NOTAMs are issued based on radial distance and direction from Very High Frequency Omni-Directional Radio Range Tactical Air Navigation Aid (VORTACs). As a result, a better general strategy for COA applications is the use of circular regions that can be specified easily from a VORTAC. Because the COA areas and VORTACs are stationary, all pertinent information needed to issue a NOTAM for a specific COA could be determined in advance. The example NOTAM in Table 90.2 has all of the necessary information for other pilots operating in the NAS. The NOTAM, number 09/003, informs of UAS operations in a 3NM radius centered around a point that is 15NM from the BJC VORTAC off the 330° radial. The operations are conducted 400-ft AGL and below from time 1500Z to 2100Z on 3 September 2010. See FAA Order JO 7930.2M, section 6-1-7b (<http://www.faa.gov/documentLibrary/media/Order/NTM.pdf>) for more details on NOTAMs for UAS.

The final step in the notification process occurred 30 min before flight operations could commence. At this point it was necessary to contact by telephone the Denver ARTCC and any nearby airport or military operations desks. They will require both the COA number and the NOTAM number associated with the flight.

90.4.3 Reporting

There are three types of reports required by the FAA for UAS operation: accident (NTS 2010), incident (NTS 2010), and monthly operational. These are all accessed through the secure FAA web portal (<https://ioeaaa.faa.gov/oeaaa/>). There are also real-time reporting requirements for loss of communication and/or violation of COA boundaries, which are covered in the Emergency Procedures section. Monthly operational reports were submitted for all active COAs within five business days of

| | |
|-------------------------|---|
| Table 90.2 NOTAM | 09/003 – AIRSPACE UNMND AC 3 NMR BJC330015 |
| Example | 400/BLW. 03 SEP 15:00 2010 UNTIL 03 SEP 21:00 2010. CREATED: 31 AUG 22:42 2010 |

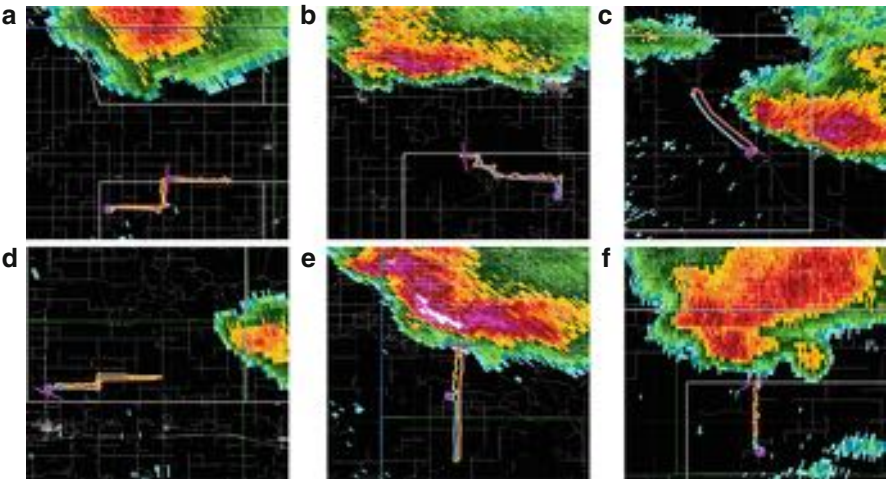


Fig. 90.5 Composite radar, flight path, and COA boundary for supercell intercepts during the VORTEX2 campaign (Elston et al. 2011b) (a) 6 May 2010, sampling scenario S3 (b) 26 May 2010, sampling scenario S2 (c) 6 June 2010, sampling scenario S1 (d) 7 June 2010, sampling scenario S3 (e) 9 June 2010, sampling scenario S2 (f) 10 June 2010, sampling scenario S2

the end of each month. They were submitted even if no operations took place under a specific COA. The reports required information about the COA along with number of operations and total hours.

90.5 Case Study/Lessons Learned

A synopsis of a typical deployment will be presented next to better illustrate the timing and pace of nomadic operations. This will be followed by some of the lessons learned along with specifics on what can be improved for future nomadic sampling missions within the confines of FAA regulations. There were six successful flight experiments flown into storms during the VORTEX2 deployment. Figure 90.5 contains trajectories of six different flight experiments that were done during the VORTEX2 campaign utilizing different sampling scenarios.

The planning phase of the deployment was conducted from 36 to 2 h prior to launch of the UA. This began with VORTEX2 PIs using data and models to predict

storms the evening before flight experiments. It continued the next morning with examining any new data and setting a departure time and initial destination. The entire armada would then begin driving to the destination, with any changes to the target area being relayed to the individual team leaders.

At approximately 2 h prior to the predicted launch time, the meteorology lead of the UAS team would select 4 COA areas that were most likely to be within the target area, and these would be activated using NOTAMs. It should also be noted that there were other notifications that may have to go out depending on the specific COA areas such as nearby airports or Air force bases. Since these were very specific to each COA area, they will not be mentioned in detail here. These groups required anywhere from 2 h to 5 min notice prior to launch. Also, during this phase of the deployment, NOTAMs could be cancelled and new ones put in if the storm changed direction. However, 2 h lead time was required from whenever the new NOTAM was issued.


At 1 h prior to launch the flight preparations would commence for all parties utilizing checklists. An example of one such checklist is given in Table 90.3 for the ground control station operator. Somewhere in this time up to 10 min to launch the meteorology lead would select a launch area and begin preparing the flight plan. The team would then arrive at the location, finishing prepping the UA, deploy all mobile ground vehicles, and launch the UA.

During the flight experiment the flight plan would be changed depending on both RADAR imagery of the storm and in situ data from the UA. Following the flight experiment the UA would return to base, land, and all postflight checklists would be utilized. This would include cancelling any active NOTAMs, notifying any flight service groups as required by the COAs, and logging all necessary data for monthly COA reports. For a more detailed description of the entire CONOPS, see Elston et al. (2011b).

The system designed for VORTEX2 led to six successful and safe flight experiments with no deviation from FAA rules for UAS. There were however several important lessons learned that could lead to improved operations in the future and more useful data for the scientific community.

The first major issue that was encountered was the issue of predicting which COA areas the storms would pass through. The standard provisions require activating a COA area between 48 and 72 h prior to flight operations. The initial solution was to simply activate all 59 COA areas 2 days before we planned on flying. However, this was found to be unsustainable for ATC since each COA area required a separate NOTAM, and this many would overwhelm their system. In order to address this issue, we worked with the FAA to reduce the activation time to 2 h. Permission was also given to activate up to 4 COA areas at a time. However, the dynamic nature of the storms made even this difficult to predict. An example of when this issue came up was during sampling the June 10th storm; the team had to wait for the start of time of the NOTAM while there was already a tornado on the ground from the storm. Being able to deploy sooner than the required-2 h time could potentially lead to more useful data. The FAA has since reduced this time to 1 h for the 59 COA areas.

Table 90.3 Example of One of the Checklists Used

| |
|--|
|  |
| Operator checklist |
| <i>After power on</i> |
| Turn off van Wi-Fi |
| Start averaging groundstation GPS |
| Disable engine |
| Set the pilot address |
| Copy commands and verify all loops are auto |
| Uncheck auto center and zoom as appropriate |
| Verify COA area altitude and mission limits altitude |
| Set up flight plans: lost comm, take-off, landing |
| Zero air data with GPS altitude |
| Test pitot-tube airspeed |
| Turn on APS, verify tracking, and set to take-off orientation |
| Start up B.A.T.M.A.N on focus through ssh |
| <i>Preflight</i> |
| Check Piccolo voltage |
| Check Servo voltage |
| Check UA GPS |
| Set the tracked waypoint to 10 |
| Check Piccolo 900MHz link |
| Verify sonde operation |
| Launch, start NetUAS timer |
| Start APS tracking |
| Notify tracker of handover |
| <i>Postflight</i> |
| Kill engine |
| Save config if it was changed |
| Close OI and save log files to a new (reliable) location |
| Run NetUAS log save scripts |
| Turn off Piccolo GCS |

The second major issue we encountered was the boundaries of the COA areas. [Figure 90.5a, b, f](#) were successful sampling missions that got as far as the active COA boundary and could not sample any further losing out on some more data that may have proved useful. There were also several other potential missions that never flew because the storm was just outside of the COA area, including a very promising one on June 11th. The solution to this problem is not as obvious. The FAA did not want us to have a COA area larger than 20 × 20 miles since that would be too large an area to keep traffic out of. Also, there is the issue of avoiding airports,

we arbitrarily chose 5 miles and major roads. A potential solution to this issue is to grant a single large COA spanning the entire 200×200 mile area and restrict us to activating only a 20-mile diameter circle in that area at a given time where we are responsible for avoiding airports and major roads. This gives the meteorologist more precision to work with when activating COA areas. It was also found that ATC prefers circular areas for the NOTAMS since they can just list a center point and radius in the NOTAMS.

Conclusion

The capabilities of current UAS to perform these nomadic science missions currently exist, and the main bottleneck for these missions is satisfying FAA regulations to perform safe operations. It is important to work with the FAA to reach a compromise that satisfies both the scientific and engineering goals of the project while being able to prove the necessary level of safety for operating in the National Airspace System. This chapter used experience from the VORTEX2 UAS campaign to present information on obtaining COAs, working with the FAA to conduct successful flight operations, and some lessons learned that could lead to improved operations in the future.

References

- V.G. Ambrosia, E. Hinkley, Nasa science serving society: improving capabilities for fire characterization to effect reduction in disaster losses, in *IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2008*, Boston, vol. 4, 2008, pp. IV-628–IV-631. doi:10.1109/IGARSS.2008.4779800
- M. Ballinger, D. Bossert, Faa certification process for a small unmanned aircraft system: one success story, in *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, Rohnert Park, CA, 2007
- T.X. Brown, B.M. Argrow, E.W. Frew, C. Dixon, D. Henkel, J. Elston, H. Gates, Experiments using small unmanned aircraft to augment a mobile ad hoc network, in *Emerging Technologies in Wireless LANs: Theory, Design, and Deployment*, chapter 28, ed. by B. Bing (Cambridge University Press, Cambridge, 2007), pp. 123–145. ISBN-13:9780521895842
- Cloudcap, The cloudcap website (2011), <http://cloudcaptech.com>
- K.D. Davis, *Interim Operation Approval Guidance 08–01: Unmanned Aircraft Systems Operations in the U.S. National Airspace System*, FAA Unmanned Aircraft Systems Program Office, 2008
- J. Elston, Semi-autonomous small unmanned aircraft systems for sampling tornadic supercell thunderstorms. Ph.D. thesis, University of Colorado, 2011. data/publications/11_thesis.pdf
- J. Elston, M. Stachura, B. Argrow, C. Dixon, Guidelines and best practices for faa certificate of authorization applications for small unmanned aircraft, in *AIAA Infotech@Aerospace Conference*, St. Louis, MO, 2011a
- J.S. Elston, J. Roadman, M. Stachura, B. Argrow, A. Houston, E.W. Frew, The tempest unmanned aircraft system for in situ observations of tornadic supercells: design and vortex2 flight results. J. Field Robot. (2011b). Accepted <http://www.journalfieldrobotics.org/Home.html>
- S.A. Erickson, H. Brooks, Lead time and time under tornado warnings: 1986–2004, in *23rd Conference on Severe Local Storms*, St. Louis, MO, 2006
- E.W. Frew, C. Dixon, J. Elston, B. Argrow, T.X. Brown, Networked communication, command, and control of an unmanned aircraft system. *AIAA J. Aerosp. Comput. Inf. Commun.* **5**(4), 84–107 (2008)
- A. Houston, B. Argrow, J. Elston, J. Lahowetz, P. Kennedy, The collaborative colorado-nebraska unmanned aircraft system experiment. *Bull. Am. Meteorol. Soc.* **93**(1), 39–54 (2012)

- R. Murphy, B. Argrow, Uas in the national airspace system: research directions. *Unmanned Syst.* **27**(6), 23–28 (2009)
- National Oceanic and Atmospheric Association, Vortex2: verification of the origins of rotation in tornadoes experiment (2010), <http://www.nssl.noaa.gov/vortex2/>
- Part 830 notification and reporting of aircraft accidents or incidents and overdue aircraft, and preservation of aircraft wreckage, mail, National Transportation Safety Board, 2010
- RCSpeeds (2011). <http://rcspeeds.com/aircraftspeeds.aspx?rpt=LL>
- Skip Miller, The skip miller models website (2010), <http://skipmillermodels.com>
- B. Tarbert, T. Wierzbowski, *Comprehensive Set of Recommendations for Suas Regulatory Development*, FAA Small Unmanned Aircraft System Aviation Rulemaking Committee, 2009
- R.J. Van Vuren, *Advisory Circular 91–57: Model Aircraft operating Standards*, FAA Air Traffic Organization, 1981
- VORTEX2 SPO, Vortex2 scientific program overview (2007), <http://www.vortex2.org/Documents/vortex2-spo-2007-0131.pdf>

Konstantinos Dalamagkidis

Contents

91.1 Introduction 2200

91.2 Equivalent Level of Safety 2201

 91.2.1 Manned Aviation Requirements 2201

 91.2.2 Derivation of an ELOS for UAS 2202

91.3 UAS Accident Types 2205

91.4 Ground Impact Fatality Risk Modeling 2206

 91.4.1 Ground Impact ELOS 2208

 91.4.2 Exposure to Ground Impact Accidents 2209

 91.4.3 Probability of Fatality of Exposed Persons 2210

 91.4.4 Frequency of Ground Impact Accidents 2215

91.5 Midair Collision Fatality Risk Modeling 2216

 91.5.1 Midair Collision ELOS 2216

 91.5.2 Exposure and Risk of Fatality 2218

 91.5.3 Conflicting Trajectory Expectation 2219

 91.5.4 Collision Probability 2219

91.6 Model Choice 2220

91.7 Case Studies 2223

91.8 Conclusion 2226

References 2227

Abstract

This chapter presents aspects of risk modeling with a focus on UAS. It provides an overview of the current level of safety of manned aviation in terms of accident statistics. These are then mapped as target levels for UAS under the “Equivalent Level of Safety” principle to provide a glimpse at what that may entail for UAS regulations. Different methodologies are presented for estimating the risk of ground impact and midair collision accidents and how these estimates can be

K. Dalamagkidis
Institut für Informatik I6, Technische Universität München, Garching bei München, Germany
e-mail: dalamagkidis@tum.de

translated to system requirements. The chapter also provides guidelines on the use of different risk models and then applies a selection of them to five different UAS in two distinct scenarios, to compare the results of different choices.

91.1 Introduction

The primary goal of regulating UAS operations is to assure an appropriate level of safety for those directly involved or indirectly affected. Many national aviation agencies have stated that the goal is to achieve an “Equivalent Level of Safety” with that of manned aviation, also known as the ELOS principle.

Since many UAS are based on military or general aviation aircraft, they share aspects of the design and construction of the airframe and mechanical components. This of course does not mean that unique designs have not evolved as well. The key difference between manned and unmanned aviation lies in the separation of the pilot from the cockpit and the level of automation introduced. This difference introduces new failure modes and, as a result, an increased perceived risk that needs to be evaluated and mitigated. Nevertheless, it is noteworthy that manned aviation has also benefited from increased automation. According to Haddon and Whittaker (2002), a considerable percentage of modern commercial aviation operations – including landing – takes place autonomously with the pilots responsible only for monitoring the computers.

This chapter is an analysis of what the ELOS requirement may entail for UAS regulations. To accomplish this, the safety performance of manned aviation is first evaluated. Then, an analysis is carried out to derive a model for estimating UAS risk as the number of fatalities following an accident. Methods to derive estimates for the parameters that affect risk and the rationale for choosing between the alternatives are also presented. This analysis targets two major accidents types, ground impact and midair collisions. It should be noted that actual regulations will need to take into account a number of issues and as a result may in the end diverge significantly from the conclusions of this chapter.

This chapter cites material from two publications of the U.S. Range Commanders Council. The first is the RCC 321-07 titled “Common Risk Criteria Standards for National Test Ranges,” while the second is RCC 323-99 titled “Range Safety Criteria for Unmanned Air Vehicles.” Both are to be used by range commanders to ensure that operations within their ranges adhere to safety standards that are acceptable by the military. The following is quoted from the first chapter of Range Safety Group, Range Commanders Council (2007a):

The policies and criteria in this document are intended for use by members of the DoD national ranges and Major Range and Test Facility Base (MRTFB). These policies and criteria apply to launch and reentry hazards generated by endoatmospheric and exoatmospheric range activities including both guided and unguided missiles and missile intercepts, space launches, and reentry vehicles. This does not include aviation operations or UAV operations. The RCC Document 323-99 (Range Safety Criteria for Unmanned Air Vehicles) provides criteria for unmanned air vehicles.

Although this distinction is made, it was felt that in certain cases, the reader would benefit from a presentation of certain, selected information from RCC 321-07. This is because either this information is general and applies to most risk/reliability assessments or valuable insight is to be gained from contrasting it with information specific to UAS.

91.2 Equivalent Level of Safety

According to Joint JAA/Eurocontrol Initiative on UAVs (2004) and European Aviation Safety Agency (2009), one of the guiding principles for UAS regulation should be *equivalence*, and based on that, they assert the following:

Regulatory airworthiness standards should be set to be no less demanding than those currently applied to comparable manned aircraft nor should they penalize UAS systems by requiring compliance with higher standards simply because technology permits.

This principle has been widely adopted by most national aviation agencies worldwide and is known as the ELOS requirement. For example, the Range Commanders Council in its guidance on UAS operations states:

Any UAV operation or test must show a level of risk to human life no greater than that for an operation or test of a piloted aircraft.

Range Safety Group, Range Commanders Council (1999a).

It should be noted that there is significant criticism aimed at the usefulness of this principle in general. This is because of the difficulty inherent in specifying what exactly the ELOS requirement entails and more importantly quantifying it. In any case, in an effort to better define what the ELOS could be, the next section investigates the requirements of current regulations for manned aviation.

91.2.1 Manned Aviation Requirements

Manned aviation is regulated through a code of requirements, which often refers to standards for various aircraft subsystems as well as for all stages of design, manufacture, and operation (Haddon and Whittaker 2002). Use of standards ensures that the components of the system are reliable enough so that the whole system is compliant with a set target level of safety (TLS). Regulations normally also contain safety targets found in *paragraph 1309* of current CS or the corresponding AMC sections. These targets are typically presented as a risk reference system. Such a system categorizes events based on their severity and assigns a maximum rate of occurrence for each event category.

Figure 91.1 presents the risk reference system proposed in the *1309 AMC* section of EASA CS 25. There, a failure condition that includes injuries and/or fatalities is categorized as hazardous and as such it should be extremely remote ($<10^{-7}$ events per flight hour) (European Aviation Safety Agency (EASA) 2007). On the other

| | | Catastrophic | Hazardous | Major | Minor | No safety effect |
|----------------------|----------------------------|--------------|-----------|-------|-------|------------------|
| Probable | $> 10^{-5} \text{ h}^{-1}$ | | | | | |
| Remote | $< 10^{-5} \text{ h}^{-1}$ | | | | | |
| Extremely remote | $< 10^{-7} \text{ h}^{-1}$ | | | | | |
| Extremely Improbable | $< 10^{-9} \text{ h}^{-1}$ | | | | | |

Fig. 91.1 Risk reference system for large manned aircraft (the *grayed areas* signify unacceptable risk) (Source: European Aviation Safety Agency (EASA) (2007))

Table 91.1 FAR Part 23 aircraft classes and corresponding acceptable failure condition probability based on severity, as defined in AC 23.1309-1C (Source: Federal Aviation Administration (1999))

| Aircraft class | Minor | Major | Hazardous | Catastrophic |
|---|-----------|-----------|-----------|--------------|
| Class I (<2,720 kg, SRE) | 10^{-3} | 10^{-4} | 10^{-5} | 10^{-6} |
| Class II (<2,720 kg, STE, MRE) | 10^{-3} | 10^{-5} | 10^{-6} | 10^{-7} |
| Class III (>2,720 kg, SRE, MRE, STE, MTE) | 10^{-3} | 10^{-5} | 10^{-7} | 10^{-8} |
| Class IV (commuter) | 10^{-3} | 10^{-5} | 10^{-7} | 10^{-9} |

SRE, single reciprocating engine; *MRE*, multiple reciprocating engine; *STE*, single turbine engine; *MTE*, multiple turbine engine

hand, the same publication considers multiple fatalities to be of catastrophic severity with a likelihood requirement of 10^{-9} or less.

The risk reference system presented in Fig. 91.1 does not apply to all aircraft, and variations exist for smaller or different types of aircraft. This is because it was found that applying certification standards developed for transport category aircraft to smaller ones led to unrealistically high equipment reliability requirements (Federal Aviation Administration 1999). In addition to that, the results of accident investigations showed that the main accident cause in manned aviation is pilot error. As such, high equipment reliability would have only a minor effect on overall aviation safety. In 1999, the FAA issued AC 23.1309-1C that contains AMC for aircraft certified based on FAR Part 23. With this AC, four classes of aircraft within that category were defined, each with different acceptable probabilities for failure conditions, as shown in Table 91.1.

91.2.2 Derivation of an ELOS for UAS

Use of the same risk reference system like the one presented in Fig. 91.1 or even Table 91.1 is not straightforward because of the wide range of UAS sizes and characteristics. In addition to that, UAS depend on the onboard flight control system and/or the communication link to operate. This requirement introduces additional failure modes that may increase the total number of accidents for the same reliability requirement. On the other hand, since UAS do not carry passengers, the number of

Table 91.2 Fatality rates from all accidents based on analysis of NTSB accident data (National Transportation Safety Board (NTSB) 2008b) between 1983 and 2006

| Rates per hour | Air carrier | Commuter | General aviation | Total |
|-------------------|----------------------|----------------------|----------------------|----------------------|
| Accident | $2.43 \cdot 10^{-6}$ | $2.37 \cdot 10^{-5}$ | $8.05 \cdot 10^{-5}$ | $5.05 \cdot 10^{-5}$ |
| Fatalities aboard | $8.68 \cdot 10^{-6}$ | $1.64 \cdot 10^{-5}$ | $2.77 \cdot 10^{-5}$ | $2.06 \cdot 10^{-5}$ |
| Ground fatalities | $3.37 \cdot 10^{-7}$ | $8.30 \cdot 10^{-6}$ | $6.54 \cdot 10^{-7}$ | $1.31 \cdot 10^{-6}$ |

people exposed to risk can be significantly lower. As a result, the probability of injuries and fatalities after an accident is greatly reduced when compared with that of general aviation or transport aircraft. The average number and severity of injuries per accident is also expected to be lower.

Since failure frequency requirements prescribed for manned aircraft of the same size cannot be used directly, other means to derive such requirements for UAS need to be employed. A different approach frequently used in safety engineering is to define safety constraints for a specific accident based on the desired likelihood of the worst possible outcome (Range Safety Group, Range Commanders Council 1999b). This can in turn be used to determine maximum failure frequency. For UAS operations, the worst outcome of most accidents is the occurrence of one or more fatalities, and as a result, the ELOS can be based on that. Although the ELOS may be based on the frequency of accidents that result in one or more fatality, such an ELOS would also be inadequate since it fails to consider the effect of multiple fatalities.

Although current manned aviation regulation does not impose limits on fatality rates, a statistical analysis of historical data can provide valuable insight on the fatality rates of manned aviation and be the basis for defining the ELOS for UAS. An analysis of NTSB accident data from 1983 to 2006 is presented in Table 91.2. It should be noted that the exact rates may vary depending on the type of aviation (general, regional/commuter, air carrier). Moreover, the period over which the data are averaged can also play a role (Clothier and Walker 2006), since there is significant variation from year to year, as shown in Fig. 91.2. This is also evident from a U.S. Navy survey (Range Safety Group, Range Commanders Council 1999b) that found an average of 18, 7, and 4.7 ground fatalities per 10 million flight hours for U.S. Navy, commercial aviation, and general aviation, respectively. The survey included data from 1980 to 1998 for U.S. Navy flights and from 1982 to 1998 for civil aviation. In most cases, an individual, annual risk of fatality of 10^{-6} is considered sufficiently low and below that of other typical everyday activities (Range Safety Group, Range Commanders Council 2007b).

Other approaches have employed serious but nonfatal injuries as the basis for their safety assessments, especially when due to the nature of the accident fatalities are unlikely. As an example, Range Safety Group, Range Commanders Council (2007b) proposes the use of casualty limits, the latter defined as the people sustaining injuries of a certain severity and over. A study of NTSB data provided in Range Safety Group, Range Commanders Council (2007b) shows that in aviation, the ratio of casualties to fatalities shows little variation and is close to 2.5. Based on this ratio, casualty limits can be derived from those used for fatalities. A common

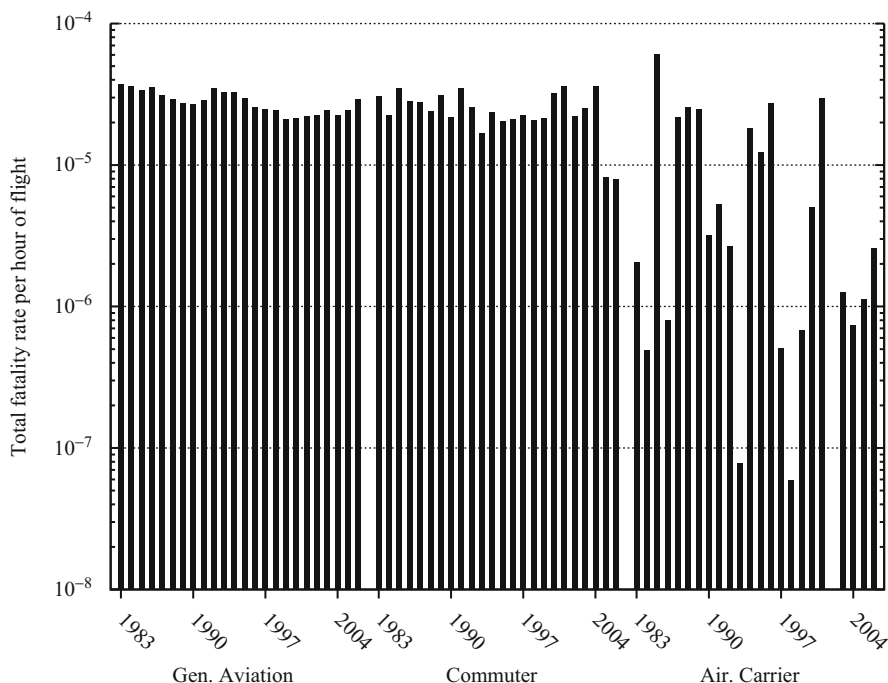


Fig. 91.2 Fatality rates from general aviation, commuter, and air carrier accidents as a function of time. Based on analysis of NTSB accident data (National Transportation Safety Board (NTSB) [2008b](#)) between 1983 and 2006

casualty limit is that of 10^{-4} casualties per event, which has been used by a number of agencies like the NASA, the U.S. DoD, and the Space Licensing and Safety Office of Australia (Range Safety Group, Range Commanders Council [2007b](#)). The same limit was chosen in Range Safety Group, Range Commanders Council ([2007a](#)), complemented by an individual casualty risk probability limit of 10^{-6} . Nevertheless, the use of fatalities has also been advised as a supplemental metric, to better assess the risk involved in an activity (Range Safety Group, Range Commanders Council [2007b](#)).

The aforementioned proposed casualty and fatality limits can be contrasted with estimated rates from other sources or activities, provided in Table 91.3, as well as aviation accident statistics given in Table 91.2. When comparing these rates, one must consider that in some activities, a higher risk may be acceptable because of the perceived benefits of participation. It should also be noted that these rates refer to collective risk (Range Safety Group, Range Commanders Council [2007b](#)), i.e., the averaged risk for the entire population. Depending on a person's location, activity, and other factors, the actual individual risk may be significantly higher or lower.

Table 91.3 Estimated injury, casualty, and fatality rates from different sources or activities

| Activity/source | Injury rate (h^{-1}) | Casualty rate (h^{-1}) | Fatality rate (h^{-1}) |
|--|---------------------------------|-----------------------------------|-----------------------------------|
| Motor vehicle accidents (all) ^a | $1.35 \cdot 10^{-5}$ | $1.13 \cdot 10^{-6}$ | $1.40 \cdot 10^{-7}$ |
| Motor vehicle accidents (occupant) ^a | $8.80 \cdot 10^{-6}$ | $6.73 \cdot 10^{-7}$ | $5.89 \cdot 10^{-8}$ |
| Pedestrian involved in collision with motor vehicle ^a | $5.10 \cdot 10^{-7}$ | $8.92 \cdot 10^{-8}$ | $1.04 \cdot 10^{-8}$ |
| Unintentional falls ^a | $2.45 \cdot 10^{-5}$ | $2.20 \cdot 10^{-6}$ | $6.06 \cdot 10^{-8}$ |
| Natural environment ^a | $1.31 \cdot 10^{-7}$ | $1.44 \cdot 10^{-8}$ | $7.59 \cdot 10^{-9}$ |
| Bicycles and accessories ^b | $1.50 \cdot 10^{-6}$ | $8.98 \cdot 10^{-8}$ | |
| Household appliances (ranges, refrigerators, washers) ^b | $4.23 \cdot 10^{-7}$ | $1.81 \cdot 10^{-8}$ | |
| Baseball, basketball, and football combined ^b | $2.59 \cdot 10^{-5}$ | $3.44 \cdot 10^{-7}$ | |
| London Blitz (civilian only) ^c | N/A | $1.04 \cdot 10^{-6}$ | $6.22 \cdot 10^{-7}$ |

^aThe number of injuries and fatalities are from the Web-based Injury Statistics Query and Reporting System available from the Centers for Disease Control and Prevention, National Centers for Injury Prevention and Control and correspond to emergency department admissions in the year 2005. The rates were derived based on an estimated population of 296,410,404 and assuming that every individual (regardless of age, sex, or location) is exposed to hazards involving each activity/source an average of 3 h per day. The casualty rate is based on injury cases that required hospitalization or transfer to other facilities such as trauma centers

^bThe data are from the National Electronic Injury Surveillance System maintained by the U.S. Consumer Product Safety Commission and concern the year 2007. The reported data did not make a distinction between incidents requiring hospitalization and incidents involving deaths. The rates are derived the same way as above, with the exception of the baseball, basketball, and football activities, where an average exposure of 3 h per week is assumed instead

^cFor the London Blitz, the civilian casualties were drawn from historical sources, and the rate was obtained by dividing by the population of London in 1939 and by the number of days the Blitz lasted, assuming continuous exposure

Significant differences can be expected between the effects of various accident types. As a consequence, the ELOS and the risk analysis that will follow need be derived for each accident separately. The following section provides an overview of the accidents of interest in the case of UAS.

91.3 UAS Accident Types

UAS operations are subject to various hazards that can lead to three primary accidents: unintended or abnormal system mobility operation (U.S. Department of Defense 2007), midair collision, and early flight termination (Clothier et al. 2007).

Unintended or abnormal mobility operation refers to accidents that occur when the UAS is still on the ground. In this case, the UAS may move unexpectedly, potentially seriously injuring ground crew members. Such accidents usually happen because of operator error. They may occur when the UAS operator does not have a view of the UAS and incorrectly assumes that everyone has cleared the area. This accident type will not be further investigated since the risk can be adequately mitigated with better management of operations and stricter adherence to standard operating procedures.

Midair collisions may occur between two UAS systems or between a UAS and a manned aircraft. Depending on the nature of the collision, they can result in the loss of one or both of the aircraft. A secondary accident usually following midair collisions is ground impact of debris that may injure people and damage property.

Finally, early flight termination, either controlled or uncontrolled, will result in ground or water impact. Under controlled flight termination, it may be feasible to select the point of impact and possibly the speed and orientation of the aircraft. In that case, this capability can be used to reduce the probability of fatalities as well as damages to property and the aircraft itself.

Potential damages resulting from all these accidents include injury or fatality of people on the ground or onboard another aircraft, damage or loss of the vehicle, and damage to property. An indirect damage is environmental pollution either from the payload of the aircraft or as a result of fuel leakage and/or fire following the accident. This is especially important for UAS that will carry chemicals toxic to human beings, for example, those used in agricultural applications.

A possible damage that is often ignored is that of societal rejection or outrage that may disrupt future operations. This can occur as a consequence of a high accident rate (even if no injuries occur) or if the accident involves cultural-/societal-sensitive areas like national parks or monuments, schools, and churches. Figure 91.3 summarizes possible accidents and corresponding damages stemming from the operation of UAS in the NAS.

The following sections present models to evaluate UAS risk for the two major accident types: midair collisions and ground impact.

91.4 Ground Impact Fatality Risk Modeling

The expected frequency of fatalities (f_F) can be calculated by the following equation:

$$f_F = N_{\text{exp}} P(\text{fatality}|\text{exposure}) f_{\text{GIA}} \quad (91.1)$$

where N_{exp} is the number of people exposed to the accident, $P(\text{fatality}|\text{exposure})$ is the probability a person will suffer fatal injuries given exposure to the accident, and f_{GIA} is the rate of ground impact accidents. The $P(\text{fatality}|\text{exposure})$ depends on the resilience of the human body to injury as well as the level of sheltering.

A similar formulation to the model of (91.1) is provided in Range Safety Group, Range Commanders Council (1999b):

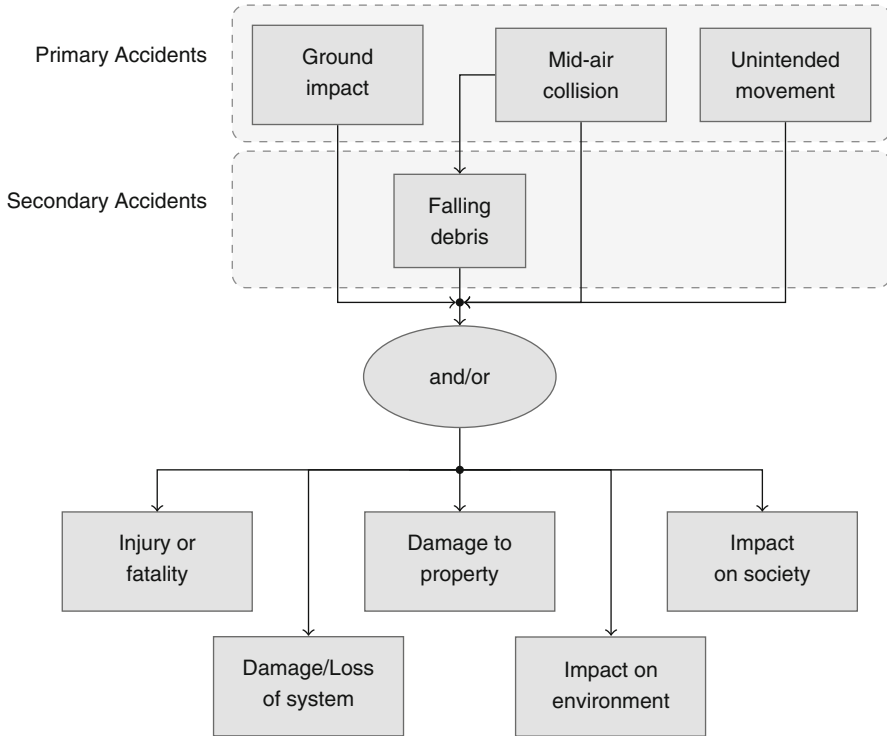


Fig. 91.3 Primary and secondary accidents that can result from the operation of UAS and their possible outcomes

$$f_F = A_{\text{exp}} \rho P(\text{fatality}|\text{exposure}) f_{\text{GIA}} f_{\text{shelter}} \quad (91.2)$$

In (91.2), N_{exp} is replaced by the product of the lethal area (A_{exp}) by the population density (ρ). An additional term, the sheltering factor f_{shelter} , is introduced that takes values from 0, denoting that everyone is sheltered from the impact, to 1, denoting that nobody is sheltered. This formulation implies a type of “absolute sheltering,” where any person considered sheltered is not affected by the impact.

In certain cases, the exposed population can be divided in groups. Each group is assigned a different probability of fatality given exposure. This can occur when, for example, part of the population is inside buildings and part is outside. In this case, (91.1) can be expressed as follows:

$$f_F = f_{\text{GIA}} \sum_i N_{i, \text{exp}} P_i(\text{fatality}|\text{exposure}) \quad (91.3)$$

where subscript i refers to the i th group. Although this approach offers better accuracy, it also requires the availability of a library that contains the number of

people in each location and the level of their sheltering or at least estimates thereof (Range Safety Group, Range Commanders Council 2007b).

Since the acceptable f_F is provided by the ELOS requirement, if the N_{exp} and $P(\text{fatality}|\text{exposure})$ can be estimated, it is possible to determine the target level of safety (TLS) for ground impact accidents given by f_{GIA} . Although (91.1) is simple and straightforward, calculation of the terms involved in it is not. In fact, there is a number of options that have been proposed for calculating both N_{exp} and $P(\text{fatality}|\text{exposure})$. The following section will start with the calculation of the f_F term based on the ELOS principle.

91.4.1 Ground Impact ELOS

In determining the fatality rate requirement after ground impacts, special consideration should be given to the fact that UAS is unmanned. This means that only the number of fatalities on the ground is to be taken into account. According to Table 91.2, this number represents only a very small percentage of the total fatalities, about 6 %. The ground fatality rate calculated is in the order of 10^{-6} h^{-1} , although a more conservative ELOS can be derived based on the ground fatality rate of air carriers, which is in the order of $f_F = 10^{-7} \text{ h}^{-1}$.

It should be noted that Table 91.2 considers all accidents. An alternative analysis can be used by considering only accidents where an in-flight collision with terrain or water occurred (approximately 35 % of the total). The updated fatality rates based on NTSB data for the period 1983–2006 are presented in Table 91.4. In this case, the proposed ELOS would be in the order of $f_F = 10^{-8} \text{ h}^{-1}$, although it does not include fatalities after emergency landings, ditching, and other situations. If the latter are included, the ELOS is closer to $f_F = 10^{-7} \text{ h}^{-1}$ as shown in Table 91.5.

For the subsequent analysis, the value for f_F is set to 10^{-7} h^{-1} , which is the same with that proposed in Range Safety Group, Range Commanders Council (2007b). However, it should be noted that lower or higher acceptable fatality rates have also been proposed in the past. In Weibel and Hansman (2004), although an ELOS of 10^{-7} h^{-1} was derived, a target of 10^{-8} h^{-1} is proposed instead. This choice was made in an effort to account for the fact that the benefits of UAS operations are not evident to the general public, and as a result, the tolerance for fatalities will be lower. In Clothier et al. (2007), analysis is based on multiple acceptable fatality likelihoods ranging from 10^{-6} to 10^{-9} h^{-1} . The Range Safety Criteria for UAS proposed a fatality rate of 10^{-6} h^{-1} or less based on the U.S. Navy survey discussed previously (Range Safety Group, Range Commanders Council 1999b), but their requirements are for military operations that allow higher fatality rates. Finally the NATO USAR adopted a TLS of 10^{-6} h^{-1} for catastrophic UAS accidents (Joint Capability Group on Unmanned Aerial Vehicles 2007), which corresponds to an equal or higher fatality rate.

Although stricter requirements may be attractive, they can seriously impede commercialization of UAS as well as their integration in the NAS.

Table 91.4 Fatality rates for accidents where an in-flight collision with terrain or water occurred. Based on analysis of NTSB accident data (National Transportation Safety Board (NTSB) [2008a](#)) between 1983 and 2006

| Rates per hour | Air carrier | Commuter | General aviation | Total |
|-------------------|----------------------|----------------------|----------------------|----------------------|
| Accident | $2.06 \cdot 10^{-7}$ | $9.33 \cdot 10^{-6}$ | $2.84 \cdot 10^{-5}$ | $1.77 \cdot 10^{-5}$ |
| Fatalities aboard | $4.71 \cdot 10^{-6}$ | $1.32 \cdot 10^{-5}$ | $2.16 \cdot 10^{-5}$ | $1.55 \cdot 10^{-5}$ |
| Ground fatalities | $9.84 \cdot 10^{-8}$ | $2.86 \cdot 10^{-8}$ | $4.46 \cdot 10^{-8}$ | $5.99 \cdot 10^{-8}$ |

Table 91.5 Fatality rates for accidents where one or a combination of in-flight collision with terrain or water, hard/forced landing, runway overrun, or ditching occurred. Based on analysis of NTSB accident data (National Transportation Safety Board (NTSB) [2008a](#)) between 1983 and 2006

| Rates per hour | Air carrier | Commuter | General aviation | Total |
|-------------------|----------------------|----------------------|----------------------|----------------------|
| Accident | $5.64 \cdot 10^{-7}$ | $1.56 \cdot 10^{-5}$ | $5.18 \cdot 10^{-5}$ | $3.21 \cdot 10^{-5}$ |
| Fatalities aboard | $4.85 \cdot 10^{-6}$ | $1.46 \cdot 10^{-5}$ | $2.41 \cdot 10^{-5}$ | $1.71 \cdot 10^{-5}$ |
| Ground fatalities | $1.01 \cdot 10^{-7}$ | $7.63 \cdot 10^{-8}$ | $8.43 \cdot 10^{-8}$ | $8.89 \cdot 10^{-8}$ |

Therefore, a conservative evaluation of the risk from emerging hazards is preferable, since it can be later accommodated as flight hours accumulate and confidence in risk estimates improves.

91.4.2 Exposure to Ground Impact Accidents

Assuming a uniform population density in the area affected by the crash, N_{exp} can be calculated as the product of that area (A_{exp}) by the population density (ρ):

$$N_{\text{exp}} = A_{\text{exp}} \times \rho \quad (91.4)$$

The population density used in [\(91.4\)](#) is typically estimated using the average population density over the area the UAS will operate. Although use of the actual population density will offer better precision, a standard population density can be used as a reasonable estimate instead. Specifically, EASA has proposed the use of a standard density of 200 ppl/km² (European Aviation Safety Agency (EASA) [2005](#)). This density was derived taking into account typical civil aviation operations, where a significant percentage of flight time is spent over less densely populated areas. For UAS designed to loiter over populated areas, a higher density will be necessary to avoid underestimating the risk involved. A worst-case scenario of impact at the most densely populated area within the area of operations may also be used to provide a conservative estimate of ρ (Range Safety Group, Range Commanders Council [1999b](#)).

There are several ways to determine the A_{exp} based on impact characteristics. For a vertical crash, this area may be approximated by the frontal area of the aircraft

augmented by a small buffer to account for the width of an average human (Weibel and Hansman 2003). For a gliding descent, it can be approximated by (91.5), where the wingspan and length of the aircraft have been increased by the radius of an average person (Clothier and Walker 2006):

$$A_{\text{exp}} = W_{\text{aircraft}} \left[L_{\text{aircraft}} + \frac{H_{\text{person}}}{\sin(\text{glide angle})} \right] \quad (91.5)$$

It should be noted that in some cases instead of the area exposed to the impact, a *casualty* or *lethal area* is mentioned. In this case, attention should be given on how these areas are defined. This is because in some cases they are the same as A_{exp} , while other times they are defined as the areas within which 100 % casualties or fatalities are expected, respectively.

For example, Range Safety Group, Range Commanders Council (2007b) defines a casualty area as the area where everyone is expected to receive injuries of such severity that they will require hospitalization. On the other hand, the same organization in Range Safety Group, Range Commanders Council (1999b) defines the lethal area merely as an *area of concern*, obtained by

$$A_{\text{exp}} = (L_{\text{aircraft}} + 2 \text{ ft}) (W_{\text{aircraft}} + 2 \text{ ft}) \quad (91.6)$$

for a vertically falling vehicle and

$$A_{\text{exp}} = (L_{\text{aircraft}} + L_{\text{glide}} + L_{\text{stop}} + 2 \text{ ft}) (W_{\text{aircraft}} + 2 \text{ ft}) \quad (91.7)$$

where L_{glide} is the gliding distance at an altitude of 6 ft and L_{stop} is the distance required for the aircraft to come to a stop.

91.4.3 Probability of Fatality of Exposed Persons

The human body is capable of sustaining a certain level of force or injury, and as a result, presence of a person in an area affected by a crash does not guarantee a fatality. Moreover, obstacles such as trees and buildings may provide shelter, thus, increasing the chances of survival. It is evident, therefore, that the probability of fatality of a person exposed to a crash need be modeled taking into account the aforementioned factors, namely, human vulnerability and sheltering. This section presents some of the approaches available. Nonetheless, a detailed account of the problem of human vulnerability is beyond the scope of this chapter.

Despite the observations above, the most commonly used estimate for the probability of fatality of an exposed person is the number one (Range Safety Group, Range Commanders Council 1999a). This is because it is a conservative measure that is not susceptible to criticism. On the other hand, it can easily be argued that this measure can be overconservative especially in the case of small UAS.

According to the Range Commanders Council, an exception may be made to the use of unit probability in the case of very light systems (Range Safety Group, Range Commanders Council 1999b). In this case, a limit must be defined that divides UAS into two categories, those that do and those that do not cause fatalities. The limit can be based on that used for inert debris, specifically a kinetic energy during impact of 15 J (11 ft lb) for casualty (i.e., reversible injury) and 34 J (25 ft lb) for fatality (Range Safety Group, Range Commanders Council 2007b). The 15 J criterion has been found to be close to the threshold of serious injury for a child (Haber and Linn 2005), and, as a result, it can be stated that it is a conservative measure that protects the entire population.

In an effort to take into account the fact that a person may survive a UAS impact, Weibel et al. introduced a penetration factor for calculating the probability of fatality (Weibel and Hansman 2004). This factor depends on the characteristics of the UAS and aims at taking into account sheltering. Dalamagkidis (2010) argued that Weibel's estimate for smaller vehicles is overconservative while at the same time it underestimates the lethality of larger systems. It was also mentioned that the usefulness of the "penetration parameter" was limited since no method was provided to consistently estimate it.

In a 1968 study, Feinstein et al. investigated the effects of blast, debris, and other factors to people (Feinstein et al. 1968). To correlate the probability of fatality with kinetic energy, log-normal curves were derived for impacts to different body parts. Additionally, an averaged curve was proposed, given by

$$P(\text{fatality}|\text{exposure}) = Z\left(\frac{\ln E_{\text{imp}} - \ln a}{b}\right) \quad (91.8)$$

where Z is the cumulative normal distribution whose value is typically obtained from tables, $a = 103 \text{ J}$ is the energy required for a probability of fatality of 50 %, and $b = 0.538$ is a parameter that affects how fast the probability rises as a function of kinetic energy.

A drawback of Feinstein's model is that it is difficult to adjust it to take into account sheltering. This is because if parameter b was increased so that the curve slope is decreased, the probability of fatality for low kinetic energies would increase. As a result, any change of the b parameter to take into account sheltering would require an adjustment of the a parameter as well.

A model to estimate $P(\text{fatality}|\text{exposure})$ as a function of kinetic energy at impact (E_{imp}) that also takes into account sheltering in addition to human vulnerability was proposed in Dalamagkidis et al. (2008) and then revised in Dalamagkidis et al. (2012). The model is a variation of the logistic growth model that aligns well with other human vulnerability models including models such as Feinstein's. That model is given by

$$P(\text{fatality}|\text{exposure}) = \frac{1 - k}{1 - 2k + \sqrt{\frac{\alpha}{\beta}} \left[\frac{\beta}{E_{\text{imp}}} \right]^{\frac{3}{p_s}}} \quad (91.9)$$

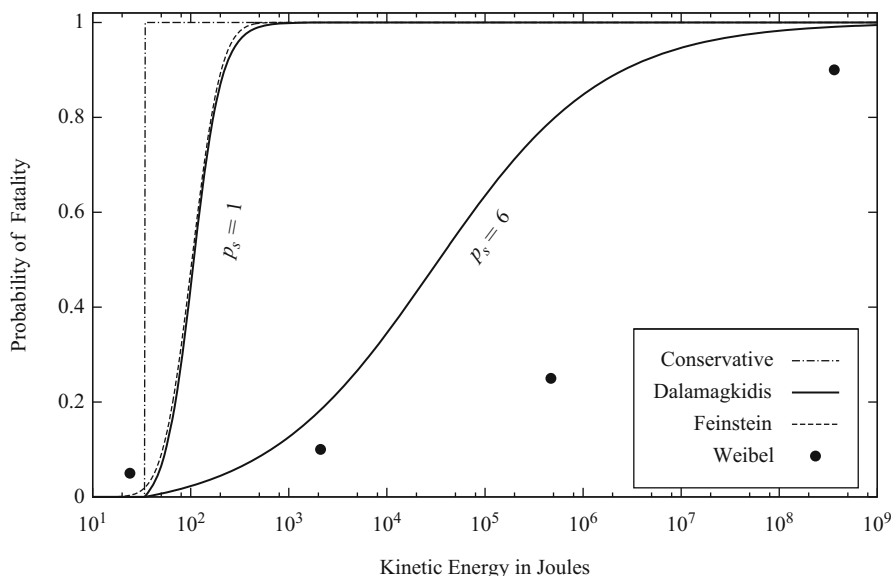


Fig. 91.4 A comparison of the vulnerability models of Feinstein, Dalamagkidis, and Weibel as well as the conservative approach of unit probability above the 34 J kinetic energy threshold

where $k = \min \left[1, \left(\frac{\beta}{E_{\text{imp}}} \right)^{\frac{3}{p_s}} \right]$ is a correction factor k , used to improve the estimates given for low kinetic energies, especially those close to, or below, the threshold limit of 34 J.

The sheltering parameter p_s determines how exposed is the population to an impact and takes values in the range $(0, \infty)$. It is a function of the amount of obstacles in the crash trajectory of the aircraft that can absorb impact energy or deflect debris as well as the ability of people to take shelter behind such obstacles. It takes an average value of 1, with higher values, meaning better sheltering and a lower probability of fatality for the same kinetic energy. The α parameter is the impact energy required for a fatality probability of 50% when $p_s = 6$. Finally the β parameter is the impact energy threshold required to cause a fatality as p_s goes to zero. Based on the fatality limit of Range Safety Group, Range Commanders Council (2007b), the β parameter can be considered to be a constant with value 34 J.

The fatality probability models presented in this section are compared in Fig. 91.4.

91.4.3.1 Kinetic Energy Estimation

Many human vulnerability models involve the kinetic energy at impact or other functions of impact speed and object mass. This speed may vary depending on the UAS and the descent characteristics, and as a result, a suitable estimate must be

used. A useful conservative substitute for the impact speed is terminal velocity. The latter can be calculated from (91.10), where m is the vehicle mass, g is the acceleration of gravity, ρ_α is the air density, A is the cross-sectional area of the vehicle, and C_d is its drag coefficient. The latter two parameters are not always available, since they vary with the orientation of the aircraft during a descent:

$$E_{\text{imp}} = \frac{m^2 g}{\rho_\alpha A C_d} \quad (91.10)$$

The use of the maximum between the terminal velocity and the *velocity not to exceed* provided by the manufacturer is proposed as an alternative in Range Safety Group, Range Commanders Council (1999b).

In Haddon and Whittaker (2002), Joint JAA/Eurocontrol Initiative on UAVs (2004), and European Aviation Safety Agency (EASA) (2005), instead of the terminal velocity, the use of the maximum operating velocity (v_{op}) increased by 40 % is proposed, instead. This choice overcomes the problem of accurately estimating the parameters required to calculate the terminal velocity, greatly simplifying calculations. The kinetic energy can then be calculated as

$$E_{\text{imp}} \approx m v_{\text{op}}^2 \quad (91.11)$$

It may also be argued that when the mass of the impacting object is comparable or larger than that of the body part struck, not all of the object's kinetic energy will be absorbed. In fact, after the impact, the object will continue to move, in unison with the body, retaining some kinetic energy. The energy absorbed during the collision can be calculated based on momentum conservation by Sturdivan et al. (2004):

$$E = \frac{1}{2} m_1 v_1^2 \left(1 - \frac{m_1}{m_1 + m_2} \right) \quad (91.12)$$

where m_1 and v_1 are the mass and velocity of the object and m_2 refers to the effective mass of the body part struck. As a result, when $m_1 \ll m_2$, the effective energy is equal to the kinetic energy of the projectile. It should be noted that when body movement is constrained (e.g., from a wall), then (91.12) no longer applies and the entire kinetic energy is to be used (Sturdivan et al. 2004).

Irrespective of which of the aforementioned methods is used to calculate the kinetic energy of the impacting object, if a person is sheltered within a building or a vehicle, some of the kinetic energy will be exhausted to penetrate the shelter. As a result, the energy used in the vulnerability model may be also adjusted to take into account the effects of sheltering. The following section provides a more detailed account of the factors that need to be considered when incorporating the effects of sheltering, either by kinetic energy adjustments or by estimating the value of a parameter in the vulnerability model.

91.4.3.2 Sheltering Estimation

In certain situations, even when some sheltering is available, a conservative estimate of no sheltering may be used (Range Safety Group, Range Commanders Council 1999b). Nevertheless, this could lead to overconservative results, especially when a significant percent of the population is protected by vehicles and buildings. When estimating the sheltering effect, all the contributing factors need to be evaluated. These factors include the position of the body (standing, sitting, or prone), the sheltering provided by buildings or other structures, and the level of danger awareness.

Body position is normally taken into account in the vulnerability model. This is typically achieved by averaging the injury expectations calculated for different body positions. It should be noted that even when a specific area implies that the majority of the exposed population will be in a specific position (e.g., prone for a beach), using just the prone position parameters may be too conservative (Cole et al. 1997).

Buildings, vehicles, and other objects may reduce the risk of injury by absorbing part of or even the entire impact energy. For buildings the kinetic energy required to penetrate the structure needs to be considered (Range Safety Group, Range Commanders Council 2007b). This energy can be as low as 23 J for a roof made by a sheet of 24 gage corrugated aluminum and can exceed 560 J for roofs made of light concrete on 22 gage corrugated steel decking (Range Safety Group, Range Commanders Council 2007b). Although the examples given concern vertical impacts, in most cases, side impacts will also need to be considered (Range Safety Group, Range Commanders Council 2007b). Additionally, if an object manages to penetrate a structure, it is possible for material from the damaged structure to fall, causing secondary hazards (Range Safety Group, Range Commanders Council 2007b).

It should be noted that the aforementioned limits concern an irregular, compact, tumbling steel fragment (Range Safety Group, Range Commanders Council 2007b). As a result, a UAS may require additional energy to penetrate the same roof. This is because they are constructed from lighter, less dense materials, because their kinetic energy is going to be distributed over a larger area and because some energy may also be expended for fragmentation instead of penetration. If the types of structures in an area vary significantly, it is possible to group buildings into different classes and apply the approach of (91.3), assuming that the distribution of the population within the buildings is known or can be estimated. Accurate prediction of the effect of sheltering with the aforementioned methodology is difficult. This is because it requires precise knowledge regarding the construction of all structures in the affected area as well as the people and their activities during impact. To overcome this limitation, it is generally acceptable to make estimates based on average or worst-case scenarios. Compatible with the worst-case scenario is to assume that all buildings offer the sheltering effect of the worst buildings.

Besides the approach of subtracting the energy required to penetrate the structure from the energy available to cause a fatality that was mentioned in the previous section, a more conservative approach is also possible. The latter assumes that if an

object has sufficient energy to penetrate a structure, everyone inside is going to be a casualty or fatality (Range Safety Group, Range Commanders Council 2007b).

Danger awareness may also significantly improve the odds of a person surviving a UAS impact. This is because a person that is aware of the UAS can react sooner and either avoid the UAS entirely or take better shelter. To account for this, as well as the fact that the risk is taken on a voluntary basis, it has been proposed to use a larger allowable risk of fatality or casualty (Range Safety Group, Range Commanders Council 2007b). On the other hand, it is unlikely that people other than the UAS operating crew will have that benefit.

91.4.4 Frequency of Ground Impact Accidents

The frequency of ground impact accidents (f_{GIA}) is a function of the reliability of the UAS. If a limit for it is known or estimated from (91.1), then the UAS can be designed to meet set TLS. For example, in Range Safety Group, Range Commanders Council (2007b), a limit of 10^{-7} aircraft impacts per mission is proposed that could be used to obtain a limit for f_{GIA} . Alternatively, if the actual f_{GIA} is known, it can be used to determine whether an existing system with known reliability would violate safety limits.

The actual f_{GIA} can be estimated from the history of the UAS itself, provided that a sufficient number of flight hours have accumulated. If the UAS has a long history, the most recent data are to be preferred, since older data may not represent its current performance (Range Safety Group, Range Commanders Council 1999b).

An issue arises with newer systems, with few flight hours that may have not exhibited a failure that led to an accident yet. In this case, if the UAS is based on a previous model and shares a significant amount of design characteristics, its reliability can be assumed to be the same with that of the previous model (Range Safety Group, Range Commanders Council 1999b). Otherwise, the f_{GIA} can be estimated using Table 91.6 (Range Safety Group, Range Commanders Council 1999b). The aforementioned table was derived under the assumption of stochastic system behavior, exponential failure distribution and no significant changes in the system or its environment. As a result, care should be taken when using Table 91.6, especially in the case of UAS that are experimental or still in development, since these assumptions may not hold. In that case or when there is a need for a conservative estimate, worst-case assumptions can be made. Such assumptions may take the form of one crash per flight or per flight hour (Range Safety Group, Range Commanders Council 1999b).

If a formal reliability assessment has been carried out for the particular UAS model that included a Failure Mode and Effects Analysis (FMEA) or similar study, then its conclusions can also be used to estimate f_{GIA} (Range Safety Group, Range Commanders Council 1999b). That estimate may be increased by a safety margin to account for failure modes that may have been overlooked or underestimated.

Table 91.6 Estimate of the expected frequency of ground impact accidents for new UAS. The value shown represents a confidence of 95 % that the actual f_{GIA} is less than or equal to \hat{f}_{GIA} (Source: Range Safety Group, Range Commanders Council (1999b))

| Flight hours without crash | \hat{f}_{GIA} |
|----------------------------|----------------------------------|
| 10 | $3 \cdot 10^{-1} \text{ h}^{-1}$ |
| 30 | $1 \cdot 10^{-1} \text{ h}^{-1}$ |
| 100 | $3 \cdot 10^{-2} \text{ h}^{-1}$ |
| 300 | $1 \cdot 10^{-2} \text{ h}^{-1}$ |

91.5 Midair Collision Fatality Risk Modeling

The expected frequency of fatalities (f_F) following midair collision accidents can be calculated by the following equation:

$$f_F = E(\text{fatality}|\text{collision}) f_{\text{MaC}} \quad (91.13)$$

$$= N_{\text{exp}} P(\text{fatality}|\text{collision}) P(\text{collision}|\text{CT}) f_{\text{CT}} \quad (91.14)$$

where $E(\text{fatality}|\text{collision})$ is the expected number of fatalities after a crash and f_{MaC} is the rate of midair collision accidents. The former can be calculated as the product of the number of people exposed to the accident (N_{exp}) by the probability a person will suffer fatal injuries given exposure to the accident ($P(\text{fatality}|\text{collision})$). The f_{MaC} term can be calculated as the product of the probability of collision given that two aircraft are in conflicting trajectories ($P(\text{collision}|\text{CT})$) by the frequency of occurrence of conflicting trajectories (f_{CT}).

As in the case of ground impact accidents, since the acceptable f_F is provided by the ELOS requirement, it is possible to determine the target level of safety (TLS) for midair collision accidents given by f_{MaC} . Moreover, if the actual frequency of occurrence of conflicting trajectories is known from historical data, the maximum acceptable value of $P(\text{collision}|\text{CT})$ can be determined, which maps directly to the capabilities of the sense and avoid system.

91.5.1 Midair Collision ELOS

In order for an ELOS to be derived, accident statistics involving midair collisions are required. The NTSB has defined two categories of relevant accidents: in-flight collisions with obstacles like birds, trees, and power lines and midair collisions with other aircraft. The results from the analysis of NTSB data involving these two accidents are tabulated in Tables 91.7–91.9.

In contrast to the ground impact accidents, the ELOS for midair collision accidents should include the total number of fatalities. This is because onboard fatalities are possible when the accident is between a UAS and a manned aircraft.

Table 91.7 Fatality rates for accidents where an in-flight collision with obstacles (e.g., birds, trees, power lines) occurred. Based on analysis of NTSB accident data (National Transportation Safety Board (NTSB) 2008a) between 1983 and 2006

| Rates per hour | Air carrier | Commuter | General aviation | Total |
|-------------------|----------------------|----------------------|----------------------|----------------------|
| Accident | $1.34 \cdot 10^{-7}$ | $3.22 \cdot 10^{-6}$ | $1.33 \cdot 10^{-5}$ | $8.17 \cdot 10^{-6}$ |
| Fatalities aboard | $9.67 \cdot 10^{-7}$ | $2.67 \cdot 10^{-6}$ | $6.27 \cdot 10^{-6}$ | $4.25 \cdot 10^{-6}$ |
| Ground fatalities | $5.97 \cdot 10^{-9}$ | $3.81 \cdot 10^{-8}$ | $5.73 \cdot 10^{-8}$ | $3.93 \cdot 10^{-8}$ |
| Total fatalities | $9.73 \cdot 10^{-7}$ | $2.71 \cdot 10^{-6}$ | $6.32 \cdot 10^{-6}$ | $4.29 \cdot 10^{-6}$ |

Table 91.8 Fatality rates for accidents where a midair collision with another aircraft occurred. Based on analysis of NTSB accident data (National Transportation Safety Board (NTSB) 2008a) between 1983 and 2006

| Rates per hour | Air carrier | Commuter | General aviation | Total |
|-------------------|-------------|----------------------|----------------------|----------------------|
| Accident | None | $2.76 \cdot 10^{-7}$ | $5.90 \cdot 10^{-7}$ | $3.74 \cdot 10^{-7}$ |
| Fatalities aboard | None | $6.96 \cdot 10^{-7}$ | $1.04 \cdot 10^{-6}$ | $6.82 \cdot 10^{-7}$ |
| Ground fatalities | None | $1.91 \cdot 10^{-8}$ | $2.86 \cdot 10^{-8}$ | $1.87 \cdot 10^{-8}$ |
| Total fatalities | None | $7.15 \cdot 10^{-7}$ | $1.07 \cdot 10^{-6}$ | $7.01 \cdot 10^{-7}$ |

Table 91.9 Fatality rates for accidents where either a midair collision with an object or another aircraft occurred. Based on analysis of NTSB accident data (National Transportation Safety Board (NTSB) 2008a) between 1983 and 2006

| Rates per hour | Air carrier | Commuter | General aviation | Total |
|-------------------------------|----------------------|----------------------|----------------------|----------------------|
| Accident | $1.34 \cdot 10^{-5}$ | $3.48 \cdot 10^{-6}$ | $1.38 \cdot 10^{-5}$ | $8.53 \cdot 10^{-6}$ |
| Total fatalities | $9.73 \cdot 10^{-7}$ | $3.42 \cdot 10^{-6}$ | $7.40 \cdot 10^{-6}$ | $4.99 \cdot 10^{-6}$ |
| Total fatalities ^a | $5.97 \cdot 10^{-9}$ | $7.53 \cdot 10^{-7}$ | $1.13 \cdot 10^{-6}$ | $7.40 \cdot 10^{-7}$ |

^aExcluding fatalities aboard after collisions with objects other than aircraft

From the NTSB accident data in Table 91.9, it can be argued that the fatality rate following midair collisions with aircraft or other obstacles is in the order of $f_F = 10^{-6} \text{ h}^{-1}$. A more conservative estimate of $f_F = 10^{-7} \text{ h}^{-1}$ can be reached from the same table, if the onboard fatalities after a collision with obstacles other than aircraft are ignored. By deriving the expected number of fatalities after a midair collision accident, it is then possible to determine the maximum acceptable frequency of such accidents.

Another approach is to assume that in the case of midair collisions, the fatality expectation is the same, regardless of whether a UAS was involved in the accident. Although this assumption is more conservative, it simplifies subsequent analysis, since one may directly obtain the accident TLS for midair collisions. Based on the NTSB data of Table 91.9, the average rate of midair collisions involving manned aircraft is $7.40 \cdot 10^{-7}$, and under ELOS requirements, a maximum midair collision rate of $f_{\text{MaC}} = 10^{-7} \text{ h}^{-1}$ can be proposed for UAS.

Table 91.10 Maximum acceptable accident frequency depending on ATC type, flight phase, and aircraft threatened. The collision accident criteria to be applied corresponds to the one for the highest category of aircraft threatened (Source: INnovative Operational UAS Integration (INOUI) (2009))

| ATC type | Flight phase | >2,730 kg | MEP/SET <2,730 kg | SEP <2,730 kg |
|--------------|---------------------|-------------------|-------------------|-------------------|
| Area control | En route inbound | $3 \cdot 10^{-9}$ | $3 \cdot 10^{-8}$ | $3 \cdot 10^{-7}$ |
| | En route outbound | $1 \cdot 10^{-9}$ | $1 \cdot 10^{-8}$ | $1 \cdot 10^{-7}$ |
| | En route transit | $3 \cdot 10^{-9}$ | $3 \cdot 10^{-8}$ | $3 \cdot 10^{-7}$ |
| Approach | Departure | $1 \cdot 10^{-9}$ | $1 \cdot 10^{-8}$ | $1 \cdot 10^{-7}$ |
| | Init and interm app | $3 \cdot 10^{-9}$ | $3 \cdot 10^{-8}$ | $3 \cdot 10^{-7}$ |
| | Final approach | $3 \cdot 10^{-9}$ | $3 \cdot 10^{-8}$ | $3 \cdot 10^{-7}$ |
| Tower | Landing | $8 \cdot 10^{-9}$ | $8 \cdot 10^{-8}$ | $8 \cdot 10^{-7}$ |
| | Line-up | $3 \cdot 10^{-9}$ | $3 \cdot 10^{-8}$ | $3 \cdot 10^{-7}$ |
| | Start-up/push-back | $8 \cdot 10^{-9}$ | $8 \cdot 10^{-8}$ | $8 \cdot 10^{-7}$ |
| | Takeoff | $8 \cdot 10^{-9}$ | $8 \cdot 10^{-8}$ | $8 \cdot 10^{-7}$ |
| | Taxiing | $6 \cdot 10^{-8}$ | $6 \cdot 10^{-7}$ | $6 \cdot 10^{-6}$ |

MEP, multiengine piston; SET, single-engine turbine; SEP, single-engine piston

Other accident frequency limits proposed for UAS take into account both the manned aircraft threatened and the phase of flight. Such an approach was taken by INOUI that proposed the limits presented in Table 91.10.

Finally, it should be noted that not all collisions lead to catastrophic accidents. The large variability of aircraft sizes and designs, whether manned or unmanned, and the fact that not all their systems are critical for remaining airborne, means that certain collisions may be survived by one or even both of the aircraft involved. Nevertheless, since it is nearly impossible to account for every possible collision scenario and its effects, every collision is considered a catastrophic accident for both aircraft.

91.5.2 Exposure and Risk of Fatality

The number of people exposed to the accident, as well as the probability of them sustaining fatal injuries, depends on the aircraft that are involved in the accident and the passengers they carry. As a result, it is difficult to get a good estimate without a priori knowledge of all air traffic in the area of operations.

A more general estimate can be derived by noting that the product of N_{exp} and $P(\text{fatality}|\text{collision})$ is in fact the expected number of fatalities per accident. Using the NTSB accident data of Table 91.9, this product is higher for commuter aviation where it takes a value of one, while on average it is closer to 0.58. Moreover, if the onboard fatalities after a collision with obstacles other than aircraft are ignored, the expected number of fatalities per accident drops to below 0.09. It should be noted that this estimate can be considered conservative because in contrast with the accident data it was derived from, the midair collisions of interest will always involve at least one aircraft that is unoccupied.

91.5.3 Conflicting Trajectory Expectation

In Weibel and Hansman (2004), the midair collision risk assessment was based on the use of a gas model of aircraft collisions to estimate the number of expected collisions per hour of flight (f_{MaC}) from

$$f_{\text{MaC}} = \frac{A_{\text{exp}} d}{V \times t} \quad (91.15)$$

where A_{exp} is the exposed area of the threatened aircraft, d is the distance traveled, V is the airspace volume, and t is the time required to travel the distance d .

It should be noted that this model estimates the number of midair collision hazards due to insufficient spatial and temporal separation given predetermined flight paths or simply the number of *potential* collisions. An additional term is then required to take into account the fact that one or both of the aircraft in a collision course may attempt maneuvers to avoid each other. As a result, the expected number of collisions should be calculated from

$$f_{\text{MaC}} = \underbrace{\frac{A_{\text{exp}} d}{V \times t}}_{E(\text{CT})} \times P(\text{collision}|\text{CT}) \quad (91.16)$$

where CT denotes a conflicting trajectory.

The use of the model in (91.16) to assess $E(\text{CT})$ presents significant difficulties since it requires the exact trajectories (both in space and time) of all air traffic in the area where UAS operations will take place. This requirement is almost impossible to meet because air traffic is dynamic and never identical from day to day. Furthermore, not all traffic is monitored by ATC. In addition to that, in the event of a deviation from the predefined trajectory, the number of collision hazards following that event may change. Thus, a worst-case $E(\text{CT})$ may be assumed, instead.

Based on the analysis in Weibel and Hansman (2004), high $E(\text{CT})$ is found in proximity of major airways. The highest was found at FL370, where it is approximately 4×10^{-5} CT/h. Since the results were obtained by averaging data over a 24-h period, a process that can hide higher peaks, a worst-case $E(\text{CT}) = 10^{-4}$, or even higher can be chosen to also account for future traffic growth.

91.5.4 Collision Probability

Even when two aircrafts are on conflicting trajectories, a collision is not guaranteed. One or both of the pilots may take action to avoid a hazardous situation by maintaining the required separation between the two aircrafts. As a result, the collision probability depends on the collision avoidance capabilities of all the aircraft involved as well as the measures taken to assure proper separation. If a maximum allowable collision probability is known, it can be used instead of

the maximum acceptable accident frequency to determine minimum performance requirements of UAS sense and avoid systems. In fact its use offers advantages because it maps better to what such a system should be capable of accomplishing. To evaluate the actual $P_{\max}(\text{collision}|\text{CT})$, every possible scenario involving a potential conflicting trajectory can be simulated. The results are then aggregated and tested to see whether the target safety levels are accomplished.

It should be noted that when evaluating the sense and avoid system, the worst-case $E(\text{CT})$ may be overconservative. This is because it corresponds to Class A airspace where separation is provided by and is the responsibility of ATC. In the analysis presented in Weibel and Hansman (2004), the worst-case conflicting trajectory expectation falls by about an order of magnitude in Class E airspace. If the target safety level is assumed to be a collision rate of 10^{-7} h^{-1} , then the “see and avoid” capability requirement may be based on achieving $P_{\max}(\text{collision}|\text{CT}) = 1\%$. Nevertheless, the same cannot be assumed for Class G airspace because traffic in that region is not always monitored and accurate estimates on its density are not possible. In addition to that, and especially in very low altitudes, the risk of collision with birds, power lines, trees, and buildings may be higher than that of a collision with other air traffic.

91.6 Model Choice

Tables 91.11 and 91.12 summarize the parameters involved in modeling risk from UAS ground impact and midair collision accidents, respectively, as well as the alternatives presented for estimating their values. It is evident that there are several choices available to an engineer tasked with assessing the risk of UAS operations. The subject of this section is what criteria should drive the selection of one alternative over another.

There is a multitude of modeling options available to estimate risk, each with different levels of detail and accuracy. According to Range Safety Group, Range Commanders Council (2007b), any model used for risk/reliability assessment should be based on four basic standards: transparency, clarity, consistency, and reasonableness.

A number of different models may be used in a risk evaluation involving UAS operations depending on the objective and requirements of the risk study in question. A common use of risk models in the UAS domain is for building a safety case that is then used for obtaining a permit to operate in the national airspace system. As a result, the models used must be clearly presented so that they can be reviewed by the regulators, and the assumptions and limitations contained therein must be succinctly expressed. This would lead to compliance with the clarity and transparency standards.

Typically, the regulatory framework does not specify the use of a particular modeling choice over another. Nevertheless, every choice must be defensible. This is achieved by compliance with the consistency and reasonableness standards. The former refers to the use of models that are in use and accepted by the scientific

Table 91.11 A summary of the methods presented to estimate the terms involved in ground impact risk modeling when their values are not known a priori, Eq. (91.1). Some terms can be estimated with multiple methods

| Term | Estimate |
|----------------------|---|
| f_F | Based on ELOS requirements with typical values in the range of 10^{-6} – 10^{-9} h ⁻¹ |
| N_{exp} | The product of the population density (ρ) and the area affected by the impact (A_{exp}) |
| ρ | <ul style="list-style-type: none"> (i) Assuming uniform population density (ii) Using a standard population density (e.g., 200 ppl/km²) (iii) Assuming a worst-case scenario of impact at the most densely populated area |
| A_{exp} | <ul style="list-style-type: none"> (i) The area presented by the aircraft perpendicular to its path and augmented by the width of an average person (ii) The aforementioned area, including the area the aircraft traverses on the ground until it stops |
| $P(\text{fatality})$ | <ul style="list-style-type: none"> (i) Probability of one as a conservative estimate (ii) Zero or one, based on whether the kinetic energy at impact (E_{imp}) exceeds a pre-specified threshold (e.g., 34 J) (iii) From a vulnerability model based on kinetic energy at impact (E_{imp}), e.g., Feinstein et al. (1968) (iv) From a vulnerability model that also includes the effects of sheltering, e.g., Weibel (2005) or Dalamagkidis et al. (2012) |
| E_{imp} | <ul style="list-style-type: none"> (i) Kinetic energy at terminal velocity (ii) Kinetic energy at VNE (velocity not to exceed) (iii) Kinetic energy at 140 % operational velocity (iv) The difference between kinetic energy at impact and kinetic energy remaining post-impact (combined human/object) (v) The kinetic energy calculated with one of the aforementioned means, reduced by the energy required to penetrate sheltering |
| Sheltering | <ul style="list-style-type: none"> (i) Using a conservative value that assumes little or no sheltering (ii) Based on average sheltering provided by structures, vehicles, and other objects (iii) Based on the building with the worst sheltering (iv) Using a database containing structure characteristics and population distribution |
| f_{GIA} | <ul style="list-style-type: none"> (i) From previous accident statistics, if sufficient flight hours have accumulated (ii) Assuming an exponential accident distribution for new vehicles without accidents so far (iv) Using a conservative estimate of one crash per flight or per flight hour (v) Based on the results of a formal UAS reliability assessment |

community. The latter means that model selection should be based on rational criteria, risk is not underestimated, and a potential review would not raise concerns. As a result, the easier and in fact a common approach is to make conservative estimates.

Conservative estimates are also attractive because of their simplicity and of the associated ease of achieving clarity and transparency. In general, even when high fidelity modeling is possible, a balance must be struck between precision

Table 91.12 A summary of the methods presented to estimate the terms involved midair collision risk modeling when their values are not known a priori, Eqs. (91.13) and (91.14)

| Term | Estimate |
|-----------------|--|
| f_F | (i) Based on ELOS requirements with typical values in the range of 10^{-6} – 10^{-8} h ⁻¹ (ii) As above but considering only fatalities on the ground for calculating ELOS (valid when the accident involves only UAS) |
| E (fatality) | (i) Estimated from historical data, e.g., based on NTSB data from Table 91.9, it takes values in the range 0.02–1 (ii) Estimated from the product of N_{exp} and P (fatality) |
| N_{exp} | The number of people onboard the aircraft involved in the collision as well as in the area exposed to debris |
| P (fatality) | (i) Use of the number one as a conservative estimate (ii) Estimate from historical data, if available |
| f_{MaC} | (i) From historical data with typical values in the range of 10^{-5} – 10^{-7} h ⁻¹ (ii) As the product of P (collision) with f_{CT} |
| P (collision) | (i) A conservative estimate of probability one (ii) Estimated based on the capabilities of the S & A system |
| f_{CT} | (i) Based on the gas model of aircraft collisions using actual traffic data (ii) Using worst-case air traffic density either at the flight level of operations or the entire airspace |

and limitations relevant to cost, resources, and time (Range Safety Group, Range Commanders Council 2007b).

On the other hand, conservative estimates may lead to irrationally high reliability requirements and/or very strict operational restrictions. This occurs as a consequence of a problem known as compounding conservatism, where use of successive conservative estimates can lead to overconservative results (Range Safety Group, Range Commanders Council 2007b). To illustrate this issue, consider the modeling of a ground impact accident scenario using (91.1). In addition, assume that the actual population and area affected by the crash are overestimated by 50 and 20 %, respectively, and the probability of fatality is considered to be 1 when in fact it is only 20 %. In this case, the fatality expectation will be nine times higher than what it really is. Moreover, if conservative estimates are used for evaluating the possible hazards that may lead to a crash, the reliability requirements for the various aircraft parts can be higher by two or more orders of magnitude. Although such an aircraft would be capable of performing well within the target safety levels, the higher production and maintenance costs could mean that it may never get built.

The problem of compounding conservatism may be addressed by avoiding to take conservative estimates and, instead, opting for the best available estimate (Range Safety Group, Range Commanders Council 2007b). Such an approach has also been advocated by the U.S. Nuclear Regulatory Commission (Range Safety Group, Range Commanders Council 2007b). Of course, when best or mean estimates are used, the uncertainties and possible inaccuracies affecting the final result should be clearly documented (Range Safety Group, Range Commanders Council 2007b).

This would also allow adjustments at a later stage to account for newer data as they become available. Moreover, if the uncertainties can be quantified through simulation, sensitivity analysis, or other methods, then confidence intervals can be determined. The latter can then be used to adjust results so that proposed reliability targets can be achieved with arbitrary confidence.

As an example, it will be assumed that a probability of fatality of 0.4 was estimated for a particular accident scenario. Further analysis of the model showed that for a confidence level of 95 %, the actual value is expected to be between 0.3 and 0.5. In this case, the 0.5 value may be used instead of 0.4 to determine UAS reliability requirements. If higher confidence is required, the range will be larger and the value used more conservative.

In certain cases – especially when required data are missing – a safety case can be made by comparing the UAS under investigation with a different system already authorized to fly (Range Safety Group, Range Commanders Council 1999b). Qualitative arguments may also be made without a complete analysis of the risk involved (Range Safety Group, Range Commanders Council 1999b). For example, the UAS may be too light to cause an injury or the area of operations may be so sparsely populated that the risk to the general public is too low under any conditions.

The modeling techniques described in the previous sections produce an average risk estimate from the operation of a UAS, especially when best estimates are used instead of conservative. What should be mentioned is that in certain cases when using a casualty or fatality metric, particularly catastrophic accidents involving multiple fatalities can occur without violating the target safety levels on average (Range Safety Group, Range Commanders Council 2007b). Such accidents of course are best to be avoided, and as a result, it is useful to incorporate catastrophe aversion in the models used (Range Safety Group, Range Commanders Council 2007b). The latter is accomplished by assigning activities that can be particularly dangerous to lower acceptable probabilities of occurrence. This in turn may be done either by creating a risk profile if sufficient data are available or easier by using functions of the N_{exp}^k type for expressing the affected population (Range Safety Group, Range Commanders Council 2007b). Of course the latter methodology should be employed with care and for high risk activities only, so as not to unnecessarily inflate the risk in other less dangerous activities. A risk profile gives the function between the number of expected casualties and the expected frequency for each casualty size due to various future incidents (Range Safety Group, Range Commanders Council 2007b). Although this profile is useful for obtaining a better view of the associated risks, it is usually costly and time consuming to obtain.

91.7 Case Studies

Using the methodologies described in Sect. 91.4, it is possible to derive the reliability requirements with respect to ground impact for various types of UAS and under different scenarios.

Table 91.13 Characteristics of five UAS of various sizes, used for the case analysis (Source: FSF editorial staff (2005) and U.S. Department of Defense. Office of the Secretary of Defense (2005))

| | Weight (kg) | Dimensions (m) | Oper. speed (m/s) | Oper. altitude (ft) |
|------------------|-------------|-----------------------|-------------------|---------------------|
| RQ-4 Global Hawk | 11, 612 | 35.4 (wingspan) | 177 | 65, 000 |
| MQ1 Predator | 1, 021 | 14.8 (wingspan) | 70 | 20, 000 |
| RQ-2 Pioneer | 205 | 5.2 (wingspan) | 41 | 15, 000 |
| RQ-11 Raven | 1.9 | 1.3 (wingspan) | 15 | 1, 000 |
| Rmax IIG | 94 | 3.12 (rotor diameter) | 5.6 | 500 |

Table 91.14 The parameters used for each test case and a description of a possible corresponding scenario

| Scenario | Pop. density (ppl/km ²) | p_s | Description |
|-----------------|-------------------------------------|-------|---|
| 1 – Optimistic | 50 | 7 | Low population density area. It is also assumed that people are afforded significant sheltering either by natural obstacles (e.g., trees) or they can be trained to avoid or take cover when required. This scenario may correspond to surveillance of a remote military installation or to a forest monitoring application |
| 2 – Pessimistic | 5,000 | 1 | This scenario features very high population density. Additionally the sheltering factor used corresponds to no protection from sheltering at all. This case corresponds to the scenario of a search and rescue operation in a metropolitan area, where several people are in open areas preoccupied with other tasks |

To illustrate the differences between vulnerability models, an optimistic and a pessimistic scenario were investigated using five UAS. The systems were chosen to span all sizes, and their basic characteristics are shown in Table 91.13. A description of the scenarios and the parameters used are provided in Table 91.14. Although the results are subject to the uncertainties inherent in the parameters and the models themselves, they should be accurate in terms of order of magnitude and for comparisons between different vulnerability models as well as between UAS types and sizes.

For each case, the probability of fatality was first estimated. To illustrate the different results that may be reached due to the choice of a model, three different modeling options were employed:

1. The probability of fatality was based on the vulnerability model of (91.9) using a kinetic energy estimate based on (91.11).
2. Using the same model as above, but with a more conservative kinetic energy estimate based on terminal velocity.
3. The probability of fatality is 0 or 1, based on the kinetic energy threshold of 34 J, initially presented in Sect. 91.4.3. The kinetic energy is calculated using the terminal velocity.

Table 91.15 Fatality probability with respect to ground impact accidents for five UAS under the pessimistic scenario. Four different means of obtaining the fatality probability are used

| UAS model | <i>P</i> (fatality exposure) | | |
|------------------|--------------------------------|--------------------------------|---------------------------------|
| | Eq. (91.9) ^a (%) | Eq. (91.9) ^b (%) | 34 kJ limit ^b (%) |
| RQ-4 Global Hawk | 100.0 | 100.0 | 100.0 |
| MQ1 Predator | 100.0 | 100.0 | 100.0 |
| RQ-2 Pioneer | 100.0 | 100.0 | 100.0 |
| RQ-11 Raven | 97.4 | 100.0 | 100.0 |
| Rmax type IIG | 100.0 | 100.0 | 100.0 |

^aUsing vehicle kinetic energy estimated from (91.11)

^bWorst-case vehicle kinetic energy estimate

Table 91.16 Reliability requirement for five UAS with respect to ground impact accident under the pessimistic scenario. Five different fatality probability estimates are used

| UAS model | Required time between ground impact accidents in hours | | | |
|------------------|--|-------------------------|--------------------------|--------------|
| | Eq. (91.9) ^a | Eq. (91.9) ^b | 34 kJ limit ^b | <i>P</i> = 1 |
| RQ-4 Global Hawk | 28,002,000 | 28,002,000 | 28,002,000 | 28,002,000 |
| MQ1 Predator | 7,879,500 | 7,879,500 | 7,879,500 | 7,879,500 |
| RQ-2 Pioneer | 1,738,500 | 1,738,500 | 1,738,500 | 1,738,500 |
| RQ-11 Raven | 280,485 | 287,987 | 288,000 | 288,000 |
| Rmax type IIG | 566,023 | 566,069 | 566,069 | 566,069 |

^aUsing vehicle kinetic energy estimated from (91.11)

^bWorst-case vehicle kinetic energy estimate

For the first three options and under both scenarios, parameter α was chosen to be 100 kJ and β equal to 34 J.

In addition to the probability of fatality, the required system reliability was also calculated for each of the aforementioned fatality probability models as well as for the conservative estimate of probability of 1. The UAS reliability requirement has been given in minimum hours between ground impact accidents, and its calculation is based on a target level of safety of 10^{-7} fatalities per hour of flight. The system reliability requirement was derived since it allows a comparison with the current performance of manned and unmanned aviation. The results for each UAS and each case are summarized in Tables 91.15–91.18.

In the pessimistic scenario, the probability of fatality associated with each UAS is, almost in every case, 100 %. As a result, there are no differences between the reliability requirements calculated from the different models. Considering that current manned aviation accident rates are in the order of 10^{-7} h^{-1} for air carriers and 10^{-5} h^{-1} for general aviation (Table 91.4), it is obvious that for operations in high population density areas, certain UAS will need to exceed this performance.

In the optimistic scenario, the most striking differences between vulnerability models can be seen! This is due to the effect of the sheltering factor, which is not taken into account when using a threshold kinetic energy. Smaller systems feature

Table 91.17 Fatality probability with respect to ground impact accidents for five UAS under the optimistic scenario. Four different means of obtaining the fatality probability are used

| UAS model | <i>P</i> (fatality exposure) | | |
|------------------|------------------------------|-----------------------------|------------------------------|
| | Eq. (91.9) ^a (%) | Eq. (91.9) ^b (%) | 34 kJ limit ^b (%) |
| RQ-4 Global Hawk | 94.5 | 95.1 | 100.0 |
| MQ1 Predator | 75.4 | 81.1 | 100.0 |
| RQ-2 Pioneer | 49.0 | 72.5 | 100.0 |
| RQ-11 Raven | 3.6 | 10.7 | 100.0 |
| Rmax type IIG | 9.8 | 26.5 | 100.0 |

^aUsing vehicle kinetic energy estimated from (91.11)

^bWorst-case vehicle kinetic energy estimate

Table 91.18 Reliability requirement for five UAS with respect to ground impact accident under the optimistic scenario. Five different fatality probability estimates are used

| UAS model | Required time between ground impact accidents in hours | | | |
|------------------|--|-------------------------|--------------------------|--------------|
| | Eq. (91.9) ^a | Eq. (91.9) ^b | 34 kJ limit ^b | <i>P</i> = 1 |
| RQ-4 Global Hawk | 264,481 | 266,239 | 280,020 | 280,020 |
| MQ1 Predator | 59,394 | 63,909 | 78,795 | 78,795 |
| RQ-2 Pioneer | 8,514 | 12,598 | 17,385 | 17,385 |
| RQ-11 Raven | 102 | 309 | 2,880 | 2,880 |
| Rmax type IIG | 554 | 1,501 | 5,661 | 5,661 |

^aUsing vehicle kinetic energy estimated from (91.11)

^bWorst-case vehicle kinetic energy estimate

fatality probabilities of 10–25 % that are further reduced to 4–10 % when using a less conservative estimate for the kinetic energy at impact. This is also evidenced in the system reliability requirement, which is at least an order of magnitude smaller compared to that obtained using the threshold function. The benefits are evidenced in larger systems as well, where the reliability requirement is lower by a factor up to 3.

91.8 Conclusion

This chapter has investigated ways to calculate the target level of safety requirement for UAS based on the current levels of safety of manned aviation. As mentioned before, actual regulations will need to depend on a number of factors, and as a result, it is possible that they will contradict the results shown. Nevertheless, the methodologies are still useful for getting an idea of the relevant risk imposed by UAS as well as for arguing a safety-based authorization for operations.

Moving beyond the actual risk model and target safety level chosen, it is normally necessary to obtain design specifications and requirements on the hardware and software components that comprise the UAS rather than restrictions on the UAS as a whole. The way to derive these requirements is beyond the scope of the chapter,

but it normally involves a lengthy, formal process of identifying the hazards, the resulting failure conditions and their likelihoods, and then working backwards, derive requirements for the system, subsystems, and individual components. Even when a UA is designed in such a way that safety requirements are met for every conceivable application, additional risk mitigation measures may still need to be taken depending on the actual operating scenario.

References

- R. Clothier, R. Walker, Determination and evaluation of UAV safety objectives, in *Proceedings of the 21st International Unmanned Air Vehicle Systems Conference*, Irvine, 2006, pp. 18.1–18.16
- R. Clothier, R. Walker, N. Fulton, D. Campbell, A casualty risk analysis for unmanned aerial system (UAS) operations over inhabited areas, in *Proceedings of the 12th Australian International Aerospace Congress and 2nd Australasian Unmanned Air Vehicles Conference*, Melbourne, 2007
- J.K. Cole, L.W. Young, T. Jordan-Culler, Hazards of falling debris to people, aircraft, and watercraft. Sandia report, SAND97-0805, Sandia National Laboratories, 1997
- K. Dalamagkidis, On integrating unmanned aircraft systems into the national airspace system, in *Tutorial Presentation in 3rd International Symposium on Unmanned Aerial Vehicles (UAV'10)*, Dubai, UAE, 2010
- K. Dalamagkidis, K. Valavanis, L. Piegl, Current status and future perspectives for unmanned aircraft system operations in the U.S. *J. Intell. Robot. Syst.* **52**(2), 313–329 (2008)
- K. Dalamagkidis, K. Valavanis, L. Piegl, *On Integrating Unmanned Aircraft Systems into the National Airspace System: Issues, Challenges, Operational Restrictions, Certification, and Recommendations*. Intelligent Systems, Control and Automation: Science and Engineering, vol. 36, 2nd edn. (Springer, Dordrecht/New York, 2012)
- European Aviation Safety Agency (EASA), A-NPA, No. 16/2005, policy for unmanned aerial vehicle (UAV) certification (2005)
- European Aviation Safety Agency (EASA), Certification specification 25 (CS25). Amendment 3 (2007)
- European Aviation Safety Agency, Airworthiness certification of Unmanned Aircraft Systems (UAS). Policy statement, E.Y01301 (2009)
- Federal Aviation Administration, Equipment, systems and installations in part 23 airplanes. AC 23.1309-1C (1999)
- D.I. Feinstein, W.F. Haugel, M.L. Kardatzke, A. Weinstock, Personnel casualty study. Technical Report Project No. J 6067, Illinois Institute of Technology Research Institute, 1968
- FSF Editorial Staff, See what's sharing your airspace. *Flight Saf. Dig.* **24**(5), 1–26 (2005)
- J.M. Haber, A.M. Linn, Practical models of human vulnerability to impacting debris, in *Proceedings of the First IAASS Conference: "Space Safety, a New Beginning"*, Nice (ESA SP-599), 2005, pp. 543–548
- D.R. Haddon, C.J. Whittaker, *Aircraft Airworthiness Certification Standards for Civil UAVs* (UK Civil Aviation Authority, London, 2002)
- INnovative Operational UAS Integration (INOUI), Proposal for the integration of UAS into non-segregated airspace. Booklet (2009)
- Joint Capability Group on Unmanned Aerial Vehicles, STANAG 4671 – Unmanned aerial vehicle systems airworthiness requirements (USAR). Draft, NATO Naval Armaments Group (2007)
- Joint JAA/Eurocontrol Initiative on UAVs, A concept for European regulations for civil unmanned aerial vehicles (UAV). Final report, 2004
- National Transportation Safety Board (NTSB), Accident database and synopses (2008a), <http://www.nts.gov/ntsb/query.asp> (online)

- National Transportation Safety Board (NTSB), Aviation accident statistics (2008b), <http://www.nts.gov/aviation/Stats.htm> (online)
- Range Safety Group, Range Commanders Council, Range safety criteria for unmanned air vehicles. Document 323–99 (1999a)
- Range Safety Group, Range Commanders Council, Range safety criteria for unmanned air vehicles – rationale and methodology supplement. Supplement to document 323–99 (1999b)
- Range Safety Group, Range Commanders Council, Common risk criteria standards for national test ranges. Document 321–07 (2007a)
- Range Safety Group, Range Commanders Council, Common risk criteria standards for national test ranges: supplement. Supplement to document 321–07 (2007b)
- L.M. Sturdivan, D.C. Viano, H.R. Champion, Analysis of injury criteria to assess chest and abdominal injury risks in blunt and ballistic impacts. *J. Trauma* **56**(3), 651–663 (2004)
- U.S. Department of Defense, *Unmanned Systems Safety Guide for DoD Acquisition*, 1st edn. (Version .96) (2007)
- U.S. Department of Defense Office of the Secretary of Defense, Unmanned aircraft systems roadmap 2005–2030. Report, 2005
- R.E. Weibel, Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system. Master's thesis, Department of Aeronautics & Astronautics, Massachusetts Institute of Technology, 2005
- R.E. Weibel, R.J. Hansman, Safety considerations for operation of small unmanned aerial vehicles in civil airspace. Presented in MIT Joint University Program Quarterly Meeting, Boston, 2003
- R.E. Weibel, R.J. Hansman, Safety considerations for operation of different classes of UAVs in the NAS, in *Proceedings of the AIAA 4th Aviation Technology, Integration and Operations Forum and AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, Chicago, 2004

Reece A. Clothier and Rodney A. Walker

Contents

| | | |
|--------|--|------|
| 92.1 | Introduction | 2231 |
| 92.1.1 | Scope | 2232 |
| 92.1.2 | Aim and Overview of Chapter | 2233 |
| 92.2 | Establishing the Context | 2234 |
| 92.2.1 | Safety Risk Management Process and UAS | 2234 |
| 92.2.2 | The Objective | 2235 |
| 92.2.3 | Considerations and Constraints on the UAS Safety Risk Management Process | 2235 |
| 92.2.4 | Stakeholders | 2235 |
| 92.2.5 | High-Level Safety Criteria | 2238 |
| 92.2.6 | Summary | 2243 |
| 92.3 | Risk Identification | 2243 |
| 92.3.1 | Risk Identification Tools | 2244 |
| 92.3.2 | The Identification of Hazards | 2247 |
| 92.3.3 | The Contributing Failures and Conditions | 2249 |
| 92.3.4 | Assessing the Potential Consequences | 2253 |
| 92.3.5 | The Set of Scenarios | 2254 |
| 92.4 | Risk Analysis | 2255 |
| 92.4.1 | Assessing the Consequence | 2255 |
| 92.4.2 | Likelihood of Occurrence | 2256 |
| 92.4.3 | Assessing the Risk | 2256 |
| 92.4.4 | Uncertainty | 2257 |
| 92.5 | Risk Evaluation | 2257 |
| 92.5.1 | The ALARP Framework | 2258 |
| 92.5.2 | Evaluating the Risk | 2260 |

R.A. Clothier (✉)

School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Bundoora, Melbourne, VIC, Australia

e-mail: reece.clothier@rmit.edu.au

R.A. Walker

Australian Research Centre for Aerospace Automation, Queensland University of Technology, Brisbane Airport, Brisbane, QLD, Australia

92.6 Risk Treatment 2260

 92.6.1 Prioritization of Treatment 2261

 92.6.2 Determining Available Mitigation Options 2261

 92.6.3 The Selection of Mitigation Strategies 2264

 92.6.4 Summary 2268

92.7 Monitor and Review 2269

 92.7.1 The Importance of Accident and Incident Recording 2269

 92.7.2 Triggers for Review 2270

 92.7.3 Tracking Safety Performance 2270

92.8 Communication and Consultation 2270

92.9 Conclusion 2271

References 2271

Abstract

The safety risk management process describes the systematic application of management policies, procedures, and practices to the activities of communicating, consulting, establishing the context, and assessing, evaluating, treating, monitoring and reviewing risk. This process is undertaken to provide assurances that the risks associated with the operation of unmanned aircraft systems have been managed to acceptable levels. The safety risk management process and its outcomes form part of the documented safety case necessary to obtain approvals for unmanned aircraft system operations. It also guides the development of an organization’s operations manual and is a key component of an organization’s safety management system. The aim of this chapter is to provide existing risk practitioners with a high level introduction to some of the unique issues and challenges in the application of the safety risk management process to unmanned aircraft systems. The scope is limited to safety risks associated with the operation of unmanned aircraft in the civil airspace system and over inhabited areas. This chapter notes the unique aspects associated with the application of the safety risk management process to UAS compared to that of conventionally piloted aircraft. Key challenges discussed include the specification of high-level safety criteria; the identification, analysis and evaluation of the risks; and the effectiveness of available technical and operational mitigation strategies. This chapter also examines some solutions to these challenges, including those currently in practice and those still under research and development.

Acronyms

| | |
|-------|---|
| ACAS | Airborne collision avoidance systems |
| ADF | Australian Defence Force |
| ADS-B | Automatic dependent surveillance-broadcast |
| ALARP | As low as reasonably practicable |
| ALoS | Acceptable level of safety |
| ATSB | Australian Transport Safety Bureau |
| CAA | Civil Aviation Authority (United Kingdom) |
| CASA | Civil Aviation Safety Authority (Australia) |

| | |
|-------|---|
| COTS | Commercial-Off-The-Shelf |
| CPA | Conventionally-piloted aircraft |
| DoD | U.S. Department of Defense |
| EASA | European Aviation Safety Agency |
| ELoP | Equivalent level of performance |
| ELoS | Equivalent level of safety |
| FAA | Federal Aviation Administration |
| FMEA | Failure modes and effects analysis |
| GCS | Ground control station |
| HAZOP | Hazard and operability analysis |
| HLSC | High-level safety criteria |
| HSE | Health and Safety Executive (United Kingdom) |
| ICAO | International Civil Aviation Organization |
| ISO | International Organization for Standardization |
| LoS | Line of sight |
| NAA | National aviation authority |
| NTSB | National Transportation Safety Board |
| RPA | Remotely piloted aircraft |
| SARPS | Standards and Recommended Practices |
| SMS | Safety management system |
| SRMP | Safety risk management process |
| SSP | State Safety Plan |
| TCAS | Traffic Alert and Collision Avoidance System |
| UAS | Unmanned/uninhabited aircraft/airborne/aerial system/s (plural same as singular) |
| UAV | Unmanned/uninhabited aircraft/airborne/aerial vehicle/s (plural same as singular) |

92.1 Introduction

Unmanned aircraft systems (UAS) are one of a number of emerging sectors of the aviation industry. The potential benefits from the use of UAS have been demonstrated in a variety of civil and commercial applications including crop and infrastructure management, emergency management, search, and rescue, law enforcement, environmental research, and many other applications often described as being too *dull, dirty, dangerous, or demanding* for conventionally piloted aircraft (CPA). However, as well as benefits, the operation of UAS has associated risks.

Intrinsic to the realization of any system is a finite degree of risk; subsequently, accidents involving UAS will occur no matter how stringent the conditions prescribed or draconian the regulatory oversight provided. One could argue that the only way to assure absolute safety is to prohibit the deployment of UAS altogether.

However, to justify this argument, one must also address the philosophical question of *what are the risks of not using UAS technologies?*

The starting premise of this chapter, and one which is consistent with modern aviation safety thinking (ICAO 2009) is that UAS operations, like CPA operations, are not currently, and never will be, absolutely safe (i.e., have zero associated risks). The challenge for UAS stakeholders is to establish a safety case detailing how these inherent risks can be managed to an acceptable level.

Achieving an *acceptable level* of risk is a multidisciplinary problem. It requires a balancing of complex social, psychological, technical, political, and economic factors arising due to the following:

- Limited knowledge and resources available to identify characterize, and treat the safety risks associated with a technology
- Subsequent need to make trade-offs between available risk mitigation strategies based on assessments of the associated costs and benefits
- Potentially conflicting values, beliefs, perceptions, objectives, and expectations held by the different stakeholder groups involved in the decision-making process (e.g., those held by the UAS industry, other airspace user groups, and the general public)
- Conditions and environment under which the decisions are made (e.g., hidden political or time pressures)

Achieving a balanced outcome from such a problem space is the objective of the safety risk management process. This objective is achieved through the application of the safety risk management process (SRMP), which can be described as

the systematic application of management policies, procedures and practices to the activities of communicating, consulting, establishing the context, and identifying, analyzing, evaluating, treating, monitoring and reviewing risk. [Definition 3.1, (ISO 2009)]

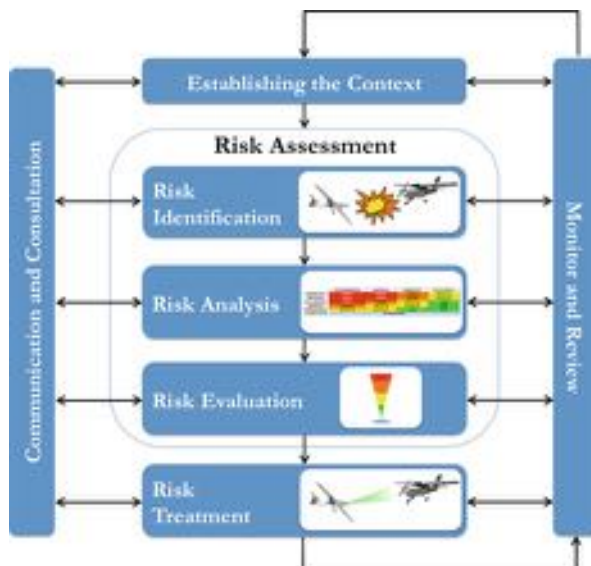
This chapter explores some of the unique aspects, issues, and challenges associated with application of the SRMP to the safety risks associated with UAS operations.

92.1.1 Scope

Discussion in this chapter is limited to the safety risks associated with civil UAS operations. There are a variety of descriptions of the SRMP, and these descriptions can differ in their scope, subprocesses, and structure. For the purposes of this chapter, the generalized and domain-independent description of the SRMP provided in ISO 31000:2009 is used and illustrated in Fig. 92.1 (ISO 2009). Some aviation-specific descriptions of the SRMP can be found in references (FAA 2000; ICAO 2009; CAA 2010b).

Establishing, maintaining, and improving safety requires more than the application of an SRMP. The SRMP is conducted as part of an organizational risk framework developed in accordance with a fundamental set of organizational risk principles (ISO 2009). In aviation parlance, these principles and the organizational framework in which the SRMP is applied are part of an organization's safety

Fig. 92.1 The safety risk management process, based on ISO (2009)



management system (SMS) (ICAO 2009). The scope of this chapter does not include the SMS. For general information on the components of the SMS, the reader is referred to the references (ICAO 2009; ISO 2009).

92.1.2 Aim and Overview of Chapter

The aim of this chapter is to provide existing risk practitioners with a high-level introduction to some of the unique issues and challenges in the application of the SRMP to unmanned aircraft systems. This chapter does not provide a comprehensive description of the SRMP itself. The discussion is intentionally high level in its nature to ensure applicability to a broad range of UAS and their potential concepts of operation.

The structure of this chapter follows the SRMP illustrated in Fig. 92.1. The first step in any SRMP is to establish the context, which is described in Sect. 92.2. This is followed by the risk assessment process. The objective of the risk assessment process is to comprehensively characterize the safety risks associated with UAS operations and, based on this information, determine which of the characterized risks can be tolerated and which of the characterized risks require mitigation (treatment). As illustrated in Fig. 92.1, the risk assessment process comprises the subprocesses of risk identification, risk analysis, and risk evaluation. These are discussed in Sects. 92.3–92.5, respectively. The objective of the risk treatment process (described in Sect. 92.6) is to identify, implement, and evaluate suitable measures to reduce (mitigate, modify, treat, or control) the risk. The SRMP is a *living* process being a key component of an organization's overarching SMS. The process of monitoring and reviewing (Sect. 92.7) is pivotal to maintaining

and improving the management of the risks. Finally, there is the process of communication and consultation (Sect. 92.8). The communication and consultation process is key to addressing broader stakeholder concerns and those issues that stem from a lack of knowledge of the risks and benefits associated with civil UAS operations.

92.2 Establishing the Context

Understanding the complexity of challenges to be faced in the safety risk management of UAS requires consideration of the social, psychological, political, and economic factors associated with the broader integration of UAS into society. These factors are identified as part of the context for the SRMP and are commonly overlooked in UAS safety discussions. Establishing the context is the process of “defining the external and internal parameters to be taken into account when managing risk, and setting the scope and risk criteria for the risk management policy” (ISO 2009). This subprocess of the SRMP involves consideration of the “cultural, social, political, legal, regulatory, financial, technological, economic, natural, and competitive environment, whether international, national, regional or local; the key drivers and trends having impact on the objectives of the organization; and relationships with, and perceptions and values of external stakeholders” (ISO 2009).

92.2.1 Safety Risk Management Process and UAS

The SRMP can be used to support a range of operational, financial, or regulatory decisions concerning UAS. Here, we will focus on those decisions made in relation to the management of the safety risks associated with their operation. In this context, the SRMP provides an accepted and systematic means for providing assurances that the risks associated with UAS operations have been managed to an acceptable level. The SRMP and its outcomes form part of the documented safety case necessary to obtain approvals for UAS operations. It also guides the development of an organization’s operations manual and is a primary component of an organization’s SMS. The SRMP is also used to guide the safety policy, rulemaking, and oversight activities of a national aviation authority (ICAO 2009).

The Civil Aviation Safety Authority (CASA) has released draft guidance material describing the application of SMS principles to civil UAS operators (CASA 2011). The guidance material is believed to be the first of its kind specifically targeted to civil UAS operations. Drawing on ICAO SMS principles and internal CASA policy, the guidance material includes recommendations on how UAS operators should approach the safety risk management of UAS operations (ICAO 2009). Although not a regulatory requirement, CASA actively encourages UAS operators to develop an SMS due to the potential benefits of improved safety and reduced costs.

92.2.2 The Objective

One of the first steps is to define the objectives of the activity. The general overarching objective is to provide assurances in the safety of a particular UAS operation or organization's activities. Objectives also need to be defined in relation to the expected benefits of the operation to the different stakeholders involved. For commercial UAS operations, these objectives can often be derived from the corporate and strategic objectives of the organization (e.g., profitability, market growth, reputation). As well as being a goal, objectives can also act as constraints on decisions made throughout the SRMP. All objectives should be clearly defined to ensure transparency in decision-making to help identify potential conflicts in the SRMP.

92.2.3 Considerations and Constraints on the UAS Safety Risk Management Process

Constraints bound the decisions made within the SRMP and can arise due to a variety of financial, legal, social, psychological, technological, temporal, or spatial limitations or requirements. For example, the national aviation authority (NAA) functions of safety policy, rulemaking, and oversight must be defined in consideration of ICAO Standards and Recommended Practices (SARPS); the safety performance objectives established within a State Safety Plan (SSP); the legal, political, economic, and cultural requirements specific to their respective state; and the internal resources and capability of the NAA to define and execute these functions. Constraints are typically categorized as being either internal or external to the organization. Internal constraints are those that arise due to limits in the capability or resources of the organization or due to the organization's existing policies, procedures, or objectives. External constraints include existing regulations (e.g., existing civil aviation safety, environmental protection, or workplace health and safety legislation) or other social, cultural, political, or economic expectations held by other stakeholders (including the members of the general public).

92.2.4 Stakeholders

A stakeholder can be defined as "an individual, group of people, organization or other entity that has a direct or indirect interest (or stake) in a system" (Hull et al. 2011). An *interest* may arise through the stakeholder using, benefiting from, being disadvantaged by being responsible for, or otherwise being affected by the system (Hull et al. 2011). Stakeholders in the UAS SRMP can include other airspace users, the general public, air traffic service providers, the end users of UAS services or their data products, the aviation safety regulator, landowners, and members of the UAS

industry (inclusive of equipment and airframe manufacturers, operators, training, and maintenance organizations). Stakeholders will have their own objectives, information needs, and expectations in terms of the safety performance of UAS. These need to be identified and considered at all stages of the SRMP.

The acceptance of UAS operations requires more than a solid safety case. Understanding stakeholder concerns, the motivation for them and how they influence their decisions in relation to safety, is key to achieving the broader acceptance of UAS operations. Clothier et al. (2008) use the situation faced by horseless carriages in the 1800s as an analogy to the situation being faced by UAS today. As described in Clothier et al. (2008), there are hidden factors concerning the integration of UAS into society that can influence stakeholder decision-making in relation to their safety. UAS are a new user within an existing airspace system. Further, there exist potentially competing industries, whose value and safety performance is already widely known and tolerated by society. These and other factors (e.g., the unemployment of pilots) can manifest as hidden objectives and constraints on the SRMP. Effective stakeholder communication is pivotal to the identification, characterization, and resolution of the potential conflicts that can arise in the SRMP.

92.2.4.1 Perception

A distinction is often made between those stakeholder assessments of the safety risks that are formed through the use of *objective* data, expert domain knowledge, models, or formal assessment techniques, and those assessments that are based on the *subjective* knowledge, beliefs, emotions, values, and needs of the individual. The latter of these types of assessments is commonly referred to as perceived risk. There is a range of factors that influence how different stakeholders appraise and respond to the safety risks associated with UAS operations. Importantly, these appraisals and responses can be different to those they would make for the safety risks associated with CPA operations. These perceptions give rise to different stakeholder expectations in terms of the safety performance of UAS.

At the time of writing, no significant body of research into the perception of the safety risks associated with UAS operations could be found. Clothier and Walker (2006); Clothier et al. (2008) provide limited discussion on factors likely to influence the perception and acceptability of the risks associated with UAS operations. Also worth noting is the survey of air travelers conducted by MacSween-George (2003). This survey attempted to characterize the willingness of people to travel onboard a pilotless passenger aircraft.

In the absence of a risk perception study specific to UAS, general factors taken from existing psychometric modeling studies (Fischhoff et al. 1978; Slovic et al. 1979; Slovic 1987, 1999) are used to hypothesize the public's perception of the safety risks associated with UAS operations. An analysis of the UAS safety paradigm with respect to the factors of voluntariness of exposure, control of exposure, awareness of benefits, and uncertainty is described below.

Voluntariness. The primary risks of concern due to CPA operations are to the crew and passengers onboard the aircraft. The individuals exposed voluntarily

undertake these risks in return for a direct benefit. On the other hand, for UAS operations, the primary risks are to members of the general public overflown who are largely involuntarily exposed to the risks.

Control. The members of the general public overflown by UAS operations are largely unable to influence the level of their exposure. Whereas passengers of CPA have greater control over the level of risk they are willing to tolerate through the number and type of aircraft operations (e.g., gliding, sport aviation, or scheduled passenger flights) they partake in and through choice of a particular air service provider.

Benefit. The knowledge of the benefits of CPA operations (e.g., efficient transportation of people and freight) is broadly understood and widely known. Further, there is a direct and identifiable relationship between the individuals exposed and the benefits they receive. However, the routine operation of UAS for civil and commercial applications has yet to be realized, and as a consequence broader society has limited, if any, knowledge of the benefits. For UAS, the connection between benefits and the individual exposed may not always be identifiable to the individual exposed.

Knowledge and Information. In relation to UAS, there are limited sources of information available to stakeholders. The quality of the information that is available to stakeholders is variable, biased, and often unverified. For example, the movie *Stealth*TM portrays UAS with unrealistic capabilities. The information available predominantly relates to military UAS operations and their roles in recent conflicts (e.g., as weapons of war). This can create a bias in stakeholder knowledge of UAS. There is also a significant knowledge gradient between stakeholders (i.e., a difference in the amount and quality of knowledge held by the different stakeholder groups). The general public and the NAAs have less personal knowledge that they can use to contrast/verify the information available to them. Whereas the industry stakeholders have much more experience and knowledge relating to UAS operations and their safety performance. This knowledge gradient can lead to issues of trust and in turn higher stakeholder uncertainty in assessments of the risks. Finally, the above factors can lead to lower stakeholder certitude (e.g., belief in their self-knowledge), and potential issues of trust can lead to higher perceptions of the risk. These and other factors give rise to stakeholder uncertainty. The higher the uncertainty, the higher the perception of the risks.

Based on the above factors, it is hypothesized that stakeholder perceptions of the risks associated with UAS operations will be higher than that for a comparable CPA operation. Addressing the issues relating to risk perception requires the development of communication strategies (Sect. 92.8). Psychological factors influence not only stakeholder assessments of the risks but also their appetite for them. It has been proposed that stakeholders will expect UAS to demonstrate a level of safety performance better than that currently expected of CPA operations. If true, this expectation will need to be taken into consideration when defining high level safety criteria (HLSC) for UAS. Most qualitative specifications of HLSC for UAS express a desire for UAS to exhibit a level of risk less than, or equal to, that currently demonstrated by CPA. Some quantitative specifications of HLSC for UAS include

a multiplicative factor to account for the hypothesized difference in stakeholder appetite for risk, for example (Weibel and Hansman 2004).

92.2.5 High-Level Safety Criteria

HLSC are qualitative or quantitative statements describing “the terms of reference against which the significance of a risk is evaluated” (ISO 2009). A review of regulations, regulatory guidance material, and industry position papers yielded a disparate array of qualitative and quantitative statements of the HLSC for UAS. Based on this review, the existing HLSC can be broadly categorized into one of two general categories: acceptable level of safety (ALoS) and equivalent level of safety (ELoS) criteria. These HLSC are not to be confused with equivalent level of performance (ELOP) requirements, which are briefly described in Sect. 92.2.5.3.

92.2.5.1 Acceptable Level of Safety Criteria

The first category of HLSC for UAS are those defined in relation to an ALoS. Examples of existing qualitative statements of ALoS HLSC are provided in Table 92.1. Although ALoS HLSC avoid many of the issues associated with making a direct comparison to the safety performance of CPA (discussed in the next section), they provide no guidance as to what constitutes an *acceptable* level of safety.

Table 92.1 Examples of qualitative specifications of the acceptable level of safety criteria for UAS

| Statement | Reference |
|--|---|
| “UAS must operate safely, efficiently, and compatibly with manned aircraft operation in the airspace so that the overall safety of the airspace is not degraded. The fundamental safety requirement for the UAS is to provide an acceptable level of risk for people and property in the air and on the ground” p. 1 | RTCA Guidance Material (RTCA 2007) |
| “... UAS are to provide and acceptable level of risk for people and property on the ground and in the air and to operate without adversely affecting the existing users of the NAS.” p. 11 | |
| “Enable the operation of sUAS [<i>small UAS</i>] by mitigating, to an acceptable level of risk, the hazards posed to manned aircraft and other airborne objects operating in the National Airspace System (NAS) as well as the public on the surface.” p. iii | Recommendations from the Aviation Rulemaking Committee, FAA (SUAS 2009) |
| “Any sUAS may be operated in such a manner that the associated risk of harm to persons and property not participating in the operation is expected to be less than acceptable threshold value(s) as specified by the Administrator.” p. 53 | |
| “Regulations are intended to ensure that the UAV systems and their operations achieve an acceptable level of safety for people and property in other aircraft and on the surface.” pp. 2–46 | MITRE Issues paper (DeGarmo 2004) |

One approach for qualifying/quantifying *acceptable* is to base it on the de facto levels of risk determined for other activities (e.g., smoking or riding a bike) or naturally occurring events (e.g., death due to being struck by lightning). For example, the Swedish Aviation Authority use the probability of someone dying in a road accident to guide the setting of ALoS criteria for UAS (Wiklund 2003). Another approach is to directly adopt existing safety criteria specified in the regulation of other industries (e.g., as used for land use planning, space launch activities, and nuclear energy industry).

92.2.5.2 Equivalent Level of Safety Criteria

The second and most common category of HLSC for UAS qualifies *acceptable* through reference to the safety performance currently exhibited by CPA. Safety performance is expressed as the level of risk or the potential for harm (i.e., the existence of hazards). These comparative HLSC are widely referred to as ELoS criteria, and some qualitative examples are provided in Table 92.2.

There are a number of critical assumptions that need to be considered in the use of ELoS HLSC. Firstly, there is the foundational assumption that risks as tolerated or accepted in the past (i.e., those associated with CPA operations) provide a suitable basis for judging the acceptability of future risks associated with a different technology (i.e., those associated with UAS operations). Many of the factors discussed in Sect. 92.2.4 would challenge this assumption.

Secondly, ELoS criteria require a mechanism for making comparisons between the different categories of CPA and of UAS. For example, CASA states that HLSC for UAS should be defined in relation to CPA of *equivalent class or category* (CASA 2002). A range of mechanisms for making such a comparison have been proposed and include those based on similarities in the maximum takeoff weight of the aircraft, the maximum kinetic energy of the aircraft under different failure modes (JAA/EUROCONTROL 2004) or in the expected number of casualties (Grimsley 2004). For some UAS, it is not possible to establish an equivalent type within the CPA fleet on the basis of a similarity in the attributes of the aircraft alone. This issue is clearly illustrated in the comparative histogram plots of the UAS and CPA fleets presented by Clothier et al. (2011). Even if equivalence in terms of a similarity in aircraft attributes can be made, such attributes may not account for the differences between the safety risk profiles associated with the two different aviation concepts. These differences are discussed in the risk assessment subprocesses of Sects. 92.3, 92.4, and 92.5.

A range of measures, reference data, and approaches have been used to quantify ELoS criteria and some examples are provided in Table 92.3. There are a number of issues associated with the use of these measures. Firstly, most of the ELoS HLSC were determined through an historical analysis of CPA accident and incident data. As discussed by Clothier and Walker (2006), this quantification approach can be sensitive to the period over which the historical analysis is conducted and the type of CPA activity considered. Averaging over a historical period does not reflect trends in the safety performance of CPA operations or the infrequent nature of the events being characterized. Further, the averaged/aggregated measures do not

Table 92.2 Examples of qualitative specifications of the equivalent level of safety criteria for UAS

| Statement | Reference |
|--|---|
| <p>“The principal objective of the aviation regulation framework is to achieve and maintain the highest possible uniform level of safety. In the case of UAS, this means ensuring the safety of any other airspace user as well as the safety of persons and property on the ground.” p. 4</p> <p>“[<i>this framework</i>] ... will provide, at a minimum, an equivalent level of safety for the integration of UAS into non-segregated airspace and at aerodromes.” p. 4</p> <p>“The introduction of RPA [<i>remotely piloted aircraft</i>] must not increase the risk to other aircraft or third parties and should not prevent or restrict access to airspace.” p. 17</p> | ICAO circular (ICAO 2011) |
| <p>“UAV operations should be as safe as manned aircraft insofar as they should not present or create a hazard to persons or property in the air or on the ground greater than that created by manned aircraft of equivalent class or category.” p. 11</p> <p>“When considering a request for approval to conduct a particular operation with a UAV, CASA must ensure that the operation of the UAV will pose no greater threat to the safety of air navigation than that posed by a similar operation involving a manned aircraft. This characteristic may be termed ‘acceptable’.” p. 18</p> | CASA advisory circular (CASA 2002) |
| <p>“... UAS operations must be as safe as manned aircraft insofar as they must not present or create a greater hazard to persons, property, vehicles or vessels, whilst in the air or on the ground, than that attributable to the operations of manned aircraft of equivalent class or category.” Sect. 1, Chap. 1, p. 1</p> | CAA-UK Guidance material (CAA 2010a) |
| <p>“A civil UAS must not increase the risk to people or property on the ground compared with manned aircraft of equivalent category.” p. 4</p> | EASA Policy statement (EASA 2009) |
| <p>“UAV Operations shall not increase the risk to other airspace users or third parties.” p. 12</p> <p>“If civil UAV Systems are to become a reality the industry must gain the acceptance and confidence of these people [<i>general public and existing airspace users</i>], and this could be achieved by demonstrating a level of safety at least as demanding as the standards applied to manned aircraft.” p. 12</p> | JAA and EUROCONTROL, Report (JAA/EUROCONTROL 2004) |
| <p>“... it is broadly accepted by European military authorities that UAV operations outside segregated airspace should be conducted at a level of safety equivalent to that for manned aircraft. Similarly, UAV operations should not increase the risk to other airspace users and should not deny the airspace to them.” p. 6</p> | EUROCONTROL Specifications (EUROCONTROL 2007) |
| <p>“... UAVs must demonstrate that they do not pose an undue hazard to other aircraft or persons on the ground. They must, in short, provide for an equivalent level of safety to manned aircraft.” pp. 2–1</p> | MITRE Issues paper (DeGarmo 2004) |
| <p>“UASs shall operate to equivalent levels of safety as manned aircraft in regard to the risk they pose to people on the ground, other aircraft and property.” MILAVREG 7.1 p. 1, Sect. 2, Chap. 7</p> <p>“The objective of the unmanned aerial systems (UAS) airworthiness regulations is to ensure that UAS operations present no greater risk to personnel, other aircraft and property than that accepted for the operation of manned aircraft, without undue compromise to operational flexibility.” Sect. 5, Chap. 3, p. 1</p> | Australian Defence Force airworthiness regulations and guidance material (ADF 2009) |

(continued)

Table 92.2 (continued)

| Statement | Reference |
|--|---|
| “Any UAV operation or test must show a level of risk to human life no greater than that for an operation or test of a piloted aircraft.” p. 3 | U.S. Range Commanders Council, Flight test range safety requirements (RCC 1999) |
| “The hazards associated with a specific UAV are defined in the hazard analysis (risk management criteria). The range must ensure that the risks to people identified in the hazard analysis are reduced to an acceptable level. . . .The criteria is[sic] met if the hazard is confined to unpopulated areas (2.1) or if the combined vehicle reliability and population distribution results in a risk is[sic] no greater than that for manned aircraft operations (2.2).” p. 4 | |
| “The Army vision is to have “file and fly” access for appropriately equipped UAS by the end of 2012 while maintaining an equivalent level of safety (ELOS) to aircraft with a pilot onboard.” p. 105 | U.S. Army, Unmanned systems roadmap (DoD 2010b) |
| “... it is MoD policy that the operation of a RPAS [Remotely Piloted Air Systems] should be no more likely to cause injury or fatality to personnel or the general public than the operation of a manned aircraft.” | UK Mod 1000 Series (GEN) (MAA 2011) |

Clarifications indicated in [bracketed] and *italicized* text

account for peak risks that can occur due to geospatial or temporal concentrations in aviation activity or variations in the level of exposure of different subgroups within the populations exposed to the risks (e.g., the level of risk to pilots and aircrew compared to the level of risk to members of the general flying public).

To ensure a more comprehensive management of the risks associated with UAS operations and to be consistent with the safety risk management of other industries (see HSE 2001b), Clothier et al. (2011) propose that the specification of HLSC for UAS includes measures indicative of the individual and societal risk, in addition to the measures of group/collective risk that have been previously proposed. Further, it is recommended that the HLSC for UAS be defined based on the peak risks associated with CPA operations as opposed to averaged values.

Irrespective of the measures used or where the baseline level of safety is set (e.g., equivalent to that of CPA or not), there is the inherent difficulty of verifying that a system or operation actually satisfies the HLSC.

92.2.5.3 Equivalent Level of Performance Requirements

The ELoS criteria described in the previous section should not be confused with the requirement for an ELoS as described in (FAA 2009, 2011b; Wolfe 2009) which are referred to here as ELoP requirements. Whereas ELoS criteria are expressed in terms of levels of safety or of risk, ELoP criteria are typically expressed in terms of equivalence to the following:

- Existing regulations, standards, or procedures for CPA (e.g., design or operating requirements)
- Functions or functional performance (e.g., UAS must demonstrate a *sense and act* function equivalent to the *see and avoid* function provided by a human pilot).

ELoP requirements are not HLSC but lower-level requirements mandated to control (or mitigate) the risks associated with UAS operations (discussed further

Table 92.3 Examples of quantitative specifications of the equivalent level of safety criteria for UAS

| Value | Measure | Method and data used | Reference |
|----------------------------------|--|--|---|
| Ground Fatality Criteria | | | |
| 1.0×10^{-06} | Number of ground fatalities per flight hour | Based on civil aviation and U.S. Navy aviation accident data compiled over the period 1982–1998. Used by Grimsley (2004) to determine permissible system failure rates for UAS. Also used in Appendixes 3–5, enclosure 3 of (JAA/EUROCONTROL 2004) | U.S. Range Commanders Council (RCC 1999, 2001) |
| 1.0×10^{-07} | Number of ground fatalities per flight hour | Based on references to existing historical studies | Dalamagkidis et al. (2008) |
| 8.4×10^{-08} | Number of involuntary ground fatalities per flight hour | Derived directly from an analysis of aviation accident and incident data compiled by the NTSB over the period 1984–2004 | Clothier and Walker (2006) |
| 1.0×10^{-08} | Number of involuntary ground fatalities per flight hour | Based on analysis of aviation accident and incident data compiled by the NTSB over the period 1983–2003. Includes multiplicative factor to account for differences in the public’s appetite for UAS risks | Weibel and Hansman (2004) |
| 1.0×10^{-08} | Nominal likelihood of a mishap causing serious injury, loss of life, or significant damage per flight hour | Interim/recommended level for management of all UAS hazards (including risks to people on the ground). Reference level based on existing Australian Defence Force (ADF) aviation risk management guidelines and assumptions made in relation to the average duration of UAS operations | Australian Defence Force airworthiness regulations (ADF 2009) |
| Midair Collision Criteria | | | |
| 4.10×10^{-07} | Midair collision rate per flight hour | Determined directly from an analysis of aviation accident and incident data compiled by the NTSB over the period 1984–2004 | Clothier and Walker (2006) |
| 2.32×10^{-07} | Fatal midair collision rate per flight hour | Determined directly from an analysis of aviation accident and incident data compiled by the NTSB over the period 1984–2004 | Clothier and Walker (2006) |
| 1.0×10^{-07} | Midair collision rate per flight hour | Based on aviation accident and incident data compiled by the NTSB over the period 1983–2006 | Dalamagkidis et al. (2008) |
| 1.0×10^{-08} | Nominal likelihood of a mishap causing serious injury, loss of life, or significant damage per flight hour | Interim/recommended level for management of all UAS hazards (including midair collisions). Reference level based on existing ADF aviation risk management guidelines and assumptions made in relation to the average duration of UAS operations | ADF airworthiness regulations (ADF 2009) |
| 1.0×10^{-09} | Midair collisions per flight hour | Existing target level of safety criteria provided in Federal Aviation Administration (FAA) System Safety guidelines (FAA 2000) | Weibel and Hansman (2004) |

in the Section 92.2.5.3). Satisfying an ELoP does not necessarily give rise to an ELoS. The use of ELoP as de facto safety criteria requires assumptions to be made in relation to the nature of the relationship between system performance (e.g., reliability) and the level of risks to different entities of value (e.g., the potential damage to people and property).

92.2.6 Summary

Establishing the context defines the inputs, desired outputs, and the boundaries and constraints on decisions made throughout the SRMP. It is important to note that obtaining a *public license* for UAS operations must take into consideration a broad range of issues. The integration of a new technology into society is subject to a wide range of broader social, political, cultural, and economic considerations. For example, one of the primary concerns identified in the survey of air travelers conducted by MacSween-George (2003) was the potential unemployment of pilots. A search of mainstream media sources reveals numerous articles identifying a broad range of public concerns including privacy, noise and public disturbance, and the potential misuse of UAS by drug traffickers or terrorists. Such concerns can be as significant as those issues relating to their safety. Further research is needed to characterize the safety criteria for UAS and to better understand different stakeholder concerns, perceptions, and expectations. In the interim, guidance can potentially be found through exploring the safety risk management of other new technologies, such as genetically modified foods, nanotechnologies, stem cell research, nuclear power, and the use of automation in the rail and shipping industries.

92.3 Risk Identification

The objective of the risk identification process is to identify how the system can fail, how these failures and conditions manifest as hazards, and the potential undesired outcomes that can result from the occurrence of the hazards. The identification of a specific combination of these three components describes a risk scenario. The set of all risk scenarios can be defined through the identification of the set of hazards, and for each particular hazard the associated sets describe the following:

1. The different conditions, failures, and events contributing to the occurrence of the particular hazard
2. The potential types and levels of consequential outcomes associated with the occurrence of the particular hazard

The set of all scenarios identified with a given activity is described as the risk profile. By way of general introduction, the high-level UAS and CPA risk profiles are illustrated in Figs. 92.2 and 92.3, respectively. Illustrated in Figs. 92.2 and 92.3 are the primary and secondary hazards and their potential consequential outcomes to people and property. Not shown are the conditions, failures, and events contributing to the occurrence of the hazards. The profiles, and the tools, data, and techniques that can be used to identify and characterize them are described in the following subsections.

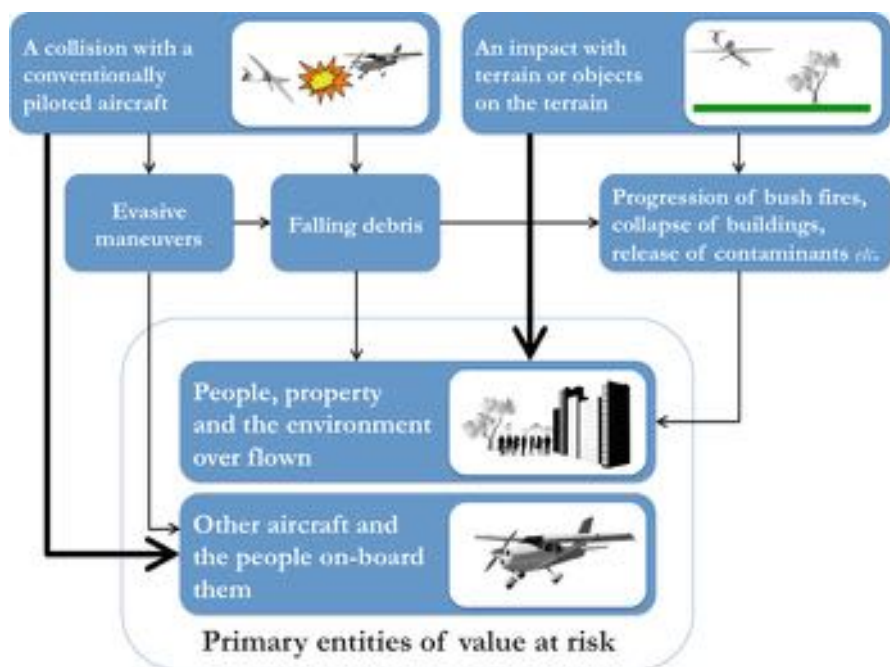


Fig. 92.2 Illustration of the high-level risk profile associated with UAS operations

92.3.1 Risk Identification Tools

A range of techniques can be used to identify and characterize the risk scenarios associated with UAS operations. The CAA categorizes these techniques into historical (e.g., a review of accident and incident data), brainstorming (e.g., elicitation of knowledge from domain experts), and systematic (e.g., formal tools and processes) techniques (CAA 2010b).

A typical starting point for any risk identification process is a review of existing accident and incident data. Such a review can provide general insights into the key hazards and their likely consequential outcomes and, depending on the scope and quality of the investigative reports available, the factors contributing to their occurrence. Some notable examples of UAS accidents and incidents are provided in Table 92.4.

There is limited data on UAS accidents and incidents. The majority of publicly available data relate to military UAS operations primarily because of the limited amount of nonmilitary UAS activity to date (a product of the current regulatory environment) and that mandatory reporting of accidents and incidents involving nonmilitary UAS has only recently come into force (refer to Sect. 92.7.1). Seldom does a review of accident and incident data provide a *comprehensive* identification of the potential hazards and their outcomes. This is particularly the case for UAS,

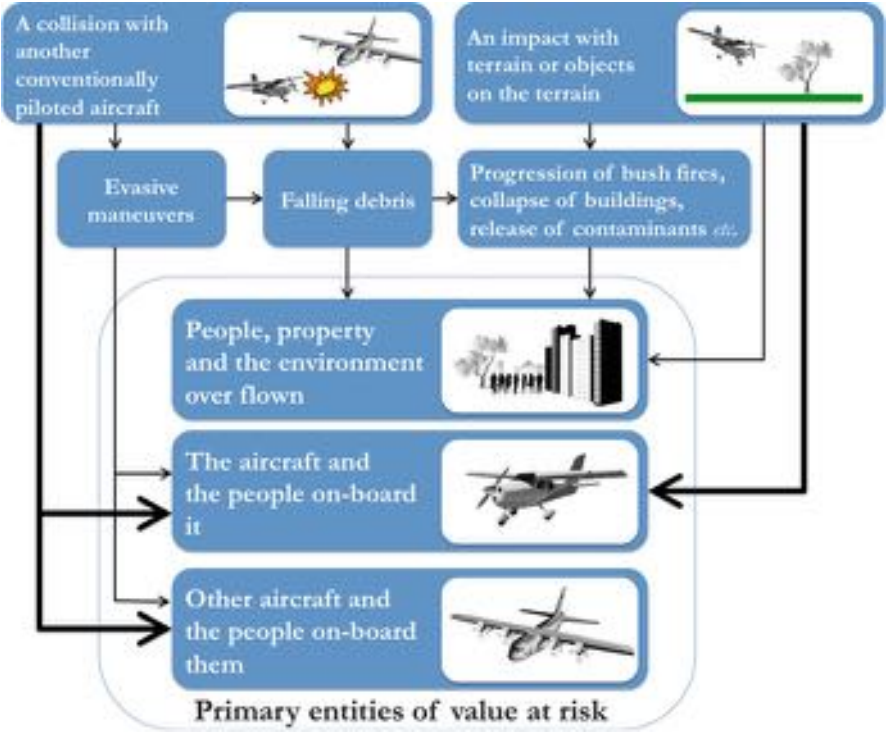


Fig. 92.3 Illustration of the high-level risk profile associated with CPA operations

where there is limited data available and the primary hazards are inherently rare events. Further, the ability to identify the complexity of factors contributing toward the occurrence of an accident or incident is often restricted by the method and quality of the records available. Incidents occur more frequently than accidents. Incidents provide valuable information as precursor or *lead indicators* for accidents; however, less information is typically available in incident reports due to the limited amount of resources available to investigate them. There is also a bias in the data toward military UAS operations, and therefore, when using this data, it is important to consider some of the differences between military and nonmilitary UAS operations. For example, the potential differences:

- Between the design and operational philosophies adopted for military and nonmilitary UAS (e.g., trade-offs made between survivability and mission risk vs. public, and personnel risk)
- Between the environments they are operated in (e.g., natural environment, mix and types of other airspace users, and electromagnetic environment)
- In how they are managed within the airspace system (e.g., procedures for separation, the situational awareness available to air traffic control, the UAS operators and other airspace users, and the type of services provided)

Table 92.4 Notable accidents and incidents involving UAS

| | Date | Location | Description of accident/incident | Reference |
|----|-------------------|--|--|--------------------------------------|
| 1. | May 10, 2012 | Incheon, South Korea | A Schiebel S-100 Camcopter crashed into its ground control station, killing one crew member and injuring two others. The cause for the crash is still being investigated | Mortimer (2012) |
| 2. | Aug 17, 2011 | Afghanistan | RQ-7 Shadow and a C-130J military transport aircraft both operated by the U.S. Department of Defense collided over Afghanistan. The C-130J made an emergency landing. The RQ-7 was destroyed on impact with the ground. No casualties were reported | Hodge (2011) |
| 3. | July 19, 2011 | Karachi, Pakistan | An Uqab UAS operated by the Pakistan Navy encountered bird strike and subsequently crashed in the vicinity of an oil refinery. No casualties or damage to the refinery was reported | Siddiqui (2011) |
| 4. | December 17, 2010 | El Paso, USA | An Aeronautics Orbiter Mini UAS operated by the government of Mexico crossed the U.S. border and crashed in a residential area. No injuries resulted | Washington Valdez and Borunda (2010) |
| 5. | May 10, 2007 | Dili, East Timor | A SkyLark UAS operated by the Australian Defence Force during a peacekeeping mission crashed into a house. The house was unoccupied at the time | Fitzpatrick (2007) |
| 6. | October 05, 2006 | Kinshasa, Democratic Republic of Congo | A Belgian B-Hunter UAS operated as part of the European Union Force peacekeeping mission crashed shortly after takeoff. The crash and ensuing fire killed one civilian and injured at least three others | La Franchi (2006a) |
| 7. | April 25, 2006 | Arizona, USA | A General Atomics Aeronautical Systems MQ-9 Predator UAS crashed near Nogales, Mexico, in the USA while on a U.S. Customs and Border Patrol mission. The aircraft was destroyed on impact with no casualties reported | NTSB (2007) |
| 8. | August 30, 2004 | Afghanistan | A near midair collision between a Luna UAS operated by the German Army and an Ariana Afghan Airlines Airbus A300B4 with over 100 passengers and crew onboard. The two aircraft passed within 50 m of each other. The wake turbulence from the A300B4 caused the Luna UAS to crash. No casualties were reported | La Franchi (2006b) |
| 9. | December 06, 1999 | Edwards Air Force Base, USA | RQ-4A Global Hawk autonomously responded to an erroneous command to taxi at a speed of 155 knots. The aircraft left the taxiway before ground crew could respond. The aircraft sustained substantial damage, but no injuries or damage to other aircraft occurred | AIB (2000) |

(continued)

Table 92.4 (continued)

| | Date | Location | Description of accident/incident | Reference |
|-----|----------------|-----------------|---|----------------------------|
| 10. | March 29, 1999 | China Lake, USA | A RQ-4A Global Hawk received and executed a command to terminate flight. The aircraft was destroyed on impact with the ground, and no casualties were reported. The command was sent by operators conducting radio testing at a neighboring test range. The high altitude of operation of the global hawk was not considered as part of the frequency management plan | Drezner and Leonard (2002) |

- In the nature of the missions performed (e.g., low-level flights, maneuver, and mission profiles)
- In their hazards (e.g., for military UAS, there are unique hazards associated with the carriage of ordinance, self-protection systems, and payload self-destruct mechanisms)

These and many other differences can give rise to unique sets of risk scenarios for military and nonmilitary UAS operations. Although a valuable input to the risk identification process, UAS accident and incident data should not be used as the sole means for risk identification. This data should be complemented by other risk identification techniques to ensure a comprehensive identification of the risks. References (SAE 1996; FAA 2000; FAA and EUROCONTROL 2007) describe a number of tools, and that can be used in the identification and analysis of aviation safety risks. A domain-independent review of over 100 different risk identification and analysis techniques can be found in Stephens et al. (1997). Commonly used risk identification and analysis tools are provided in Table 92.5.

92.3.2 The Identification of Hazards

The specification of a risk scenario starts with the identification of the hazards. A hazard is a state or condition that has the potential to cause loss to something of value. ISO31000:2009 describes the analogous concept of a risk source, defined as an “element which alone or in combination has the intrinsic potential to give rise to risk” (ISO 2009). Prescriptive definitions of hazard can be found in ICAO (2009); DoD (2010a).

92.3.2.1 Primary Hazards of Concern Associated with UAS Operations

A primary hazard is one that has the potential to directly cause harm. Some definitions of primary hazard include the additional condition of *immediately* (Dalamagkidis et al. 2008); however, such definitions preclude primary hazards that have delayed effects or require long-term exposures (e.g., radiation, psychological losses, exposure to carcinogens, or damage to ecosystems). With respect to the operation of UAS in the civil airspace system and over inhabited areas, the primary hazards of concern are well known and common to those for CPA operations. As described

Table 92.5 Some common risk identification and analysis tools

| Description | References |
|---|--|
| <i>Functional hazard analysis</i> is a predictive risk identification technique that attempts to identify and explore the effects of functional failures at different representative levels of a system (aircraft and system levels) | SAE (1996) |
| <i>Failure modes and effects analysis</i> (FMEA) explores how different components and functions can fail (modes) and the potential effects in relation to other components or functions of a system. Failure mode, effects and criticality analysis (FMECA) includes assessments of the likelihood of end consequences | SAE (1996, 2001) and CAA (2010b) |
| <i>Hazard and operability analysis</i> (HAZOP) is a structured and qualitative group brainstorming approach for identifying hazards and their contributing failures | FAA and EURO-CONTROL 2007 and CAA (2010b) |
| <i>Common cause analysis</i> is a technique aimed at identifying risk scenarios in which two or more events could occur as the result of one common event/failure. It combines a number of sub-techniques: zonal analysis, particular risks assessment, and common mode analysis | SAE (1996) and FAA and EURO-CONTROL (2007) |
| <i>Bow-tie analysis</i> primarily used as a risk analysis/modeling tool; a bow-tie analysis combines models (e.g., event tree or fault trees) with consequence modeling tools (e.g., consequence tree) to explore how a particular hazard (the bow) arises and the subsequent manner and types of possible consequential outcomes | FAA and EURO-CONTROL (2007) |
| <i>Event tree analysis</i> used as both a risk identification and an analysis tool; an event tree models sequences of causally related events from an initiating event | CAA (2010b) |
| <i>Fault Tree Analysis</i> is a deductive (top down) graphical risk identification and analysis tool for determining different logic paths in which the top level undesired event could occur | SAE (1996) |
| <i>External events analysis</i> is a risk identification tool focusing on how external/environmental factors/inputs can influence the behavior of the system. It is also useful for the identification of security threats | FAA and EURO-CONTROL (2007) |

in JAA/EUROCONTROL (2004), Clothier and Walker (2006), and Dalamagkidis et al. (2008) and as illustrated in Fig. 92.2 these hazards are the following:

- (A) A collision with a CPA (situated on the ground or in the air) and the potential harm caused to people onboard the CPA (e.g., incident 2 in Table 92.4)
- (B) The controlled or uncontrolled impact with terrain or objects on the terrain (such as people or structures), for example, incidents 5 and 6 in Table 92.4

92.3.2.2 Secondary Hazards of Concern Associated with UAS Operations

Secondary hazards of concern are those that can occur as a result of a primary hazard. Some of the secondary hazards associated with the primary hazard A above, include the potential harm caused to people:

1. On the ground due to falling aircraft or debris from a midair collision (e.g., the falling debris described in incident 2 in Table 92.4)

2. On the ground due to falling aircraft or debris from a near midair collision (e.g., incident 8, Table 92.4, where wake turbulence caused the loss of the UAS)
3. Onboard the CPA due to evasive maneuvers performed in order to avoid a collision with a UAS (while either of the aircraft is in the air or on the ground)

Some of the secondary hazards associated with the primary hazard B above include the potential harm caused to people on the ground due to the following:

1. Release of hazardous materials (e.g., chemical payloads, composite materials, or ordnance) following an impact with terrain or an object on the terrain
2. Progression of fires, the collapse of buildings, motor vehicle accidents, or other hazards arising as a result of the UAS coming to earth (e.g., in incident 3 of Table 92.4 there was the potential for an explosion or fire had the UAS damaged critical components of the oil refinery)

As can be observed in Figs. 92.2 and 92.3, the primary and secondary hazards identified within the UAS risk profile also exist within the CPA risk profile. However, not shown are differences in the failures and conditions contributing to the occurrence of these hazards and in the types and levels of consequence associated with their occurrence.

92.3.3 The Contributing Failures and Conditions

There are a variety of ways in which the hazards illustrated in Fig. 92.2 can eventuate. The specification of a risk scenario includes identifying how a particular hazard can occur. A hazard is typically the result of a series of active failures in combination with latent conditions that involve all components of the system (i.e., the interaction of the components of man, machine, and organization) and the interaction of the system within its operating environment. Some key techniques for identifying these failures and conditions include FMEA, HAZOP, fault tree analysis, human factors studies (discussed below), and anticipatory failure determination.

High-level guidance on common factors contributing to UAS mishaps can be found in studies of existing accident and incident data. For example, some frequent causes of mishaps reported by the U.S. Department of Defense (DoD) are summarized in Table 92.6.

92.3.3.1 Unique Components and Functions

There are some obvious differences in the design and operation of UAS when compared to CPA. For example, a communications link for command and control is a critical component of the safe operation of UAS particularly in the absence of aircraft autonomy (i.e., a remotely piloted aircraft). Other unique components of an UAS include the ground control element, flight termination systems, and devices used in the launch and recovery of the air vehicle. The existence of these components can create unique hazards and contribute toward the occurrence of the

Table 92.6 Percentage of mishaps attributed to different failure mode categories, from OSD (2003)

| Failure mode category | Description | % of total mishaps ^a attributed to category ^b |
|-----------------------|---|---|
| Power/propulsion | Encompasses the engine, fuel supply, transmission, propeller, electrical system, generators, and other related subsystems onboard the aircraft | 37 |
| Flight control | Includes all systems contributing to the aircraft stability and control such as avionics, air data system, servo-actuators, control surfaces/servos, onboard software, navigation, and other related subsystems. Aerodynamic factors are also included in this grouping | 26 |
| Human/ground | Accounts for all failures resulting from human error and maintenance problems with any non-vehicle hardware or software on the ground | 17 |
| Communications | The datalink between the aircraft and the ground | 11 |
| Miscellaneous | Any mission failures not attributable to those previously noted, including airspace issues, operating problems, and other nontechnical factors | 9 |

^aDefined as an accident resulting in significant vehicle damage or total loss of human life, or causing more than \$1,000,000 in damage

^bAveraged over 100,000 flight hours across five different UAS types

primary hazards illustrated in Fig. 92.2. Consideration of such components (and their failures) is not captured in existing CPA risk identification studies.

92.3.3.2 The Importance of a “Systems” Mentality

UAS are more than an aircraft. Consideration of the individual components of the UAS in isolation of the other components of the system and its environment would fail to provide a comprehensive identification of the risks. An emergent property is one which is not determined solely from the properties of the system’s parts but which is additionally determined from the system’s structure and behavior (Thomé 1993). These emergent properties and the boundaries and constraints on them are all potential sources for active failures or latent *accident-producing* conditions. For example, the UAS *system* has the property of line-of-sight (LoS) communication range. LoS range is an emergent property, arising due to the interactions between the system and its environment. Specifically, it is a function of the state of the air vehicle (e.g., antenna attitude), the properties of the communications system (e.g., frequency and minimum permissible signal to noise ratio for a given bit error rate), the ground control system (e.g., geographical position), the mission (e.g., the flight path), and the environment (e.g., weather, terrain, and ambient radio frequency environment). Together, these properties interact to define the maximum LoS range of the system at a given time. Exceeding this range can contribute to the occurrence of a hazard (i.e., a loss of command and control, which for an RPA, could lead to a mishap).

92.3.3.3 The Human Element

Despite the relocation of the pilot, the *human element* still has a significant contribution to the safety of UAS operations. A clear example of this is incident 7 in Table 92.4 (refer to associated accident report). An analysis of U.S. DoD operations recorded over the 10-year period ending in 2003 (Tvaryanas et al. 2005) found that 68.3 % of the 211 mishaps reviewed involved *operations or maintenance organizational, supervisory, or individual human factors*. References (Manning et al. 2004; Williams 2004; McCarley and Wikens 2005; Tvaryanas et al. 2005; Hobbs 2010) provide further analysis and discussion of the contribution of the *human element* to UAS accidents using a variety of modeling frameworks. Common human factors identified in these studies include crew resource management, decision-making, situational awareness, human machine interface design, training, task load, and fatigue. Psychological issues can include the apparent *risk-taking* behavior of UAS operators due to the absence of a *shared fate* between the operator and the UAS; issues of operator trust, awareness, and dependency on automation; issues associated with a handover between remotely located operators; and issues relating to the simultaneous control of multiple UAS. It is important to consider human factors in all aspects of a UAS deployment and not just its launch, operation, and recovery. For example, Hobbs and Stanley (2005) identify the personnel issues of *complacency* and a *model aircraft culture* in the maintenance of UAS; such factors can contribute toward the 8 % of U.S. DoD UAS accidents that were the direct result of maintenance errors (Tvaryanas et al. 2005). For some UAS, much of the maintenance can be performed in the field during an active deployment (e.g., change of payloads, replacement of wings, minor repairs). Maintenance in the field can be subject to additional time pressures (e.g., push for readiness for next deployment), poor working conditions (e.g., exposure to the environment), and the need to make decisions and actions without access to all the necessary information or tools (e.g., arising due to poor logistics and operational planning).

92.3.3.4 The Operation and the Environment

It is important to consider how failures can eventuate through the interaction of the UAS and its operational environment. Many of the hazards arising from the natural environment are common to CPA and are well known, for example, storms and bird strikes (e.g., incident 3 in Table 92.4). However, for UAS, the detection of these conditions can be difficult as the operator is not located onboard the aircraft, and even if it is detected, many UAS do not have the same resilience to them as CPA (e.g., the absence of anti-icing systems or bird strike protection).

A single UAS type can be used for a wide range of applications. The potential failures and conditions need to be investigated for these different operations and environments. For example, the low-altitude operation of UAS in the vicinity of structures creates a number of additional challenges over UAS operations in relatively clear areas. For example, large structures can impede communications, create turbulent environments, and degrade navigation performance through increased multipath and a reduction in the number of visible GPS satellites.

92.3.3.5 Software

Most nonmilitary UAS make use of Commercial-Off-The-Shelf (COTS) consumer-grade software that is often provided without warranty or assurance. Without such assurances, it can be extremely difficult to assess the likelihood of encountering latent errors or undesired behavior. Often, the dependability of software can only be gauged through extensive experience in its use under a variety of conditions. Configuration control is also particularly important for those systems using COTS software. Small bug fixes and *auto updates* to operating systems can introduce new latent conditions and significantly change the stability and behavior of the software system as well as its performance under existing conditions. Software considerations should extend to include any electronic databases (e.g., publicly available digital elevation maps), firmware, operating systems, and applications used during flight or prior to and after flight (e.g., flight planning, software, and documentation control systems). In addressing software-related risks, there are two separate, yet often confused, considerations. Firstly, there are risks associated with the behavior of algorithms and, in the case of UAS, the validity of autonomous behavior. The latter is particularly of concern when the level of autonomy increases (Parasuraman et al. 2000). The second consideration relates to the implementation of the algorithm and is addressed by standards such as DO-178B (RTCA 1992).

92.3.3.6 Security

Security threats are a subcategory of hazards. More specifically, they are hazards that arise, either directly or indirectly, through the intentional disturbance of the safe or normal operational state of the UAS. Most often, these disturbances originate from objects external to the system, which exploit the interfaces between the UAS and its environment (e.g., interference, jamming, or the overriding of control via communications links or physical access to the ground control station (GCS)). The security of the UAS should take into consideration:

- The type of radio control gear, voice, and data links used for communication between all components of the system (including ground personnel and air traffic control)
- Whether the links are vulnerable to intentional or unintentional interference and whether the loss of this link has a safety impact for different phases of the operation
- The type of information conveyed on these links and its criticality to the safety of the operation of the aircraft if corruption, disruption, or spoofing occurs
- Whether the sender or recipient of the information on these links needs to be verified or not (e.g., incident 10 described in Table 92.4)
- The location and physical security of the GCS and any launch, recovery, communications relay, maintenance, and storage sites
- Whether software security, such as firewalls and antivirus programs, is installed and used
- Policies in relation to access to the Internet and the transfer of media via removable storage.

92.3.3.7 The Criticality of Failure Modes

Firstly, flight critical failures are no longer restricted to the aircraft; one must also identify those flight critical failures that exist in the GCS and communications components of the UAS. Secondly, what is considered a catastrophic failure for CPA may not necessarily be catastrophic for a UAS, and vice versa. For CPA, the assignment of criticality to a failure is based on an assumed exposure probability of one (i.e., there is always at least one person onboard; thus, someone is always exposed, see Fig. 92.3). For UAS, the exposure probability is a complex function that depends on where the UAS is operated. In some cases, the exposure probability may approach zero (e.g., those UAS operations restricted to uninhabited areas and in segregated airspace). In such cases, the failure criticality can potentially be assigned to a lower severity category (e.g., major or hazardous), and this assignment should be based on the potential impact of the failure on the ability of the UAS to remain in its predefined operational area. On the other hand, some failures for UAS may have a higher criticality due to the absence of the additional protection provided by a pilot onboard. Thus, adopting existing CPA failure criticality assignments for UAS must be treated with caution.

92.3.4 Assessing the Potential Consequences

The final component of the specification of a risk scenario is the identification of the potential consequential outcomes. Explicitly linked to the concept of hazard are the concepts of loss, harm, or consequence. For example, the definition of hazard provided by ICAO (2009) includes a specification of the types of consequential outcomes to be considered:

Any real or potential condition that can cause injury, illness, or death to personnel; damage to or loss of a system, equipment or property; or damage to the environment. pp. 4–1 (ICAO 2009)

As can be observed in Fig. 92.3, the risks associated with CPA operations include consideration of the potential harm to people onboard the aircraft in addition to those onboard other CPA or on the ground. An analysis of worldwide accidents involving conventionally piloted commercial jet aircraft over the period 2001–2010 reveals that more than 95 % of all fatal injuries were to people onboard an aircraft (Boeing 2011). Therefore, for both of the primary hazards associated with CPA operations, the consequences of principal concern are those to the passengers and crew onboard the aircraft and, secondarily, to the population of people external to the aircraft (e.g., those living in the regions overflowed). For UAS, there are no people onboard the aircraft, and the primary risks are instead to those entities of value considered external to the system. Consequently, the primary types and spectra of consequential outcomes associated with UAS operations are different to those associated with CPA operations.

92.3.4.1 Domains of Consequence

There are a variety of potential consequential outcomes associated with the occurrence of a hazard. For example, MIL-STD-882D (DoD 2010a) defines loss in terms

of damage to people, equipment or property, or the environment. These *types* of loss describe the different domains of consequence. Typically, the primary domain of consequence is that of physical harm to people, with secondary domains being the potential loss registered to equipment and property (inclusive of the air vehicle), the environment, the organization (e.g., financial, reputational, capability, market, or mission losses), clients, the broader industry, and the losses registered to other less tangible values held by society (e.g., culture, trust).

Distinctions are often made between consequences of the same type. For example, the risk management of CPA operations makes a distinction between those operations with fare-paying passengers onboard and those without. Such a distinction is made due to social and psychological factors that influence the general public's perception and acceptance of risk (e.g., the assigned value, dread, fear, control, voluntariness of exposure). For UAS, similar distinctions will need to be made in relation to the primary entities of value at risk on the ground. For example, a distinction is often made between third-party casualties (e.g., a member of the public) and a first-party casualty (e.g., personnel supporting the deployment of the UAS). Similar distinctions are made between damage to property and the damage to hospitals, schools, residential areas, historical or culturally significant sites, etc.

92.3.4.2 The Spectra of Consequences

A qualitative or quantitative spectrum of consequence needs to be defined for each domain of consequence identified. Take, for example, the consequence domain of people. The associated scale of loss could be defined from no injury to multiple fatal injuries. As shown in the studies by Clothier et al. (2010) and Fraser and Donnithorne-Tait (2011), there can be categories of UAS which are unable to cause significant and direct physical harm to other aircraft or people or property on the ground. For these categories of UAS, the losses associated with secondary domains of consequence (e.g., organizational, financial, or environmental) or those losses arising due to the occurrence of secondary hazards (e.g., ensuing bushfires or downstream losses due to damage to critical infrastructure) are likely to be more significant in the evaluation and management of the safety of their operation.

92.3.5 The Set of Scenarios

The outcome of the risk identification process is a set of characterized scenarios. This set is seldom complete as there will always be unknown hazards or failures and conditions that can give rise to existing hazards. It is important that the hazard identification process is periodically reviewed to make use of new knowledge, information, or identification techniques (refer to Sect. 92.7). A hazard log should be maintained to record and track any new scenarios identified during the course of operations and should form a valuable input to any review of the risk assessment. Finally, the endeavor to ensure the set of scenarios is as comprehensive as possible, coupled with the use of conservative assumptions in their

characterization, can lead to the specification of unrealistic scenarios. There are limited resources available to treat the risks associated with the identified scenarios; therefore, it is important that all scenarios be subject to a test of plausibility.

92.4 Risk Analysis

The third step in the SRMP, Fig. 92.1, is an analysis of the risk. Risk analysis describes the process of characterizing the nature and level of the risk for each of the identified risk scenarios. A measure of risk is expressed through the combination of assessments of the consequence and the likelihood of occurrence of the given scenario.

92.4.1 Assessing the Consequence

A qualitative or quantitative table is often used to group and rank the different types and levels of consequence associated with the identified risk scenarios (examples shown in Table 92.7). An assessment of the consequence for a given risk scenario is made by mapping its potential outcomes to one of the consequence levels defined within the table. As there can be more than one consequential outcome associated with the occurrence of a single-risk scenario, a mapping is typically based on the worst possible outcome identified.

Table 92.7 Examples of existing consequence/severity classification schemes

| ICAO SMM (ICAO 2009) | MIL-STD-882D (DoD 2010a) |
|--|--|
| <i>Catastrophic</i> – equipment destroyed; multiple deaths | <i>Catastrophic</i> – could result in one or more of the following: death, permanent total disability, irreversible significant environmental impact, or loss exceeding \$10M |
| <i>Hazardous</i> – a large reduction in safety margins, physical distress, or a workload such that the operators cannot be relied upon to perform their tasks accurately or completely. Serious injury and major equipment damage | <i>Critical</i> – could result in one or more of the following: permanent partial disability, injuries or occupational illness that may result in hospitalization of at least 3 personnel, reversible significant environmental impact, or loss exceeding \$1M but less than \$10M |
| <i>Major</i> – a significant reduction in safety margins, a reduction in the ability of the operators to cope with adverse operating conditions as a result of increase in workload, or as a result of conditions impairing their efficiency. Serious incident and injury to persons | <i>Marginal</i> – could result in one or more of the following: injury or occupational illness resulting in 10 or more lost work days, reversible moderate environmental impact, or loss exceeding \$100K but less than \$1M |
| <i>Minor</i> – Nuisance; operating limitations; use of emergency procedures; minor incident | <i>Negligible</i> – could result in one or more of the following: injury or illness resulting in less than 10 lost work days, minimal environment impact, or loss less than \$100K |
| <i>Negligible</i> – little consequences | |

92.4.2 Likelihood of Occurrence

A range of formal techniques can be used to assess the likelihood of a scenario occurring (e.g., the risk assessment tools described in Table 92.5). Some high-level models characterizing the primary risk scenarios illustrated in Fig. 92.2 have also been proposed, for example (Weibel and Hansman 2004; Clothier et al. 2007; Lum et al. 2011; Lum and Waggoner 2011).

Assessments can draw on a range of information sources including incident and accident data, aircraft activity data, component reliability data, and expert knowledge. Assessment can also use existing models used in other application domains (e.g., space vehicle launch and reentry, motor vehicle accident studies, munitions, debris modeling, generic human injury models, and CPA airspace collision risk models). The output is a qualitative or quantitative assessment of the likelihood of realizing a given risk scenario. Depending on the tools and modeling approach used, this assessment can be used directly in the assessment of the risk or mapped to a likelihood scale or classification scheme (Table 92.8).

92.4.3 Assessing the Risk

A range of qualitative and quantitative scales have been used to describe levels of risk. For example, MIL-STD-882D (DoD 2010a) assesses risk on the qualitative ordinal scale: low, medium, serious, and high. The component measures of consequence (Sect. 92.4.1) and of likelihood (Sect. 92.4.2) then need to be *mapped* to one of these levels of risk. A risk matrix is the most common method for illustrating this mapping, and an example of which is provided in Fig. 92.4. ICAO (2009) also provides an example of a risk matrix.

Table 92.8 Examples of existing likelihood/probability classification schemes

| ICAO SMM (ICAO 2009) | MIL-STD-882D (DoD 2010a) |
|--|--|
| <i>Frequent</i> – likely to occur many times (has occurred frequently) | <i>Frequent</i> – likely to occur often in the life of an item with a probability of occurrence greater than 10^{-1} in that life |
| <i>Probable</i> – likely to occur sometimes (has occurred infrequently) | <i>Probable</i> – will occur several times in the life of an item; with a probability of occurrence less than 10^{-1} but greater than 10^{-2} in that life |
| <i>Remote</i> – unlikely to occur but possible (has occurred rarely) | <i>Occasional</i> – likely to occur sometime in the life of an item; with a probability of occurrence less than 10^2 but greater than 10^3 in that life |
| <i>Improbable</i> – very unlikely to occur (not known to have occurred). | <i>Remote</i> – unlikely but possible to occur in the life of an item; with a probability of occurrence less than 10^3 but greater than 10^6 in that life |
| <i>Extremely improbable</i> – almost inconceivable that the event will occur | <i>Improbable</i> – so unlikely, it can be assumed occurrence may not be experienced in the life of an item; with a probability of occurrence less than 10^6 in that life <i>Eliminated</i> – incapable of occurrence in the life of an item (hazard has been eliminated) |

| | CATASTROPHIC | CRITICAL | MARGINAL | NEGLIGIBLE |
|------------|--------------|----------|----------|------------|
| FREQUENT | HIGH | HIGH | SERIOUS | MEDIUM |
| PROBABLE | HIGH | HIGH | SERIOUS | MEDIUM |
| OCCASIONAL | HIGH | SERIOUS | MEDIUM | LOW |
| REMOTE | SERIOUS | MEDIUM | MEDIUM | LOW |
| IMPROBABLE | MEDIUM | MEDIUM | MEDIUM | LOW |
| ELIMINATED | ELIMINATED | | | |

Fig. 92.4 Example of a risk matrix as per MIL-STD-882D (2010)

92.4.4 Uncertainty

A particular issue in the safety risk management of UAS is managing uncertainty in the risk assessment process. Uncertainty can pervade all stages of the SRMP. Uncertainty influences the level of risk perceived by stakeholders (i.e., the higher the perceived uncertainty the higher the perceived risks) and their preferences for risk treatment (i.e., a preference to treat those risk scenarios with a higher degree of associated uncertainty). Uncertainty arises through a lack of knowledge and information available in the risk assessment process, differences in the level of knowledge held by stakeholders (leading to issues of trust), and a lack of transparency in the SRMP. An effective communication and consultation process (Sect. 92.8) is key to addressing the uncertainty of stakeholders. However, managing the uncertainty in the assessment process is particularly difficult for those UAS that employ COTS equipment with limited or no information on their reliability. A defensible starting position is to attempt to establish the boundaries on the assessment of the risks as opposed to an estimate of the *point value* of the risk. The upper boundary on the risk can be estimated by propagating the assumption that all systems and components *will* fail. An estimate on the lower boundary can be made by adopting a less conservative assumption based on the best available data. As the SRMP is iterative and ongoing, these initial and conservative assumptions can be revised as more experience and data becomes available. An introductory paper on the types of uncertainty and how uncertainty pervades the SRMP can be found in Zio and Pedroni (2012).

92.5 Risk Evaluation

Risk evaluation is the process of comparing the results of the risk analysis with the HLSC to determine whether the risk for a given scenario is tolerable (ISO 2009) or whether further measures need to be undertaken to reduce the risk. There are a range of decision-making frameworks that can be used within the risk evaluation process; these include the as low as reasonably achievable, *globalement au moins Équivalent*, or minimum endogenous mortality frameworks used in the Netherlands, France, and Germany, respectively. Discussion in this chapter is limited to the as low

as Reasonably practicable (ALARP) evaluation framework, which is advocated by ICAO (2009) and has been widely used in the management of a broad range of risks in the UK, the USA, and Australia.

92.5.1 The ALARP Framework

The ALARP framework is intended to represent safety decisions made in everyday life (HSE 1992, 2001b). There are some risks that people choose to ignore and others that they are not prepared to entertain irrespective of the benefits associated with them. In addition, there are those risks people are prepared to take by making a trade-off between the benefits of taking the risks and the precautions required to mitigate them (HSE 2001b). These three types of decision scenarios are the basis for the development of the ALARP framework. Referring to Fig. 92.5, the ALARP framework comprises

A Region of Broadly Unacceptable Risk – Except under extenuating circumstances, risks that fall within this region are generally considered unjustified regardless of the benefits associated with the activity. Such activities would be ruled

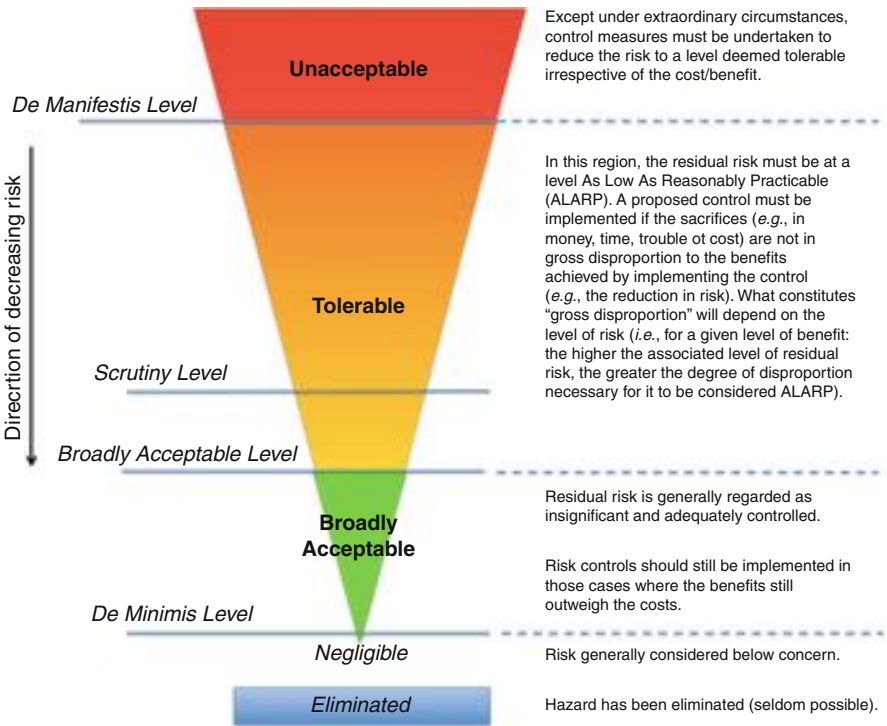


Fig. 92.5 The ALARP risk evaluation framework

out unless further action can be undertaken to reduce the risk (HSE 2001b). This region corresponds with the notion of a *de manifestis* level of risk, which is based on the legal definition of *obvious risk* (RCC 2007). It is defined as the level of risk above which a person of ordinary level of intelligence intuitively recognizes as being inherently unacceptable (Fulton 2002; RCC 2007).

A Region of Tolerability – This region describes those risks which are considered *tolerable*, specifically those situations where there is “...a willingness to live with a risk so as to secure certain benefits and in the confidence that it is being properly controlled. To tolerate a risk means that we do not regard it as negligible or something we might ignore, but rather as something we need to keep under review and reduce still further if and as we can” (HSE 1992). As described in HSE (2001b), risks that fall in the region are considered tolerable if and only if the:

- Risks have been properly assessed (e.g., assessments based on the best available scientific evidence or advice), and the results are used to determine appropriate measures to control the risks.
- Residual risks are not unduly high (e.g., above the *de manifestis* level) and are kept to level as ALARP.
- Risks are periodically reviewed.

A Region of Broadly Acceptable Risk – Risks within this region are “generally regarded as insignificant and adequately controlled” (HSE 2001b). There is no distinct *line* demarcating tolerable risks from broadly acceptable risks; instead, it has been described as the point at which “the risk becomes truly negligible in comparison with other risks that the individual or society runs” (HSE 1992). Obtaining a broadly acceptable level does not mean the pursuit for the reduction of risks to ALARP should be abandoned. As described by the UK Health and Safety Executive (HSE), “duty holders must reduce risks wherever it is reasonably practicable to do so or where the law so requires it” (HSE 2001b).

The Concept of ALARP – A risk is considered ALARP if the cost of any reduction in that risk is in gross disproportion to the benefit obtained from the reduction. Determining that risks have been reduced to a level as ALARP involves an assessment of the risk to be avoided, of the sacrifice or costs (e.g., in money, time, and trouble) involved in taking measures to treat that risk, and a comparison of the two to see if there exists a gross disproportion (HSE 2001a). General discussion on the cost-benefit process that needs to be undertaken and some guidance on the meaning of gross disproportion can be found in references (HSE 2001b,a; CASA 2010; Jones-Lee and Aven 2011).

De Minimis Level – Some specifications of the ALARP framework include a specification of the *de minimis* level of risk. The *de minimis* level stems from the legal principle *de minimis non curat lex* (the law does not concern itself with trifles) (Paté-Cornell 1994; Fulton 2002; RCC 2007). It is often used as a guide for determining when risks have been managed to a level that could be considered below concern.

A Scrutiny Level – Some implementations of the ALARP framework feature a scrutiny line, which is often used to put newly assessed risks in context with risks

that have been tolerated or broadly accepted in the past. Often, the scrutiny level represents the *de facto* risks for a similar activity/industry.

It is important to note that the meaning of ALARP and its implementation in law can change between states (an important consideration when it comes to the risk management of international UAS operations). The description of ALARP provided above is consistent with its implementation in those countries that adopt *common law* (e.g., the UK, the USA, Australia, Canada, New Zealand). Ale (2005) provides an example of some of the issues that can arise due to the application of safety decision-making frameworks such as ALARP within different legal systems.

There are psychological, social, and practical difficulties in the specification and sole use of quantifiable criteria within the ALARP framework. This has led to the use of qualitative frameworks that focus on demonstrating that all *reasonably practicable* measures have been undertaken to reduce a risk as opposed to making quantifiable comparisons of the assessed risks to specifications of the *de manifestis*, *de minimis*, or scrutiny levels. The results from comparisons of assessed risks to HLSC ultimately translate to requirements on design; hence, a quantifiable specification of HLSC within the ALARP framework is most desirable. When introducing a new technology into society one cannot avoid the commonly used *litmus test* of a comparison to similar and existing risks (as often made by the media or by members of the public). In this case, the ELoS HLSC (as described in Sect. 92.2.5) should be represented as scrutiny lines within the ALARP framework. Further research is needed to explore the psychological, social, and practical implications relating to the representation of the quantitative HLSC for UAS in the ALARP framework. There can also be general issues associated with the application of ALARP specifically to new technologies such as UAS, and these are discussed in Melchers (2001).

92.5.2 Evaluating the Risk

The ALARP framework is represented in a risk matrix by assigning the levels of risk, and hence cells of the matrix, to the different regions of the ALARP framework. This assignment is often illustrated through the use of a graduated color scale (e.g., refer to the corresponding colors used in Figs. 92.4 and 92.5). Refer to Figs. 5–4 and 5–5, pp. 5–8/9 of ICAO (2009) for another example of a representation of the ALARP framework within a risk matrix. Each risk scenario can then be mapped to one of the regions within the ALARP framework. Whether or not a particular scenario requires treatment will depend on the ALARP region it is mapped to (as described in Sect. 92.5.1).

92.6 Risk Treatment

For those risk scenarios that are not tolerable, measures need to be undertaken to reduce (mitigate, modify, treat, or control) the residual risk to a level considered as ALARP.

92.6.1 Prioritization of Treatment

The scenarios requiring treatment need to be prioritized due to a practical limit on the resources available to treat the risks. This prioritization is usually based on the level of unmitigated risk, with those scenarios having a higher level of risk given a higher priority for treatment. However, there are other factors that can influence the prioritization of scenarios, for example, the prioritization of scenarios:

- Based on the nature of their associated consequences. For example, the apparent public aversion to accidents with a higher-level consequence, psychological factors (e.g., fear or dread), or those scenarios that have prolonged or sustained consequences
- With a high level of uncertainty.

It is important to note that the treatment of some scenarios may be mandatory irrespective of their risks (e.g., due to environmental protection or workplace health and safety regulations).

92.6.2 Determining Available Mitigation Options

The first step is to determine a list of all possible treatment options. Guidance on potential mitigation strategies can be found in regulatory materials (CASA 2002; FAA 2011b) or by reviewing the safety cases used in the approval of existing operations. In general, risk mitigation strategies reduce the risk through the following:

- A. Removing the hazard altogether
- B. Reducing the likelihood that a hazardous event occurs
- C. Reducing the level of potential consequence associated with the occurrence of an hazardous event
- D. Sharing the retained risk with other organizations
- E. Combinations of the above

92.6.2.1 Risk Mitigation Strategies for Midair Collision

A range of strategies can be used to mitigate the risks associated with the hazard of a midair collision between a UAS and a CPA. Some example strategies are summarized in Table 92.9. The strategies in Table 92.9 are classified based on how the primary reduction in risk is achieved, specifically (1) through elimination of the hazard, (2) through a reduction in the likelihood the hazard occurs, or (3) through a reduction in the consequence given the occurrence of the hazard. Category 2 mitigation strategies are divided into the subcategories of the following:

- A. *See* – strategies that provide the UAS with an awareness of its air traffic environment
- B. *Be seen* – strategies that provide other airspace users with an awareness of the UAS
- C. *Staying away* – UAS operational strategies that reduce the likelihood of encountering other aircraft

Table 92.9 Examples of existing midair collision mitigation strategies

| | | |
|--|---|--|
| (1) <i>Elimination of the hazard</i> | Segregation of UAS from other aircraft (e.g., use of prohibited or restricted airspace); not conducting the operation | |
| (2) <i>Reduction in the likelihood of a hazard occurring</i> | (i) <i>Active</i> | (ii) <i>Passive</i> |
| (A) <i>See</i> | Periodic radio broadcasts; airborne or ground-based systems that employ primary radar (e.g., Fig. 92.6), transponder interrogators, or LIDAR sensors; existing airborne collision avoidance systems (e.g., TCAS-II) | Chase plane; radio listening watch; airborne or ground-based systems that employ electro-optical (Fig. 92.7), infrared, or acoustic sensors; automatic dependent surveillance-broadcast (ADS-B) in; ground-based visual observers; subscription to traffic information feeds |
| (B) <i>Be seen</i> | ADS-B out; transponders; existing ACAS (e.g., TCAS-II) inter-aircraft communication systems (e.g., VHF – Data Link); anticollision strobe lights (Fig. 92.8) | Chase plane; high-visibility paint (Fig. 92.8); establishment and activation of warning or danger areas |
| (C) <i>Staying away</i> | Flying in airspace of known low aircraft activity, over the oceans, above or below international en route airspace; at night, below the CPA minimum safe altitude, outside peak CPA traffic times | |
| (D) <i>Services</i> | Utilization of third-party air traffic services; flying in controlled airspace | |
| (E) <i>Strategic</i> | Survey and crew familiarization with airspace operating environment; crew training in procedures; general awareness (briefing local airspace user groups) | |
| (3) <i>Reduction of the level of potential consequences</i> | Established procedures for responding to an emergency; frangible aircraft; not flying in areas where there are aircraft with a high consequence value (e.g., commercial passenger aircraft) | |

- D. *Services* – third-party air traffic separation services that provide situational awareness and separation management services to the UAS and/or other air traffic
- E. *Strategic* – ongoing strategies that improve the effectiveness or proficiency of the UAS crew in managing the risk of midair collisions or build a general awareness of UAS operations

The subcategories of A and B comprise technological and operational strategies that help to provide an *alerted see-and-avoid* environment and can be further



Fig. 92.6 An example mitigation technology: the INSITU Pacific Mobile Aircraft Tracking System with communications, primary radar, and ADS-B In (Wilson 2012) (Image courtesy of Dr Michael Wilson)



Fig. 92.7 An example mitigation technology: the Australian Research Centre for Aerospace Automation (ARCAA) electro-optical sense-and-act system fitted onto the wing strut of the ARCAA flight test aircraft (Lai et al. 2012) (Image courtesy ARCAA)

categorized based on whether the additional situational awareness is achieved through active transmission or not. Some mitigation strategies can be assigned to more than one category, and it is important to note that some of the proposed mitigation technologies are concepts still under development; their suitability as effective mitigation strategies has yet to be determined.

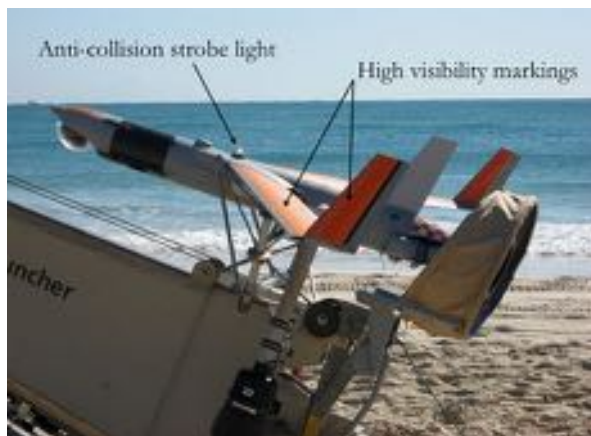


Fig. 92.8 An example mitigation technology: the INSITU Pacific ScanEagle™ on launcher with high-visibility markings and strobes (Image courtesy INSITU Pacific Ltd)

92.6.2.2 Risk Mitigation Strategies for an Impact with Terrain

A range of strategies can be used to mitigate the risks associated with the hazard of a controlled or uncontrolled impact with terrain or objects on the terrain. Some example strategies are summarized in Table 92.10. The example approaches summarized in Table 92.10 are classified based on whether the reduction in risk is achieved through (1) the elimination of the hazard, (2) a reduction in the likelihood the hazard occurs, or (3) a reduction in the consequence given the occurrence of the hazard. Category 2 mitigation strategies are further divided into the subcategories of operational, technological, and strategic. The strategies summarized in Table 92.10 are in addition to those that improve the airworthiness of the system (e.g., the adoption of sound engineering design practices, fault-tolerant design principles, the certification of software to high levels of assurance, the implementation of quality control in manufacturing processes, increasing the depth and frequency of preventative maintenance cycles, completion of pre-flight checks, and procedures and policies for crew management, training and licensing).

92.6.3 The Selection of Mitigation Strategies

ICAO (2009) evaluates mitigation strategies on the basis of their effectiveness (in terms of risk reduction), associated costs and benefits, practicality, whether they create new problems (e.g., introduce new risks), and other factors such as whether they stand up to scrutiny, the acceptability to other stakeholders, whether they are enforceable or durable, and whether the residual risks can be further reduced.

Table 92.10 Example strategies for mitigating the risks of a controlled or uncontrolled impact with terrain or objects on the terrain

| | | |
|--|----------------------|---|
| <i>(1) Elimination of the hazard</i> | | Not conducting the operation |
| <i>(2) Reduction in the likelihood of a hazard occurring</i> | <i>Operational</i> | Isolating UAS operations to designated and controlled ranges where there are no people or property exposed; minimizing/avoiding the overflight of people and property, or limiting operations to areas of low population density; operating over the oceans and away from known fishing areas or shipping lanes; establishing designated recovery or ditching points; flying at night when people are more likely to be sheltered; ability to operate under more than one mode of operation (e.g., autonomous or manual remote operation) |
| | <i>Technological</i> | Automated recovery systems capable of flying to preprogrammed recovery sites; emergency forced landing systems (e.g., Mejias et al. 2009); failure warning systems (e.g., icing or fuel warnings, breach of operational boundaries); controlled ditching in preprogrammed areas; containment systems (e.g., automated <i>fencing</i> , parachute, ditching, or explosive termination systems) |
| | <i>Strategic</i> | survey and crew familiarization with operating environment; crew training in failure and emergency procedures; general awareness (briefing local population) |
| <i>(3) Reduction of the level of potential consequences</i> | | Sheltering of people or assets; frangible aircraft; energy dissipating flight profiles (manual or pre-programmed); air bags; parachute systems; avoiding areas with the potential for consequences of <i>high value</i> (e.g., areas with hospitals, schools, or areas of high population density); personal protective equipment (e.g., helmet and eye protection – for micro/small UAS operations); established emergency procedures; emergency response equipment (e.g., first aid, environmental spill kits, fire fighting, and personnel protective equipment for post accident cleanup) |

92.6.3.1 Effectiveness and the General Hierarchy of Mitigation Strategies

The effectiveness of a mitigation strategy is measured in terms of the magnitude of the reduction in risk achieved. The most effective strategy is to eliminate the hazard, followed by those strategies that reduce the severity of the hazard or the likelihood of its occurrence. The third most effective strategies are those that employ engineering controls preventing the mishap from occurring, followed by warning devices, and procedures and training (DoD 2010a).

Effectiveness of Midair Collision Avoidance Mitigation Strategies

The most effective strategy is to segregate UAS from other airspace users; however, due to issues of practicality and cost, this is not always a viable treatment option. Those safety cases that are primarily based on the situational awareness of other airspace users (e.g., *be seen*, Table 92.9) or strategies that reduce the level of consequence given the occurrence of a mishap are the least effective and, on their

own, are not likely to provide an acceptable safety case. Reducing exposure (e.g., *staying away*, Table 92.9) in combination with other *see* and *be seen* mitigation strategies is likely to provide the most effective approach for managing the risk of a midair collision. In assessing the effectiveness of the different strategies, consideration should be given to the following:

- Types of airspace users that are likely to be encountered and their:
 - Resilience to damage due to a collision with the particular type of UAS (e.g., bird strike protection of transport category aircraft)
 - *Observability* to the different sensing or awareness approaches that could be used (e.g., radar cross-sectional area)
 - Equipage (e.g., whether they have radios or transponders onboard)
 - Ability to detect the UAS
 - Ability to maneuver
 - Typical operating speeds (e.g., determination of closing speeds and time to react)
 - Conditions of right of way
- Operating conditions (e.g., instrument meteorological conditions vs. visual meteorological conditions) or the operational profile flown (e.g., variation in radar clutter performance with altitude)
- Geographical volumes over which protection or awareness needs to be provided
- Temporal changes (e.g., use of strobes during the day vs. at night) and the duration of activity (e.g., effectiveness of ground observers for extended missions)

Effectiveness of Mitigation Strategies for Managing the Risks to People and Property on the Ground

The most effective mitigation strategies for mitigating the risks to people and property on the ground are those that reduce the following:

- Probability of a flight critical failure or human error occurring (e.g., through fault-tolerant design, maintenance, crew resource management, and training)
- Exposure of people and property to the hazard. Specifically the *operational* mitigations in Table 92.10 that restrict UAS operations to uninhabited areas or avoid/minimize the overflight of densely populated regions, critical infrastructure or culturally sensitive sites

Automated emergency flight termination systems and recovery systems are effective but not for all failure modes (e.g., typically only those for failure modes where there is still a degree of control over the flight path of the UA). Least effective are those strategies that rely solely on the general public being sheltered, wearing personal protective equipment, or emergency equipment and procedures employed following a mishap.

Effectiveness of CPA Mitigations

It is important to note that a risk control strategy that is effective for CPA may not be effective for UAS. For example, a number of studies have been conducted to evaluate the effectiveness of ACAS as a means for self-separation, collision

avoidance, or situational awareness for UAS (FAA 2011a). These studies identified a number of technical and operational issues, which have a significant impact on the effectiveness of ACAS as a midair collision avoidance system for UAS.

Evaluating Layers of Mitigations

Seldom will a single mitigation strategy be sufficient for a risk to be considered tolerable. In evaluating the cumulative effectiveness of multiple mitigation strategies, it is important to consider the potential for one strategy to degrade the effectiveness of another, thus forcing a reevaluation of the residual risks. The selection of strategies should ensure coverage of the complete spectrum of risk scenarios (e.g., implementing strategies that are only effective under visual meteorological conditions) and how the mitigation strategies, in isolation and in combination, can be overcome or can fail. The selection of strategies should try to ensure that these failure modes are not common to all of the strategies adopted.

92.6.3.2 Practicality

The practicable feasibility of mitigation strategies needs to be considered in relation to the physical and performance limitations of the system. For example, there are fundamental limits in relation to the maximum takeoff weight, payload volume, and power available to support mitigation systems onboard an unmanned aircraft. Similarly, there are fundamental limits in relation to the maneuverability, speed, range, endurance, glide performance, or ceiling of the UAS.

92.6.3.3 The Costs and Benefits Associated with Treatment Options

Costs should be considered in relation to a broad range of stakeholder groups (e.g., existing airspace users, air service providers, NAAs, the UAS industry, and ultimately, the general public) and include the indirect costs beyond those immediately associated with the occurrence of an accident (e.g., beyond the compensation for loss of life, the damage to property, and fines). Take for example, the mitigation strategies of (a) the use of redundant flight critical systems and (b) the equipage of a collision avoidance system. These mitigation options can result in increased:

- *Platform costs* due to the direct added cost of the collision avoidance system or the use of duplicate subsystems and the increased costs incurred in the engineering design, manufacture, and quality control processes
- *Operational costs* due to additional personnel training (e.g., in the operation of the collision avoidance system)
- *Through-life costs* due to additional maintenance
- *Mission costs* due to reductions in the following:
 - Performance of the system (e.g., the extra weight and drag and its impact on endurance, range, speed, or ceiling)
 - Ability for the UAS to support payloads (e.g., less weight, volume, and power available for payloads)
 - Subsequent ability of the UAS to meet mission objectives

- *Market cost* due to a reduction in the number of serviceable clients
- *Reduction in benefits* with respect to foregone downstream benefits to end users and the broader society

As well as costs, there can be indirect benefits associated with the implementation of mitigation strategies. For example, improving the overall reliability of the UAS can lead to the benefits of a lower platform attrition rate, a reduction in insurance premiums, an increase in availability, and an increase in customer trust and in turn repeat business. These costs and benefits, along with the direct benefit of a reduction in the risks, need to be factored into the determination of ALARP.

92.6.3.4 Other Factors

Mitigation strategies for UAS should be assessed to determine whether they introduce new risks and whether these introduced risks warrant treatment or whether they outweigh the benefits of the employing the mitigation strategy altogether (e.g., explosive flight termination systems). The selection of mitigation options can also be guided by secondary objectives, values, and constraints held by different stakeholder groups. For example, the FAA (2011b) explicitly preclude treatment options that reduce the operational freedoms of other airspace users (e.g., the designation of airspace specifically for use by UAS). Another example is the military preference for passive midair collision avoidance systems due to the requirement to reduce the observability of military UAS operations. External constraints can include applicable standards and regulations (e.g., existing aviation safety, environmental protection, or occupational health and safety regulations) or constraints imposed by insurance providers.

92.6.4 Summary

The selection of mitigation options for UAS is a complex decision-making process. Mitigation options must be evaluated in terms of their effectiveness, costs, benefits, practicalities, and other factors to determine whether their implementation is reasonably practicable or not. This decision process is guided by the ratio of the costs and benefits associated with pursuing the different options for mitigation. A risk is considered ALARP if this ratio is in *gross disproportion*, a concept which is subjective and variable. Finally, a determination of ALARP does not make a risk tolerable. For every scenario, a decision must be made by the organization as to whether it is willing to retain the residual risk in return for the benefits of the operation. Authorization should be obtained at two stages in the treatment process: (1) at the point of approving the selection of mitigation strategies and the decision to retain residual risks and (2) to verify that strategies have been implemented as described. Typically, the delegation of authority is dependent on the level of residual risk that is being retained.

Currently, operational mitigation strategies (e.g., restrictions on the flight of UAS over populous areas) are central to obtaining operational approvals. Miti-

gation technologies like sense-and-avoid and automated emergency landing systems, are currently under development and showing much promise. These mitigation technologies will reduce the need for restrictions on UAS operations and will be key to the uptake of UAS in a greater number of civil applications. These technologies also have the potential to greatly improve the safety of CPA operations.

92.7 Monitor and Review

Risk is dynamic. Key to maintaining and improving the SRMP is a process to monitor and review the SRMP in response to changes in the risk. Risks evolve with changes in the organization, technology, and operations performed and in the natural, social, regulatory, and political environments. Further, there can be opportunities to improve the safety risk management of existing activities if new information, assessment tools, or treatment options become available.

92.7.1 The Importance of Accident and Incident Recording

One of the primary triggers for an ad hoc review of the safety risk management of an activity is the occurrence of an accident or incident. Accident and incident data are a valuable source of information that can be used to identify new risk scenarios and update risk assessments. Most importantly, an analysis of accidents and incidents provides organizations with the opportunity to evaluate the effectiveness of their mitigation strategies and to put in place new measures to further reduce the risks.

The definition of accidents and incidents and the conditions for their reporting depend on the particular state in which the accident occurs. The National Transportation Safety Board (NTSB) in the USA defines an unmanned aircraft accident as the following:

“an occurrence associated with the operation of any public or civil unmanned aircraft system that takes place between the time that the system is activated with the purpose of flight and the time that the system is deactivated at the conclusion of its mission, in which: (1) Any person suffers death or serious injury; or (2) The aircraft has a maximum gross take-off weight of 300 pounds or greater and sustains substantial damage.” p. 600, 49 CFR §830.2 (GPO 2010)

Mandatory reporting of accidents involving UAS in the USA only formally came into force in October 2010 [amendments to title 49 CFR 830 (GPO 2010)]. FAA accident and incident reporting requirements were in force prior to this date and were mandated under the conditions of a certificate of waiver or authorization (FAA 2011b). Annex 13 to the Chicago Convention was amended in November 2010 to include the investigation of accidents and serious incidents involving international civil UAS operations but only for those UAS with design and/or operational approval (ICAO 2011).

92.7.2 Triggers for Review

The occurrence of an accident or incident as a trigger for a review is a reactive approach to safety management. A proactive strategy does not wait for an accident or incident to occur in order to trigger a review of the SRMP. Reviews can be periodic or triggered by certain conditions (e.g., a change in operations, operating environment, regulations, applications, operational types, business activity). Identified risks need to be continually reviewed to ensure that the level of risk has not changed, that mitigations are still effective, that stakeholder expectations are still being satisfied, to determine if new options for risk mitigation are available, or to determine whether there is new information or tools available that can be used to improve the assessment of the risks. Reporting mechanisms should be established that allow the organization to identify and track emerging risks.

92.7.3 Tracking Safety Performance

Measuring and tracking the safety performance of an activity or organization is part of the overarching SMS or SPP established by the organization or NAA, respectively. In most cases, accidents are extremely rare events, and hence, a proactive safety performance management strategy is needed. Such a strategy attempts to estimate the safety performance through the use of a variety of safety performance indicators or measures of lead indicator events (e.g., recording and tracking the number of breaches in policies or procedures, issues detected as part of the preflight inspection of an aircraft, as opposed to counts of accidents).

92.8 Communication and Consultation

The risk communication and consultation process is described as the “continual and iterative processes that an organization conducts to provide, share or obtain information and to engage in dialogue with stakeholders regarding the management of risk” (ISO 2009). Communication and consultation is key to avoiding potential conflict in the safety decision-making process, for ensuring that stakeholder concerns are being addressed, and for reducing uncertainty in the decisions and outcomes. This process is undertaken at all stages of the SRMP. Key to addressing issues of trust and uncertainty is ensuring transparency in the SRMP to the different stakeholders. Both the outcomes from the SRMP and the SRMP itself need to be communicated to stakeholders. It is also important to note that the different stakeholders will have different information needs. The right information needs to be communicated to the right stakeholder and in a method and manner that is acceptable and comprehensible to them. Finally, communication and consultation is a bidirectional process. Eliciting domain knowledge from stakeholders can

significantly improve the SRMP by reducing uncertainty and ensuring a more comprehensive management of the risks. Expert domain knowledge can be used at all stages in the SRMP (i.e., risk identification, analysis, evaluation, and treatment).

92.9 Conclusion

This chapter has highlighted many of the unique issues and challenges associated with the application of the safety risk management process to UAS. These issues and challenges can be technical, operational, economic, political, and social in nature and can influence all facets of the safety risk management process. Some sections of this chapter pose more questions than they do answers, highlighting that there is still much to be learned. The area of greatest need is in developing an understanding of the broader perceptions, beliefs, and expectations of society and how these factors influence decisions in relation to the safety of UAS operations. The challenges and issues discussed in this chapter are, in general, not unique to UAS. Challenges of a similar nature will need to be addressed in the safety risk management of other emerging aviation sectors such as reusable space launch vehicles, personal air vehicles, and hypersonic aircraft. It is hoped that the general processes developed and the lessons learned in the safety risk management of UAS will help to pave the way for these and other emerging and highly beneficial aviation sectors.

While this chapter has highlighted many issues, it is important to note that UAS are being safely operated in civil airspace today. In Australia, an approval to operate is obtained through the presentation of a suitable safety case to CASA, a safety case underpinned by a safety risk management process. Addressing the issues identified in this chapter will be pivotal to reducing the uncertainty in these safety cases, for ensuring consistency in the regulation of the industry, and for supporting the definition of more prescriptive safety regulations.

Acknowledgments The authors would like to thank Dr Neale Fulton, adjunct professor at Queensland University of Technology, Mr. Brendan Williams and Dr Michael Wilson from Boeing Research & Technology, Australia, Mr. Jim Coyne and Mr. Phil Presgrave from the Civil Aviation Safety Authority, and Mr. Kim Jones for their valuable comments and additions to this chapter.

References

- ADF, AAP7001.048(AM1), *ADF Airworthiness Manual* (Australian Defence Force (ADF), Directorate General Technical Airworthiness, Canberra, Australia, 2009)
- AIB, *RQ-4A Global Hawk UAV Accident Investigation, Executive Summary* (2000). Retrieved 8 Nov 2011 from http://usaf.aib.law.af.mil/ExecSum2000/RQ-4A_Edwards.6Dec99.pdf
- B.J.M. Ale, Tolerable or acceptable: a comparison of risk regulation in the United Kingdom and in the Netherlands. *Risk Anal.* **25**(2), 231–241 (2005)
- Boeing, *Statistical Summary of Commercial Jet Airplane Accidents, Worldwide Operations 1959–2010* (Aviation Safety, Boeing Commercial Airplanes, Seattle, 2011)

- CAA, *CAP 722 Unmanned Aircraft System Operations in UK Airspace – Guidance* (CAP 722, Civil Aviation Authority (CAA), The Stationary Office London, 2010a)
- CAA, *CAP 760 Guidance on the Conduct of Hazard Identification, Risk Assessment and the Production of Safety Cases* (UK Civil Aviation Authority, The Stationary Office, London, 2010b)
- CASA, *AC101-1(0) Unmanned Aircraft and Rockets, Unmanned Aerial Vehicle (UAV) Operations, Design Specification, Maintenance and Training of Human Resources* (AC101-1(0), Civil Aviation Safety Authority (CASA), Canberra, 2002)
- CASA, *Cost Benefit Analysis Procedures Manual* (Civil Aviation Safety Authority (CASA), Canberra, 2010)
- CASA, *AC101-8 Unmanned Aircraft Systems – Safety Management (Draft)* (Canberra, Civil Aviation Safety Authority (CASA), 2011)
- R.A. Clothier, R.A. Walker, Determination and evaluation of UAV safety objectives, in *21st International Unmanned Air Vehicle Systems Conference*, Bristol, United Kingdom, 2006
- R.A. Clothier, R.A. Walker et al., A casualty risk analysis for unmanned aerial system (UAS) operations over inhabited areas, in *Twelfth Australian International Aerospace Congress (AIAC-12), 2nd Australasian Unmanned Air Vehicles Conference*, Melbourne, Australia, 2007
- R.A. Clothier, N.L. Fulton et al., Pilotless aircraft: the horseless carriage of the twenty-first century? *Risk Res.* **11**(8), 999–1023 (2008)
- R.A. Clothier, J.L. Palmer et al., Definition of airworthiness categories for civil unmanned aircraft systems (UAS), in *27th International Congress of the Aeronautical Sciences (ICAS)*, Nice, France, 2010
- R.A. Clothier, J.L. Palmer et al., Definition of an airworthiness certification framework for civil unmanned aircraft systems. *Saf. Sci.* **49**(6), 871–885 (2011)
- K. Dalamagkidis, K.P. Valavanis et al., On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the National Airspace System. *Prog. Aerosp. Sci.* **44**(7–8), 503–519 (2008)
- M.T. DeGarmo, *Issues Concerning Integration of Unmanned Aerial Vehicles in Civil Airspace*. MP 04W0000323 (Center for Advanced Aviation System Development, MITRE Corporation, McLean 2004)
- DoD, *MIL-STD-882D Department of Defense Standard Practice, System Safety, Environment, Safety, and Occupational Health Risk Management Methodology for Systems Engineering, Draft incorporating Change 1* (U.S. Department of Defense (DoD), 2010a)
- DoD, *U.S. Army Unmanned Aircraft Systems Roadmap 2010–2035* (U.S. Army UAS Center of Excellence, U.S. Department of Defense, Fort Rucker, Alabama, 2010b)
- J.A. Drezner, R.S. Leonard, *Innovative Development: Global Hawk and DarkStar. Flight Test in the HAE UAV ACTD Program* (RAND, Santa Monica, 2002)
- EASA, *E.Y01301, Policy Statement Airworthiness Certification of Unmanned Aircraft Systems (UAS)* (Rulemaking Directorate, European Aviation Safety Agency (EASA), 2009)
- EUROCONTROL, *Specifications for the Use of Military UAVs as Operational Air Traffic Outside Segregated Airspace* (EUROCONTROL-SPEC-0102, EUROCONTROL, Brussels, Belgium, 2007)
- FAA, *FAA System Safety Handbook* (Federal Aviation Administration (FAA), Department of Transportation, Washington, 2000)
- FAA, Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS). Report for the FAA sponsored “Sense and Avoid” workshop federal aviation administration (FAA), Department of Transportation, Washington DC, USA (2009)
- FAA, *Evaluation of Candidate Functions for Traffic Alert and Collision Avoidance System II (TCAS II) on Unmanned Aircraft System (UAS)* (Aviation Safety, Flight Standards Service, Unmanned Aircraft Program Office, Federal Aviation Administration (FAA), Washington, 2011a)

- FAA, JO 7210.766, *Unmanned Aircraft Operations in the National Airspace System (NAS)*. JO 7210.766 (Unmanned Aircraft Systems Group, Federal Aviation Administration (FAA), U.S. Department Of Transportation, Washington, 2011b)
- FAA and EUROCONTROL, *FAA/EUROCONTROL ATM Safety Techniques and Toolbox* (Federal Aviation Administration (FAA) and EUROCONTROL, 2007)
- B. Fischhoff, P. Slovic et al., How safe is safe enough? A psychometric study of attitudes towards technological risks and benefits. *Policy Sci.* **9**(2), 127–152 (1978)
- S. Fitzpatrick, Australian spy plane crashes into Timorese home. *The Australian* (2007). Retrieved 8 Nov 2011, from <http://www.news.com.au/top-stories/australian-spy-plane-crashes-into-timorese-home/story-e6frfkp9-111113506458>
- C. Fraser, D. Donnithorne-Tait, An approach to the classification of unmanned aircraft, in *Bristol International Unmanned Aerial Vehicle Systems (UAVS) Conference*, Bristol, UK, 2011
- N.L. Fulton, Regional airspace design: a structured systems engineering approach. PhD Dissertation, The University of New South Wales, Australian Defence Force Academy, 2002
- GPO, 49 CFR 830 – *Notification and Reporting of Aircraft Accidents or Incidents and Overdue Aircraft, and Preservation of Aircraft Wreckage, Mail, Cargo, and Records*. GPO Federal Digital System (2010). Retrieved on 8 Nov from: <http://www.gpo.gov/fdsys/>, U.S. Government Printing Office (GPO), pp. 599–602
- F. Grimsley, Equivalent safety analysis using casualty expectation approach, in *AIAA 3rd “Unmanned Unlimited” Technical Conference, Workshop and Exhibit*, Chicago, 2004
- A. Hobbs, Unmanned aircraft systems, in *Human Factors in Aviation*, ed. by E. Salas, D. Maurino (Academic, Burlington, 2010)
- A. Hobbs, H.R. Stanley, *Human Factors in the Maintenance of Unmanned Aircraft* (Unmanned Aerial Vehicles Human Factors, Program Review, Federal Aviation Administration (FAA), U.S. Department of Transportation, Washington, 2005)
- N. Hodge, U.S. Says Drone, Cargo plane collide over Afghanistan. *Wall Str. J. Online* (2011). Retrieved 4 Nov 2011 from <http://online.wsj.com/article/SB10001424053111903480904576512081215848332.html>
- HSE, *The Tolerability of Risk From Nuclear Power Stations* (Health and Safety Executive, HMSO, London, 1992)
- HSE, *Principles and Guidelines to Assist HSE in Its Judgements that Duty-Holders have Reduced Risk as Low as Reasonably Practicable* (Health and Safety Executive Online Guidance Material, Health and Safety Executive (HSE), London, 2001a)
- HSE, *Reducing Risks, Protecting People. HSE’s Decision-Making Process* (Health and Safety Executive (HSE), Her Majesty’s Stationery Office (HMSO), Norwich, 2001b)
- E. Hull, K. Jackson et al., *Requirements Engineering* (Springer, Dordrecht, 2011)
- ICAO, *Safety Management Manual (SMM)*, Doc 9859 (International Civil Aviation Organization (ICAO), Montréal, 2009)
- ICAO, *Unmanned Aircraft Systems (UAS) Circular, CIR 328, AN/190*. CIR 328, AN/190 (International Civil Aviation Organization (ICAO), Montréal, 2011)
- ISO, *Risk Management – Principles and Guidelines*. ISO 31000:2009 (International Organization for Standardization (ISO), Geneva 2009)
- JAA/EUROCONTROL, Final report a concept for European regulations for civil unmanned aerial vehicles (UAVs), The Joint JAA/EUROCONTROL Initiative on UAVs, 2004. Brussels, Belgium. Available online: http://www.easa.europa.eu/rulemaking/docs/npa/2005/NPA.16.2005_Appendix.pdf
- M. Jones-Lee, T. Aven, ALARP—What does it really mean? *Reliab. Eng. Syst. Saf.* **96**(8), 877–882 (2011)
- P. La Franchi, *EUFOR Details Belgian B-Hunter UAV Crash that Caused Civilian Death*. *Flight International* (2006a). Retrieved 8 Nov 2011, from <http://www.flightglobal.com/articles/2006/10/06/209716/eufor-details-belgian-b-hunter-uav-crash-that-caused-civilian.html>
- P. La Franchi, *Incidents Between UAVs and Helicopters in Afghanistan and Iraq Prompt Action*. *Flight International* (2006b). Retrieved 8 Nov 2011, from <http://www.flightglobal.com/>

articles/2006/03/14/205379/animation-near-misses-between-uavs-and-airliners-prompt-nato-low-level-rules.html

- J. Lai, J. Ford et al., See and avoid using on board computer vision, in *Sense and Avoid in UAS: Research and Applications*, ed. by A. Plamen (Wiley, Hoboken, 2012)
- C.W. Lum, B. Waggoner, A risk based paradigm and model for unmanned aerial systems in the national airspace, in *AIAA Infotech@Aerospace Conference and Exhibit 2011*, St. Louis, Missouri, USA, 2011
- C.W. Lum, K. Gauksheim et al., Assessing and estimating risk of operating unmanned aerial systems in populated areas, in *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Virginia Beach, Virginia, 2011
- MAA, *Regulatory Articles 1000 Series: General Regulations, RA 1000 Series (GEN)* (Military Aviation Authority (MAA), UK Ministry of Defence (MoD), United Kingdom, 2011)
- S.L. MacSween-George, A public opinion survey – unmanned aerial vehicles for cargo, commercial, and passenger transportation, in *AIAA “Unmanned Unlimited” Systems, Technologies, and Operations Conference*, San Diego, California, 2003
- S.D. Manning, C.E. Rash et al., The role of human causal factors in U.S. army unmanned aerial vehicle accidents, USAARL Report No. 2004–11, Aircrew Health and Performance Division, U.S. Army Aeromedical Research Laboratory (UAARL), U.S. Department of Defense (2004)
- J.S. McCarley, C.D. Wikens, Human factors implications of UAVs in the National Airspace. Technical Report AHFD-05-05/FAA-05-01, Aviation Human Factors Division Institute of Aviation, University of Illinois, Savoy, Illinois, USA, 2005
- L. Mejias, D.L. Fitzgerald et al. Forced landing technologies for unmanned aerial vehicles: towards safer operations, in *Aerial Vehicles*, ed. by L. Thanh Mung (In-Tech, Kirchengasse, 2009), pp. 413–440
- R.E. Melchers, On the ALARP approach to risk management. *Reliab. Eng. Syst. Saf.* **71**(2), 201–208 (2001)
- G. Mortimer, *Schiebel S-100 Crash Kills Engineer in South Korea* (2012). Retrieved 31 May 2012 from <http://www.suasnews.com/2012/05/15515/schiebel-s-100-crash-kills-engineer-in-south-korea/>
- NTSB, *Accident Brief CHI06MA121* (National Transportation Safety Board (NTSB), 2007). Retrieved 8 Nov 2011 from <http://www.nts.gov/ntsb/GenPDF.asp?id=CHI06MA121&rpt=fi>
- OSD, *Unmanned Aerial Vehicle Reliability Study* (Office of the Secretary of Defense, U.S. Department of Defense, 2003)
- R. Parasuraman, T.B. Sheridan et al., A model for types and levels of human interaction with automation. *IEEE Trans. Syst. Man Cybern. A* **30**(3), 286–297 (2000)
- E. Paté-Cornell, Quantitative safety goals for risk management of industrial facilities. *Struct. Saf.* **13**(3), 145–157 (1994)
- RCC, *Range Safety Criteria for Unmanned Air Vehicles*. Document 323–99 (Range Safety Group, Range Commanders Council, White Sands, New Mexico, 1999)
- RCC, *Range Safety Criteria for Unmanned Air Vehicles, Rationale and Methodology Supplement*. Supplement to Document 323–99 (Range Safety Group, Range Commanders Council, White Sands, New Mexico, 2001)
- RCC, *Common Risk Criteria Standards for National Test Ranges: Supplement Standard 321–07*. Document 321–07 (Range Safety Group, Range Commanders Council, White Sands, New Mexico, 2007)
- RTCA, *DO-178B Software Considerations in Airborne Systems and Equipment Certification* (RTCA, Washington DC, USA, 1992)
- RTCA, *DO-304, Guidance Material and Considerations for Unmanned Aircraft Systems*. DO-304 (RTCA, Washington DC, USA 2007)
- SAE, *ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment* (Aircraft and Systems Development and Safety Assessment Committee, Society Automotive Engineers (SAE), 1996)

- SAE, *ARP5580 Recommended Failure Modes and Effects Analysis (FMEA) Practices for Non-Automobile Applications* (Reliability Committee, Society Automotive Engineers (SAE), 2001)
- S. Siddiqui, *Disaster Averted: Navy's Unmanned Aircraft Crashes After 'hitting bird'*. The Express Tribune (2011). Retrieved 8 Nov 2011 from <http://tribune.com.pk/story/212919/small-plane-crashes-near-oil-refinery-in-korangi/>
- P. Slovic, Perception of risk. *Science* **236**(4799), 280–285 (1987)
- P. Slovic, Trust, emotion, sex, politics, and science: surveying the risk-assessment battlefield. *Risk Anal.* **19**(4), 689–701 (1999)
- P. Slovic, B. Fischhoff et al. Rating the risks. *Environment* **21**(3), 14–20 (1979)
- R.A. Stephens, W.W. Taslon et al. *System Safety Analysis Handbook* (System Safety Society U.S., New Mexico Chapter, Albuquerque, 1997)
- SUAS, *Comprehensive Set of Recommendations for sUAS Regulatory Development* (Small Unmanned Aircraft System (sUAS) Aviation Rule-making Committee (ARC), Federal Aviation Administration (FAA), Washington, 2009)
- B. Thomé, *Systems Engineering: Principles and Practice of Computer-Based Systems Engineering* (Wiley, New York, 1993)
- A.P. Tvaryanas, W.T. Thompson et al., The U.S. military unmanned aerial vehicle (UAV) experience: evidence-based human systems integration lessons learned, in *Strategies to Maintain Combat Readiness during Extended Deployments – A Human Systems Approach* (NATO Research and Technology Organisation, Neuilly-sur-Seine, 2005)
- D. Washington Valdez, D. Borunda, Mexican drone crashes in backyard of El Paso home. *El Paso Times* (Online) (2010). Retrieved 8 Nov 2011 from http://www.elpasotimes.com/ci_16875462
- R. Weibel, R. Hansman, Safety considerations for operation of different classes of UAVs in the NAS, in *3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit* (American Institute of Aeronautics and Astronautics, Chicago, Illinois, 2004)
- E. Wiklund, *Flying with Unmanned Aircraft (UAVs) in Airspace Involving Civil Aviation Activity Air Safety and the Approvals Procedure* (English translation of "Flygning med obemannade luftfartyg (UAV) ilufterum med civil flygverksamhet") (The Swedish Aviation Safety Authority, Norrköping, Sweden, 2003)
- K.W. Williams, *A Summary of Unmanned Aircraft Accident/Incident Data: Human Factors Implications*. DOT/FAA/AM-04/24 (Civil Aerospace Medical Institute, Federal Aviation Administration, Oklahoma City, 2004)
- M. Wilson, The use of low-cost mobile radar systems, for small UAS sense and avoid, in *Sense and Avoid in UAS: Research and Applications*, ed. by A. Plamen (Wiley, Hoboken, 2012)
- R. Wolfe, *Why Demonstrating an "Equivalent Level of Safety" for See and Avoid is an Inappropriate Requirement for Unmanned Aircraft System Operations* (Modern Technology Solutions Incorporated (MTSI), Alexandria, 2009)
- E. Zio, N. Pedroni, *Uncertainty Characterization in Risk Analysis for Decision-Making Practice*. Number 2012–07 (Cahiers de la Se'curite' Industrielle, Foundation for an Industrial Safety Culture, Toulouse, 2012)

Ron van de Leijgraaf

Contents

| | | |
|--------|--|------|
| 93.1 | Introduction | 2278 |
| 93.2 | Aeronautical Products | 2279 |
| 93.3 | Certification Process | 2279 |
| 93.3.1 | Process Description | 2280 |
| 93.3.2 | Procedure to Deal with Novel Design Features | 2282 |
| 93.3.3 | Certification and Validation Projects | 2282 |
| 93.4 | Certification Safety Requirements | 2283 |
| 93.4.1 | Certification Specifications | 2283 |
| 93.4.2 | European Technical Standard Orders (ETSO) | 2284 |
| 93.5 | Relation Between Safety Requirements and RPAS Components | 2286 |
| 93.5.1 | Remotely Piloted Aircraft (RPA) | 2286 |
| 93.5.2 | Remote Pilot Station (RPS) | 2286 |
| 93.5.3 | Command, Control, and Communication System | 2287 |
| 93.5.4 | Other Systems | 2288 |
| 93.5.5 | Detect and Avoid | 2288 |
| 93.5.6 | Concluding Remarks | 2289 |
| 93.6 | Certification Organizational Requirements | 2289 |
| 93.7 | Final Remarks | 2290 |
| | References | 2291 |

Abstract

This chapter described the certification of (small) unmanned aircraft systems (UAS). It focuses on the certification process, the requirements for the safe design of a UAS, and the organizational requirements for the company designing the UAS.

R. van de Leijgraaf

Civil Aviation Authorities – The Netherlands, Airworthiness Inspectorate, Hoofddorp,
The Netherlands

e-mail: ron.vande.leijgraaf@minienm.nl

93.1 Introduction

Aviation authorities worldwide consider an unmanned aircraft (UA) an aircraft as defined by the International Civil Aviation Organization (ICAO). Since these authorities all have signed the ICAO convention, the fundamental standards and recommended practices for the integration of UAS into the airspace will be developed by ICAO. When considering the certification of small UAS, the standards are given in Annex 8 (Airworthiness) of the ICAO Chicago Convention.

ICAO has acknowledged the rapid growing market for UAS and has established in 2008 a study group to address this issue. At the moment, this UAS study group is evaluating all ICAO annexes and is preparing recommended changes to these annexes to accommodate UAS in the ICAO standards. The study group published ICAO UAS Circular 328 on unmanned aircraft systems (UAS), providing guiding information on how UAS can be introduced in the ICAO annexes to facilitate a safe introduction of UAS in the airspace worldwide. As a next step, the study group is developing detailed proposals for changes to the annexes. These changes will be written as guidance material to possible changes in the annexes. This will result in the ICAO UAS Manual, which is planned to be presented at an ICAO UAS Conference in fall 2014.

When the manual is published, it will be up to the various ICAO panels to develop the adaptations to the annexes, based on the guidance material provided by the study group. Experts from the UAS study group will most likely be involved in the work to be performed by the panels.

Autonomous UAS are not considered at the moment by ICAO in the update of the annexes. Only UAS where the remote pilot is capable of interfering with the flight path of the aircraft are considered in the ICAO Circular. Such a system, considered a subset of UAS, is defined as remotely piloted aircraft system (RPAS) by ICAO. In this chapter, the official ICAO terms and definitions, as given in Circular 328, will be used.

This chapter will only focus on the initial airworthiness of the RPAS, i.e., the certification. Continuing airworthiness, which is equally part of airworthiness, will be touched upon lightly and only in relation to the initial certification. Furthermore, this chapter will not address the issues of safety oversight of the whole airworthiness process by the national aviation authorities. When examples of the current aircraft certification practice are given, they will be primarily given in the European context. The same examples are equally valid for the American practice, the only difference is the terminology used by the European Aviation Safety Agency (EASA) and the U.S. Federal Aviation Authority (FAA).

Although ICAO is only responsible for the standards and recommended practices for international operations and operations over the high seas, its standards are usually adopted by states for their national aviation regulations. Therefore, the process and safety standards described in this chapter are applicable to international and national operations of RPAS.

93.2 Aeronautical Products

The whole concept of airworthiness is primarily aimed at developing a safe aircraft to allow safe flight operations. One would expect the topic of airworthiness only dealing with the aircraft, but within Annex 8, ICAO currently recognizes three products that need to have an airworthiness approval:

1. Aircraft
2. Engines
3. Propellers

Each product can go through the certification process as described in the next paragraph and have its design approved. When the design approval is granted, the product will have a type certificate (TC). By themselves the engine and propeller will of course not be able to fly, they still need to be integrated with the aircraft. The advantage of evaluating the engines and the propellers separate from the aircraft design is twofold:

First, it allows the manufacturer of the engines or propellers to design products that can be used on more than one aircraft design (General Electric engines that can be installed on both Airbus and Boeing aircraft, for instance).

Second, it allows the aircraft designer to be involved only with the certification of the integration of the engine and/or propeller into his aircraft, not with the actual certification of the engine or propeller itself. This seriously reduces the effort needed by the aircraft designer to go through the certification process with the aircraft design.

When studying the impact of the introduction of RPAS to the ICAO annexes, in preparation for the ICAO UAS Circular, the UAS study group concluded that the remote pilot station (RPS) should be treated as a new aeronautical product within Annex 8. Therefore, the study group will be proposing to introduce the remote pilot station (RPS) as an additional airworthiness product within Annex 8. This will allow manufacturers of RPS optimal flexibility in developing and providing their product for various RPAS designers. It will also more easily allow for the use of multiple RPS during one specific flight with an RPAS. Think about the possible operation of a long-haul flight with an RPAS and a cruise pilot using satellite communication during the cruise phase with one specific RPS and harbor pilot(s) using a different RPS and direct radio communication during take-off and landing. For small UAS, this operational concept of using multiple RPS during a flight will not be very likely. In that case, the RPS could be certificated as part of the RPAS directly, not through a separate design approval.

93.3 Certification Process

As defined in Annex 8 of the ICAO Convention, an aircraft can only be certificated by the aviation authorities of the country where the principle place of business of

the designer of the aircraft is located. In ICAO terms, the country performing the certification is the state of design. Other national aviation authorities can either fully accept the certification by the state of design, or they can do an additional investigation into the safe design of the aircraft. This additional investigation is called a “validation.” Countries that can act as a state of design usually have bilateral agreements with other countries about the certification and validation process and acceptance of certificates. In general, authorities avoid duplicating tests, so when a validation process is performed, this process focuses on the known differences in safety requirements between the state of design and the validating authority. In Europe, a regional safety oversight organization (RSOO) has been created, which has taken over most of the aviation safety responsibilities from the participating European countries. This organization is called the European Aviation Safety Agency, EASA. When in this chapter a reference is made to a country, EASA can be seen as the equal organization for the national aviation authority for the participating European countries. For instance, EASA is responsible, in the state of design role, for certification activities of Airbus and Fokker aircraft. EASA will also do the European validation process for aircraft designed in, e.g., the USA or Brazil.

In the EASA Basic Regulation (<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:2008R0216:20091214:EN:PDF>), the responsibilities of the agency regarding UAS are clearly limited. Unmanned aircraft with an operating mass of no more than 150 kg are not the responsibility of EASA, but the responsibility of the individual states.

Certification authorities around the world use a similar process when addressing the certification of an aircraft. This process is described in this paragraph for an aircraft, but as indicated in the introduction of this chapter, RPAS are considered aircraft and are treated in a similar way.

93.3.1 Process Description

A certification project for an aircraft is always started with an official application for certification by the organization responsible for the aircraft design. The organization requesting the certification is commonly referred to as the “applicant.” When the application is accepted by the aviation authority, the certification project will start. This project is normally started with a kickoff meeting where the designer of the aircraft presents the design to the complete team of specialists that have been assigned to the certification team by the authority.

The certification project follows four different phases:

1. *Establishing the Certification Basis* In this phase, the safety requirements against which the safe design of the aircraft must be proven will be defined for the project. These requirements are referred to as the “certification basis.” Normally, the designer of the aircraft is responsible for providing a first draft of this certification basis. During and after the presentation of this first draft, the discussion and interaction with the authority certification team takes place, leading to a mutually agreed and mutually accepted certification basis for the

whole of the project. The basic safety requirements for a certification project are the latest version of requirements from the state of design (usually referred to as Certification Specification (CS) by EASA and Federal Aviation Regulation (FAR) by the FAA) applicable at the moment the application for certification has been received by the authority. A certification project usually lasts several years, and it is considered unfair to the applicant to add updated requirements to the certification basis over the years, due to the further development of the safety requirements. Of course, when both the applicant and the authority agree, newer version of the requirements can be made applicable to the project.

When all safety requirements are agreed between the applicant and the certifying authority, this phase is closed.

2. *Defining the Means of Compliance* Once the safety requirements are established, the applicant and the authority have to reach agreement on how the applicant will show that his design of the aircraft is meeting these requirements. There is a variety of methods available, from expert judgment, through theoretical analysis, down to flight testing. Not only the methods of showing compliance with the requirements are agreed in this phase. The involvement of the authorities in the various compliance finding tasks will also be agreed. In general, complete test plans, documents to be delivered, etc., will be agreed between applicant and authority.

At the end of this phase, there is agreement between applicant and the certifying authority about which tests are required and which reports will have to be delivered by the applicant. Furthermore, there will be agreement about the level of involvement of the authorities in the compliance finding, e.g., which reports need to be accepted and approved by the authorities, which tests will be witnessed by the authorities, and what will be left to the responsibility of the applicant, without any authority involvement.

This phase closes when there is mutual agreement on all mentioned points.

3. *Compliance Finding* In this phase, all activities agreed between applicant and authority in the previous phase will be performed. Design documents will be developed, tests will be performed, and manuals will be written. Where needed, authority will witness testing or approve reports, as agreed in the previous phase. For all certification projects, this phase takes the most time and is the most expensive phase for the applicant.

When all the compliance finding activities have been performed and accepted by the authority, the applicant is ready for the final phase.

4. *Delivering the Type Certificate* At the end of the third phase, there is full agreement between the authority and the applicant about the safe design of the aircraft. Now the applicant can provide a statement of compliance to the authority, to indicate that all safety requirements have been met, all necessary tests have been done, and everything is to full satisfaction of the authorities. When this statement of compliance is provided, the authority then can provide the type certificate to the applicant, indicating that the aircraft is designed in accordance with the applicable safety requirements and that the aircraft can now be built and operated safely.

Although these four phases are given in sequential order, in practice these phases are not clearly separated in time. For some parts, the whole process can be straightforward, and then the process can be done in these clear steps. Other parts require extensive discussions between applicant and authority, and in this case, the process can easily jump back and forth between the various phases. The only certainty in the whole process is that the first three phases have to be fully completed before phase 4 can happen. This last phase is usually the shortest phase of the four.

A more complete description of the certification process can be found in the EASA Type Certification Procedure (<http://www.easa.europa.eu/certification/docs/internal-working-procedures/PR.TC.00001-002%20Type%20certification.pdf>).

93.3.2 Procedure to Deal with Novel Design Features

As indicated, phase 1 concerns the definition of the certification basis. Normally a standard set of safety requirements is selected as the basis. But rulemaking normally does not develop as quickly as the technical advancements in aviation. This means that in most aircraft designs, there are new features that have not yet been covered by safety regulation. A clear example of this was the first fly by wire aircraft.

Within the certification process, there are means of dealing with these novel design features. When such a new design feature is identified, the authority can develop some project-specific requirements, tailored to the specifics of the design. In EASA projects, these specific requirements are captured in so-called certification review items or CRIs, and in FAA projects, they are captured in issue papers or IPs. Normally, both the technical requirements and the accepted means of compliance against these requirements are captured in these documents. In general the novel design features are captured in the first phase, but occasionally, some features are only identified in a later stage of the design. So up to very late in the project, these CRIs or IPs can be developed.

93.3.3 Certification and Validation Projects

In large certification projects, the certification project (performed by the aviation authority of the state of design of the aircraft) and the validation project (performed by any other aviation authority) run in parallel. This saves the applicant valuable time and money and allows operation of the aircraft in all countries where the design has been approved.

For example, consider an Embraer aircraft design. The state of design in this case is Brazil. But Embraer would like to allow operation of their aircraft not only in Brazil, but worldwide. In order to enable these worldwide operations, Embraer needs approval from all other aviation authorities of countries that have signed the ICAO Convention. Luckily, most countries in the world accept either an American (FAA) or European (EASA) approval. That means that additional to the Brazilian approval, Embraer is looking for EASA and FAA approvals as well. So in that case,

there are three projects running in parallel. First, there is the certification project performed by the Brazilian authorities (Agência Nacional de Aviação Civil, ANAC), and the other two projects are validation projects with an FAA validation team and an EASA validation team.

93.4 Certification Safety Requirements

In this paragraph, the major forms of safety requirements will be described.

93.4.1 Certification Specifications

For all RPAS, the certification safety requirements still need to be developed and accepted by national authorities. In its policy on UAS certification (E.Y013-01, August 25, 2009, http://easa.europa.eu/certification/docs/policy-statements/E.Y013-01_%20UAS_%20Policy.pdf), EASA UAS policy described the methodology to develop such requirements. First, it needs to be determined which category of manned aircraft is most applicable to the remotely piloted aircraft system that will be certificated. When this is done, the associated safety requirement for manned aircraft is selected. Subsequently, this requirement needs to be adapted to be applicable to the RPAS.

For small UAS, the following two certification specifications from EASA are most appropriate:

- EASA CS-VLA for airplanes (<http://easa.europa.eu/agency-measures/docs/certification-specifications/CS-VLA/CS-VLA%20%20Amdt%201%20combined.pdf>)
- EASA CS-VLR for rotorcraft (http://easa.europa.eu/agency-measures/docs/certification-specifications/CS-VLR/MERGED_v2.pdf)

At the moment, there are a few (unofficial) certification specifications available. Firstly, the NATO FINAS group has provided a specification based on CS/FAR 23, primarily used for military purposes and not applicable to small RPAS. Secondly, there is a final draft version available of the CS-LURS (Certification Specification Light Unmanned Rotorcraft System, adapted from CS-VLR), which the JARUS group (Joint Authorities for Rulemaking on Unmanned Systems) has developed. This last group is a voluntary group of national aviation authorities, together with EASA and EUROCONTROL, that works on drafting UAS regulation.

To give an example of the sort of topics that are covered in certification specifications, the various subparts that together form the CS-LURS set of requirements are given below:

Book A:

Subpart A: General

Dealing with general requirements like applicability.

Subpart B: Flight

Dealing with flight envelope, weight, performance issues, and flight characteristics.

Subpart C: Strength requirements

Dealing with loads on the aircraft, structural, and mechanical safety and fatigue.

Subpart D: Design and construction

Dealing with materials, rotor design, (hydro-) mechanical flight control design, landing gear, fire protection, lighting protection, and construction of the airframe.

Subpart E: Powerplant

Dealing with the engine installation, rotor drive system, fuel systems, batteries for electric engines, oil system, and exhaust system.

Subpart F: Equipment

Dealing with primarily electrical, electronical and computer systems, system safety, system installation, and the airborne part of the command and control link.

Subpart G: Operating limitations and information

Dealing with all sorts of operational issues.

Subpart H: (reserved for Detect and Avoid)

Subpart I: Control station

Dealing with the remote pilot station.

Appendix A: Instructions for continued airworthiness

Appendix B: Engines

Appendix C: Test procedure for self-extinguishing materials

Book B: Acceptable means of compliance

Providing clarification of the regulations in Book A and by the authorities accepted means of showing compliance against the regulations.

93.4.2 European Technical Standard Orders (ETSO)

Technical Standard Orders (TSO) are a well-known mechanism from the manned aviation regulations to allow system manufacturers to develop approved systems irrespective of the aircraft in which it will be installed.

During the certification process, the designer of the aircraft is responsible for the total certification effort. That does not only mean that he is responsible for the fuselage, engines, wings, etc., but also for the systems that are installed in the aircraft. The designer is really acting as the system integrator of the total system that will become the aircraft.

When it comes to certification of the electrical or avionics systems onboard the aircraft, there are two ways of getting an approval. The first possibility is that the full system functionality and integration with the other onboard systems is done during the certification of the whole aircraft. The second possibility is to have the system functionality approved prior to the aircraft certification and focus on the certification of the integration with the other onboard systems during the aircraft certification. This second method provides some benefits to the system manufacturers that build these electrical or avionics systems. This described methodology is equally applicable to aircraft seats, for instance. But this is considered out of scope for UAS in this paragraph.

The system that is used for this specific functional approval is the European Technical Standard Order (ETSO, in the USA it is referred to as a Technical Standard Order or TSO). The certifying authority can establish specific functional requirements for equipment. Then, in a separate process, the manufacturer of the system can obtain approval for the functionality of the system. This approach has two distinct advantages:

1. The manufacturer of the system is able to provide his systems to different aircraft manufacturers.
2. The aircraft designer does not have to consider the functional approval of the system, only the integration of the system in the total aircraft system.

Some typical examples of these sorts of systems are VHF radios, navigation receivers, transponders, etc.

Normally, the authorities are not defining the functional requirements for the system themselves, but they work closely together with the manufacturing industry to define some acceptable functional standards. Within aviation, these standards are developed in two standardization bodies: EUROCAE (European Organisation for Civil Aviation Equipment) and RTCA (Radio Technical Commission for Aeronautics). EUROCAE is primarily European, while RTCA is primarily American. In most cases, these two bodies work together to develop a worldwide functional standard for equipment. The process by which these standards are developed is the following; the governing body of either EUROCAE or RTCA identifies the need for a new standard and drafts terms of reference for a working group (WG, EUROCAE) or special committee (SC, RTCA) to establish these new standards. Then a working group or special committee is created (most of the times these groups work together, as said earlier), and participation from members of EUROCAE or RTCA is sought. The participants are primarily from industry, but aviation authorities participate in these groups as well. Within this group, a functional standard is developed, based on consensus with all participants in the group.

When such a group has finalized its work and either EUROCAE or RTCA have published the industry functional standard, the aviation authorities adopt this standard by referring to that standard in a (E)TSO. Once the (E)TSO is published, the equipment manufacturer can apply for approval with the aviation authority. The largest part of the approval process for an (E)TSO is to show compliance with the industry standard that has been published by EUROCAE or RTCA (or both).

After obtaining this approval, the manufacturer is able to provide his equipment to an aircraft designer who is looking for that specific functionality in his aircraft design. During the certification process of the aircraft, the aircraft designer does not have to show that the system that is installed in his aircraft meets the functional requirements for that system. He can provide the (E)TSO approval of the system to the certifying authority and that is the required proof that the system meets the requirements. The aircraft designer, however, is still responsible for proving that the integration of that equipment with the other systems onboard the aircraft is still in accordance with the safety requirements for the aircraft design.

The approval for the equipment against the functional requirements requires a lot of functional testing and significant environmental testing. With this testing done at the equipment level and granting a more generic functional approval of the equipment, both the equipment manufacturer and the aircraft designer save significantly in cost and time for the certification of aircraft.

93.5 Relation Between Safety Requirements and RPAS Components

In this paragraph, the various systems that together form the RPAS will be addressed. It will look at how to relate the safety requirements given in the previous paragraph with these various parts.

93.5.1 Remotely Piloted Aircraft (RPA)

When converting a manned certification safety requirement for an RPAS, there are a few issues that need to be considered. First is the obvious issues that are directly related to not having a person onboard. This includes requirements on onboard chairs, seat belts, emergency oxygen, etc. But there also are less obvious issues that need to be considered. In general, a small UAS is a much more complex aircraft than the equivalent category of manned aircraft, especially from a systems perspective. This means that additional requirements are necessary to cover these design features, primarily the requirements in Subpart F of the various certification requirements.

Another area that usually needs additional requirements is the power plant installation section (Subpart E). Electrically driven engines and dual turbine engine installations, these kinds of design features are not covered by the equivalent manned aircraft safety requirements. Therefore, the subparts of the requirements that address these issues need special attention and most probably need a partial rewrite of the whole section.

One final topic that needs to be mentioned is the emergency control and failure warning systems. There is nobody onboard to perform a number of emergency recovery actions that are implicit in the manned aircraft safety requirements. These requirements need careful rethinking and an adaptation that allows the pilot in the remote pilot station to still be informed about possible system failures and perform some emergency control if that is required.

From the given table of contents of CS-LURS, Subparts A–G are covering the RPA requirements.

93.5.2 Remote Pilot Station (RPS)

This is a new system component of the RPAS that has no equivalent in the manned aviation regulations. In the ICAO concept as described in the UAS Circular,

the remote pilot station (RPS) will be a separate aeronautical product. This will mean that once the ICAO annexes have been adapted accordingly, there will be a separate certification safety requirement document for the RPS. At the moment, the safety requirements for the RPS are still part of the safety requirements document for the RPA. In current safety requirements documents, there is a specific subpart (Subpart I) dedicated to the RPS requirements.

Establishing the requirements for the RPS is not a matter of simply copying the requirements related to the cockpit design from manned aviation to the RPS. Of course, all human machine interfaces are part of the RPS and have a prominent place in the RPS requirements. But most of these interfaces are part of systems that have components both in the RPA and the RPS. In such a situation, it is necessary to develop a consistent strategy for dividing the requirements between the subparts for RPA and RPS. It should be avoided to duplicate requirements between the RPA and RPS. In the JARUS team that worked on the CS-LURS, the philosophy was used to include the requirements systems that had components in both the RPA and RPS part into the RPA part of the document. The requirements that refer to the RPS only are then collected in the subpart for the RPS.

From the given table of contents of CS-LURS, Subparts A and I are covering the RPS requirements.

93.5.3 Command, Control, and Communication System

The command, control, and communication (C3) system will not be certificated separately. When a radio line of sight system will be used, the airborne part and the ground-based part of the C3 system will be certificated as part of the RPA or RPS, respectively. For the safety requirements, this means that the requirements for the C3 components will be contained in the respective subparts of the certification specification. This is identical to how radio communication systems in manned aviation are currently certificated.

When a beyond radio line of sight system, e.g., satellite communication, will be used, the current proposal in the ICAO circular is to work with a certificated communication provider that is under safety oversight of the aviation authorities. It is considered impossible to certificate a satellite communication system as a fully integrated part of the RPAS. In this case, the performance requirements for the C3 system and the interfaces between RPA, RPS, and C3 system will be defined and the components of the RPA and RPS that will interface with the communication system will be included in the safety requirements of the respective subsystems.

For small UAS, in most RPAS designs, a radio line of sight system will be used, so the first approach to certificating the system, as described above, will be applicable.

From the given table of contents of CS-LURS, Subparts A, F, and I are covering the command and control requirements.

93.5.4 Other Systems

There will be very specific subsystems being designed as part of an RPAS. The first two systems that immediately come to mind are a launching and recovery system and a flight termination system. These systems are usually unknown in the manned aviation system, so specific safety requirements and associated system standards need to be developed for these. At the moment, the CS-LURS does contain some requirements on the flight termination system. Over time, the detailed safety requirements will be introduced in the certification specification. Until the time that these requirements are part of the official specifications, the authorities will be able to deal with these subsystems by introducing dedicated requirements for a certification project and document these requirements in a certification review item (EASA system) or in an issue paper (FAA system).

93.5.5 Detect and Avoid

One of the specific subsystems that are foreseen to be used in RPAS is the detect and avoid system (DAA). This system will be required to replace the collision avoidance capability that is provided by the onboard pilot in manned aviation. Another chapter in this book is fully dedicated to the DAA system, so this paragraph will only focus on the certification aspects of it. From a certification perspective, there are two issues to address.

First the functionality of the system needs to be defined. This is usually done by asking standardization organizations like EUROCAE and RTCA to define standards for the system. The aviation authorities can then accept these standards and make them part of the safety requirements by including the standard from EUROCAE and/or RTCA in a (European) Technical Standard Order ((E)TSO), which is explained in more detail in the previous paragraph. Additional to this equipment approval, some functional requirements can also be included in the certification specification.

Secondly, specific safety requirements for the system need to be developed, in line with the current practice of assuring system safety in the manned certification standards.

Various rulemaking bodies around the world are currently working together to develop both of these standards. At the time of writing of this chapter, there was no definitive regulation for this system available yet. In practice this will mean that operations of RPAS on a regular basis will be limited to visual line of sight, unless an authority has given a specific approval to a certain operational scenario to allow operations beyond visual line of sight.

From the given table of contents of CS-LURS, Subparts A and H are covering the DAA requirements.

93.5.6 Concluding Remarks

The whole area of safety requirements for small UAS is undergoing significant development at the moment. This paragraph therefore cannot address these requirements too specifically, because without a doubt these would have changed when this book is published. Therefore, the framework for certification has been given. The issues that are currently being addressed are also provided. Together this should allow understanding of the requirements once they are published by the aviation authorities.

Apart from the specifics of CS-LURS, this chapter is equally applicable to other certification specifications, ranging from airplanes (small and large) and helicopters (small and large).

93.6 Certification Organizational Requirements

Not everybody is allowed to design an aircraft, according to the aviation regulations. Before being granted this right, an organization needs to prove that it is capable of designing a safe aircraft. This whole process is referred to as Design Organisation Authorisation (DOA) approval. In the EASA Airworthiness Regulation (EASA Part 21, <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:2003R1702:20091228:EN:PDF>), the whole process of acquiring and maintaining a DOA approval is given in Subpart J.

The DOA approval process is not about demonstrating the technical ability to design an aircraft but is about the organizational capabilities to design in a structured and (safety) controlled manner. These organizational capabilities include a quality assurance system, an independent validation system to do a check on the technical design, training of relevant personnel, delegation of responsibilities within the organization, etc. So the process is aimed at assuring that the organization that is designing the aircraft is capable of:

- Designing in accordance with the QA system
- Providing an independent validation check
- Training the personnel timely and adequately
- Working in accordance with procedures that assure the quality of the work
- Appropriately delegating responsibilities within the organization

The philosophy behind the DOA approval is that the authorities can partly rely on the internal design organization procedures when certifying the aircraft. During the “defining the means of compliance” phase of the certification project, the authorities will normally agree with the applicant about which specific parts of the compliance finding can be delegated to the applicant, i.e., which reports will only be checked and validated by the applicant and which tests will be independently witnessed by the applicant only. In this case, the validation organization within the

applicants' organization can be considered to act on behalf of the authorities. That is one of the reasons why the independence of the validation team within the design organization is considered of prime importance.

93.7 Final Remarks

As said at the introduction of this chapter, UAS are aircraft. Therefore, UAS will need to fit into the currently existing aviation system, both from an airworthiness and from an operational perspective. But the introduction of UAS into civil airspace is slowly changing the way aviation regulators are thinking about airworthiness and safe operations. In manned aviation, that aircraft is the core element where everything needs to be integrated. The aircraft in itself is the end product and provides the platform where everything (fuselage, equipment, engines, etc.) are integrated. With the introduction of RPAS, the aircraft is "just" the airborne component of the total system. This means that the whole approval system and the associated safety oversight system by the national aviation authorities can no longer be aircraft centered but should be system centered. When you consider the aircraft in manned aviation as the core element where the whole system is integrated, this system-centered approach for RPAS is no different to what the authorities are used to from manned aviation. The only difference is that the system in the case of the RPAS is more than the aircraft, while in manned aviation, the system is the aircraft. In the future, this new thinking will be finding its way into ICAO Annex 8. But the fundamentals of the safety approach by ICAO, an approved design and safety oversight by countries, will not be changed.

This chapter has focused on the airworthiness issues only. Although the chapter is written with small UAS in mind, the processes and practices given in this chapter are, in general, equally valid for larger UAS. Only the specific requirements and examples given herein are tailored toward small UAS. The certification process and the layout of the certification specification is identical for larger UAS.

The authorities are looking for the differentiation between large UAS and small UAS by downscaling the requirements to an appropriate level. It is widely understood that a Global Hawk should be certificated against different requirements than a 2 kg octocopter UAS. But the whole system and process of approval should be identical.

One of the issues that the authorities are facing to deal with at the moment is that there is no manned equivalent for the certification specification for small UAS, certainly for those well below 25 kg. This leads to the following more fundamental question: To what level do the requirements for larger UAS need to be downgraded to provide safe UASs that are still economically viable? A follow-on question to this is as follows: What sort of categorization scheme can be used to classify these small UASs? Industry and authorities are currently addressing these issues.

References

- EASA Basic Regulation, Regulation (EC) No.216/2008 of the European Parliament and of the Council on common rules in the field of civil aviation and establishing a European Aviation Safety Agency. Available to download from the EASA website: www.easa.europa.eu
- EASA CS-VLA, Decision (EASA) No.2003/18/RM of the Executive Director of the Agency on certification specifications, including airworthiness codes and acceptable means of compliance for very light aeroplanes (CS-VLA). Available to download from the EASA website: www.easa.europa.eu
- EASA CS-VLR, Decision (EASA) No.2003/17/RM of the Executive Director of the Agency on certification specifications for very light rotorcraft (CS-VLR). Available to download from the EASA website: www.easa.europa.eu
- EASA Part 21, Regulation (EC) No.1702/2003 of the European Parliament and of the Council on laying down implementing rules for the airworthiness and environmental certification of aircraft and related products, parts and appliances, as well as for the certification of design and production organisations. Available to download from the EASA website: www.easa.europa.eu
- EASA Type Certification Procedure, EASA Procedure PR.TC.00001-002 Type Certification. Available to download from the EASA website: www.easa.europa.eu
- EASA UAS policy, EASA Policy Statement – Airworthiness Certification of Unmanned Aircraft Systems. E.Y013-01 25 August 2009. Available to download from the EASA website: www.easa.europa.eu
- ICAO UAS Circular, ICAO Cir 328, Unmanned Aircraft Systems (UAS). Available to purchase from the ICAO website: www.icao.int
- JARUS reference, *JARUS – Joint Authorities for Rulemaking on Unmanned System, UAS Unmanned Aircraft System – The Global Perspective, 2011*, 9th edn. (UVS International, Paris)

Technology Surveys and Regulatory Gap Analyses of UAS Subsystems Toward Access to the NAS

94

Richard S. Stansbury and Timothy A. Wilson

Contents

| | | |
|--------|--|------|
| 94.1 | Introduction | 2294 |
| 94.2 | Background..... | 2294 |
| 94.2.1 | Technology Surveys | 2295 |
| 94.2.2 | FAA Documents | 2296 |
| 94.2.3 | Regulatory Gap Analyses | 2297 |
| 94.3 | Case Study #1: Propulsion Technologies for UAS | 2298 |
| 94.3.1 | Technology Survey | 2298 |
| 94.3.2 | Regulatory Gap Analysis..... | 2304 |
| 94.4 | Case Study #2: Sense-and-Avoid Technologies for UAS | 2306 |
| 94.4.1 | Technology Survey | 2306 |
| 94.4.2 | Regulatory Gap Analysis..... | 2310 |
| 94.5 | Case Study #3: Command, Control, and Communication Technologies for UAS | 2312 |
| 94.5.1 | Technology Survey | 2312 |
| 94.5.2 | Regulatory Gap Analysis..... | 2318 |
| 94.6 | Case Study #4: Emergency Recovery and Flight Termination Systems for UAS | 2322 |
| 94.6.1 | Technology Survey | 2322 |
| 94.6.2 | Regulatory Gap Analysis..... | 2327 |
| 94.7 | Conclusion | 2333 |
| 94.7.1 | Guidance on Performing a UAS Technology Survey | 2333 |
| 94.7.2 | Guidance on Performing a Regulatory Gap Analysis..... | 2334 |
| | References | 2334 |

Abstract

To make a safe transition of UAS into the National Airspace System, new regulations must be developed by the Federal Aviation Administration. The technologies employed by UAS are in many circumstances fundamentally different

R.S. Stansbury (✉) • T.A. Wilson

Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA

e-mail: stansbur@erau.edu; wilsonti@erau.edu

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

DOI 10.1007/978-90-481-9707-1_62,

© Springer Science+Business Media Dordrecht 2015

2293

than those of traditional manned aircraft. The regulations written to support the airworthiness certification, operations, maintenance, etc. of manned aircraft often do not apply as written without interpretation, revision, and/or deletion. This chapter provides the necessary details on how to conduct a technology survey and regulatory gap analysis of UAS technology subsystems. Four past studies performed by Embry-Riddle Aeronautical University for the FAA's William J. Hughes Technology Center are discussed. These studies address UAS propulsion systems, sense-and-avoid technologies and procedures, command control and communication, and emergency recovery and flight termination systems. Each study will be discussed in this chapter, and a recommended process for future studies is provided.

94.1 Introduction

To safely transition unmanned aircraft systems (UAS) into the National Airspace System (NAS), new regulations must be developed by the Federal Aviation Administration (FAA) as will other international civil air authorities (CAAs) for their respective airspaces. There are many subtle and fundamental differences between technologies used for UAS versus traditionally manned aircraft. Before new regulations or policies can be written to support unmanned aviation, regulatory issues must be articulated.

Beginning in 2006, researchers at Embry-Riddle Aeronautical University (ERAU) began collaboration with the FAA William J. Hughes Technology Center to identify technology-based gaps within the FAA's current regulatory framework. The first study surveyed UAS propulsion technologies. Follow-on research included a regulatory gap analysis of UAS propulsion performed in 2007–2008. Three additional technology survey and gap analysis studies followed for UAS sense-and-avoid (SAA) technologies; command, control, and communication (C3) technologies; and emergency recovery and flight termination (ERFT) technologies and procedures.

The purpose of each technology survey was to articulate representative technologies and frameworks being used in current and near-future UAS. The regulatory gap analysis focuses on the alignment of the technology with existing regulations. It should allow the target audience to clearly identify and articulate where revisions or reinterpretations are required. This chapter begins with an overview of technology surveys, gap analyses, and relevant types of FAA documents. Next, the above-mentioned studies are presented as separate case studies. This chapter concludes with the authors' recommended best practices for future studies.

94.2 Background

UAS represent a near-disruptive technology for the current NAS in the United States of America and in the corresponding airspace systems of nations worldwide. While total displacement of aircraft with an onboard pilot from the NAS seems extremely unlikely at any near-future time, the introduction of remotely piloted

aircraft having varying degrees of onboard autonomy has been beyond the current regulatory framework for aircraft design and manufacture, for flight activities, and for on-ground operations and maintenance.

The introduction of disruptive or near-disruptive technologies on markets has been studied regarding technologies such as the Internet, wireless telephony, and mobile electronics. Less effort has been dedicated to the impact of these technologies in industries such as aviation and aerospace where the primary motivation for existing regulations stems from public safety, both of air passengers and of those on the ground.

Current federal air regulations (FARs) did not anticipate operation of controlled unmanned aircraft in civil airspace. There is no specific part or definition under applicable law related to unmanned aircraft. The absence of absolute legal guidance with respect to the jurisdiction of UAV regulation, the definition of UAV, and the integration of UAVs in the national airspace prevents the optimum use of UAVs for the public benefit. Yet, given the risks of a ground impact or mid-air collision with other aircraft, the need for regulatory certainty respecting UAVs is an imminent issue deserving the attention of regulators, manufacturers, and operators alike (Ravich 2009).

To better articulate issues related to UAS technologies, the FAA Office of Aviation Research and Development funded a number of technology surveys and regulatory gap analyses, including the following UAS aspects: propulsion, C3, SAA, and ERFT. Before presenting the content of these studies, the following section shall address the higher level concepts of what a “technology study” or a “gap analysis” is, in the context of the introduction of disruptive technology in a regulatory context (such as the introduction of UAS into the NAS).

94.2.1 Technology Surveys

A technology survey for any disruptive or near-disruptive suite of technologies is a summary of current and near-future technologies having the disruptive impact.

One approach to articulate the elements of a set of technologies, such as those that could serve as propulsion systems for a remotely piloted aircraft, would begin with a catalog of the different aircraft and their propulsion systems. Listings of UAS in service and development are published regularly (AIAA 2011). While such tables are rich with aircraft dimensions, weight, operating regimes, flight durations, etc., they consistently fail to include specific details for the technologies in use. Due to proprietary interests of the developers and manufacturers, details ranging from the displacement volumes of internal combustion engines to the name of a vendor that developed a particular ballistic recovery system are rarely published; developers and manufacturers are understandably reticent to supply such information outside of a nondisclosure agreement.

The lack of such details turns out to not prevent an effective technology survey. From a regulatory point of view, what matters in such a survey is not the collection of details across some number of disruptive technology exemplars, but the underlying technologies those details represent. It is more important to articulate the families

of technologies in use than to exhaustively enumerate each unique case. The technology survey must comprise a complete articulation of the classes which the particular technology might take, but only that list of classes. Specification of that list of classes becomes part of the technology survey itself, based on an initial review of the forms of the technology in use in the disruptive technology.

The results of the technology survey are usually presented in tabular form, with that data kept as a spreadsheet or database while the survey is performed. The table is populated with rows, each of which collects the attributes of the representative elements of each class in the technology in the columns of the table. While a single representative element is conceptually sufficient to articulate the collection of technologies, the table usually contains multiple representatives for each technology. A complete articulation of every last instance and representative element of the technologies is not required for a successful survey.

A technology survey articulates current *and near-future* technologies through a collective set of representative examples. Given the snail's pace of changes to aviation regulations, the technology survey needs to capture technologies on the horizon that could have an impact within the next 5 or 10 years, even if they are not currently in production use.

A second, and possibly more important, artifact of the technology survey is a conceptual framing of the technologies to be used in future works, such as regulatory gap analyses. The conceptual framework can range from formal modeling of the underlying technologies to an abstract capturing of those technologies' salient aspects and operations. The model/framework creates the context for conduct of the regulatory gap analysis.

94.2.2 FAA Documents

In addition to the regulations, there are a variety of FAA documents that must be produced and/or revised to accommodate UAS-NAS integration. This section defines some common documents related to airworthiness, certification, and FAA policy. A *technical standard order (TSO)* defines the minimum operations performance standards and minimum aviation system performance standards for an aircraft system or subsystem. A component authorized for production under a TSO is identified as compliant with that TSO. The FAA can utilize *advisory circulars (ACs)* to share information with the aviation community. ACs have varied audiences including engineers, pilots, operators, etc. ACs can be used as a means of presenting critical design requirements so that aircraft meet sufficient airworthiness standards. It should be noted that these documents are not law. Lastly, *guidance material* is published by the FAA to the community. One example is the FAA UAS Interim Guidance Document 08-01 (Federal Aviation Administration 2008). Another example is the Aeronautical Information Manual (AIM), which defines pilot procedures in the NAS (Federal Aviation Administration 2012a). Guidance materials are not regulatory or as official as an advisory circular but are used to convey FAA policy, procedures, etc. to the aviation community.

94.2.3 Regulatory Gap Analyses

The FAA is chartered through the same piece of statute law chartering the entire Department of Transportation, Title 49 of the United States Code (USC). The FAA authorized in 49 USC §44701 to regulate aircraft and their usage within the framework of Federal administrative law. Regulations issued by the FAA comprise Chap. I (Parts 1–199) of Title 14 of the Code of Federal Regulations (CFR). These are divided into several subchapters relating to definitions, rule making, aircraft, airmen, airspace, air traffic and general operating rules, air carriers and operational, schools, airports, navigation facilities, and the FAA itself. It should be noted that 14 CFR Parts 1–199 are still known to many in the aviation world as “the FARs,” where “FAR” stands for “Federal Aviation Regulation;” in Federal speak, “FAR” has since been superseded to mean “Federal Acquisitions Rules.”

The meaning of a regulatory gap varies according to which chapters of Title 14 CFR one is considering. For example, aircraft are only allowed to fly in the NAS when the aircraft has been issued a type certificate (TC) for its design (or a Supplemental Type Certificate (STC)). The issuance of such a TC or STC follows a prescribed set of activities in which the aircraft designer articulates which elements of Title 14 CFR apply to the design at hand and then demonstrates through testing, analysis, and formalized processes that the design complies with the specifications and regulations of the articulated elements.

When an aircraft design employs a novel technology, it is unlikely that there is an existing regulation applicable to the technology. The designer can qualify the use of the technology toward certification through either special conditions (SC) or equivalent level of safety (ELOS) findings. A third option, exemption, exists, but applies more often to exemptions from specified procedures than from specified requirements. In either case, the aircraft designer articulates in an issue paper the features of the new technology and how its inclusion in the aircraft design satisfies the thrust, in general SC or in particular ELOS, of the regulations; FAA engineers respond to the issue paper, and the officers responsible for certification take the designers’ and the FAA engineers’ analysis into account in deciding whether to issue a TC.

For aircraft, then, a regulatory gap exists when certification of an airframe employing a technology would require either a SC or ELOS finding to use that technology. The result of the gap analysis is a collection of annotated regulations. The annotations consist of declarations as to whether the regulation:

- Applies (as is)
- Applies with interpretation
- Applies with revision
- Does not apply

From previous literature, it was found that several approaches to presenting regulatory gaps exist (Kirk et al. 2007; Frater et al. 2006). Kirk et al. (2007) provide an examination of Federal aviation laws, regulations, and guidance materials for applicability to UAS in general. Frater et al. (2006) present a different style of regulatory gap, focused upon nanotechnology, in which the results are presented

in a tabular form where related aspects of existing legislation and regulations are summarized, the gap or potential gap due to novel technology presented, and further comments or annotations are appended (see Annex 5 of Frater et al. 2006). ERAU's approach is much closer to the former following a process by which the collection of rule applicability results was looked at from both global (all rules together) and local (one rule at a time) to produce text describing the regulatory gaps.

94.3 Case Study #1: Propulsion Technologies for UAS

94.3.1 Technology Survey

The goal of the UAS propulsion technology survey, Griffis et al. (2007), was to examine existing and novel propulsion systems for UAS such as reciprocating piston engines (RP), wankel rotary engines (RO), propeller drive systems (PR), gas turbine propulsion systems (GT), rocket-powered means of propulsion (RK), electric motor-based propulsion system (EM), battery-based propulsion system (BB), fuel cell-powered propulsion system (FC), solar/photovoltaic-powered systems (PH), and ultracapacitor-based energy storage (UC).

A conceptual framework was derived as shown in Fig. 94.1. This framework guided the study of each technology area to maintain internal consistency. The elements of this framework are defined as follows:

Energy source (ES). Propulsion requires expenditure of energy, and the ES is the origin of that energy. ES is intended to be a generic label for things like the following examples: gasoline, diesel fuel, lithium hydride, liquid hydrogen, solar energy, etc.

Energy transformer (ET). An ET converts the potential energy within the ES into a means for producing work, heat, or electrical current.

Power plant (PP). A PP is any aspect that harnesses the product of the ET into motion. For example, a motor that spins a shaft as a result of a supplied electric current classifies as a PP in this context.

Propulsion effector (PE). A PE is the interface between the motion generated and the impulse exerted to move the vehicle; it is what will give the effect of propulsion.

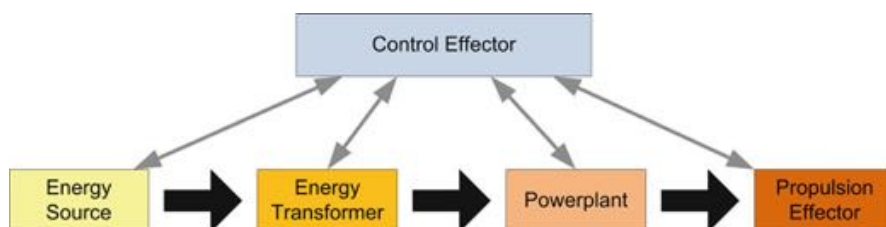


Fig. 94.1 Technology survey framework for UAS propulsion systems

Control effector (CE). A CE is whatever is in place to give the effect of control to the propulsion means in a way that serves the purpose of controlling propulsion generation for the vehicle. CE is intended to be a generic label for things that perform control on propulsion such as a full authority digital engine control (FADEC), throttle control, fuel mixture control, and current regulator.

The conceptual framework is representative of the state of propulsion systems. For example, consider the following utilization of the framework. A UAS employing an avgas-fueled RP would have petroleum distillates for its ES, combustion as its ET, the piston/crankshaft as its PP, a PR as its PE, and control of the flow of fuel, air, and electricity as its CE. It should be noted that in some cases single physical units can assume the role of two or more conceptual elements. For example, the thermodynamics of a GT can be considered an ET, PP, and PE simultaneously.

Reciprocating Piston Engines Reciprocating piston systems vary in size, geometry, and configuration. Often an engine is classified by how many cylinders it contains, how much total volume is displaced within its cylinders, or the configuration of those cylinders. They all contain some variation of the same basic parts, and there is a common principle behind how work is generated from stored energy of the fuel consumed as presented in Table 94.1. Representative examples of RP-based UAS are presented in Table 94.2.

Table 94.1 Conceptual decomposition of a reciprocating piston engine

| Conceptual unit | Description |
|----------------------------|--|
| <i>Energy source</i> | Petroleum distillates |
| <i>Energy transformer</i> | Heat production and expanding volumes resulting from contained combustion of petroleum distillates |
| <i>Power plant</i> | Piston motion resulting from expanding volumes, which in turn rotate the crankshaft |
| <i>Propulsion effector</i> | Propeller or fan unit driven directly or indirectly (geared) by the crankshaft |
| <i>Control effector</i> | Throttle, regulation of fuel flow |

Table 94.2 Representative cases of UAS using reciprocating piston engines (Office of the Secretary of Defense 2005; Parsch 2006; RCV Engines, Ltd 2006)

| Aircraft | Manufacturer | Subclass |
|----------------|------------------|---|
| MQ-1B Predator | General Atomics | 4-stroke |
| RQ-2B Pioneer | Pioneer UAV | 2-cylinder, 2-stroke |
| RQ-5A Hunter | Northrop Grumman | 4-stroke, heavy fuel |
| ScanEagle A | Insitu | 1-cylinder, 2-stroke (heavy fuel variant available) |
| T-Hawk | Honeywell | Heavy fuel engine |

Wankel Rotary Engine Wankel rotary engines use the combustion of petroleum-based fuel, and the desired output is the rotation of a power shaft that drives the rest of the system. They differ from conventional reciprocating engines in that their volume displacement and associated internal motion are rotational, as opposed to back and forth. An internal triangular core, shaped as a “Reuleaux” triangle, divides a chamber with an epitrochoid-shaped stator into three expansion areas (AREN 2006). A conceptual breakdown of the this propulsion technology is shown in Table 94.3. Examples of RO-based UAS are presented in Table 94.4.

Propeller-Based Systems A large body of knowledge exists for propellers, which is outside the scope of this document. However, as they are an important instance of commonly utilized propulsion effector, they deserve mention. Most reciprocating piston and rotary engines use propellers as their propulsion effectors, and many smaller UAS use electric motors to drive a propeller.

Gas Turbine Engines A gas turbine engine is an internal combustion engine operating on a highly dynamic process, processing air and fuel to yield high-velocity thrust. A gas turbine engine comes in various forms including turbine engine, turbofan engine, and turboprop engine. A conceptual decomposition representative of all three gas turbine approaches is presented in Table 94.5, and examples of its use in UAS are presented in Table 94.6.

Rocket Propulsion A rocket is propelled by a chemical reaction that generates extreme pressure gradients and high-velocity particles that exit a nozzle. The resulting momentum exchange provides impulse over some duration, accelerating the rocket’s mass (Brian 2007). Propulsion derived exclusively from rocket power tends to be used in applications where the asset is not expected to return home.

Table 94.3 Conceptual decomposition of a Wankel rotary engine

| Conceptual unit | Description |
|----------------------------|--|
| <i>Energy source</i> | Petroleum distillates internal energy relative to oxidation products |
| <i>Energy transformer</i> | Heat production and expanding volumes resulting from contained combustion of petroleum distillates |
| <i>Power plant</i> | Reuleaux triangular rotor motion within an epitrochoid stator turning the eccentric shaft |
| <i>Propulsion effector</i> | Propeller or fan unit driven directly or indirectly (geared) by eccentric shaft |
| <i>Control effector</i> | FADEC, carburetor, fuel/air flow control |

Table 94.4 Representative cases of UAS using Wankel rotary engines (Office of the Secretary of Defense 2005)

| Aircraft | Manufacturer | Sub class |
|------------|--------------|--------------|
| Shadow 200 | AAI | Single rotor |
| Cypher | Sikorsky | Single rotor |

Table 94.5 Conceptual decomposition of a gas turbine system

| Conceptual unit | Description |
|----------------------------|---|
| <i>Energy source</i> | Petroleum distillates |
| <i>Energy transformer</i> | Heat production and extremely high pressures (relative to operating environment) resulting from contained combustion of petroleum distillates |
| <i>Power plant</i> | Dynamic adiabatic/isentropic one of those processes of high-pressure gas converting to high-velocity gas from nozzle |
| <i>Propulsion effector</i> | High-velocity gas exiting the rear aperture |
| <i>Control effector</i> | Fuel flow, propeller pitch |

Table 94.6 Representative cases of UAS using gas turbine systems (Army-Technology.com 2007; Rolls-Royce, PLC 2006; Office of the Secretary of Defense 2005)

| Aircraft | Manufacturer | Subclass |
|-------------------|------------------|-----------|
| CL-289 | Bombardier | Turbojet |
| RQ-4A Global Hawk | Northrop Grumman | Turbofan |
| Predator B | General Atomics | Turboprop |

Table 94.7 Conceptual decomposition of rocket-based propulsion

| Conceptual unit | Description |
|----------------------------|---|
| <i>Energy source</i> | Self-contained chemical reactants (solid or liquid) |
| <i>Energy transformer</i> | Exothermic high-pressure chemical reaction in rapid release of kinetic energy |
| <i>Power plant</i> | Expulsion of reaction products through nozzle creating high-velocity exhaust |
| <i>Propulsion effector</i> | Thrust from momentum transfer of high-velocity exhaust |
| <i>Control effector</i> | Nozzle direction, reaction rate control |

Table 94.8 Representative cases of UAS using rocket propulsion (Office of the Secretary of Defense 2005)

| Aircraft | Manufacturer | Subclass |
|-------------------|-----------------|-------------------------|
| RQ-2B Pioneer | Pioneer UAV | Rocket-assisted takeoff |
| Cormorant Project | Lockheed Martin | Rocket-based takeoff |

However, takeoff assist can utilize rocket-based propulsion (Office of the Secretary of Defense 2005). Table 94.7 shows the conceptual decomposition of rocket propulsion, and Table 94.8 presents some examples of its use for UAS.

Electric Motors For electrically based propulsion systems, electric motors are used as the power plant because they can easily couple with propellers as the propulsion effector; all that is needed is a continuous source of electricity. The rotational speed of a DC motor is proportional to the voltage applied to it, and the torque is

Table 94.9 Conceptual decomposition of electric motor-based propulsion

| Conceptual unit | Description |
|----------------------------|--|
| <i>Energy source</i> | Unspecified |
| <i>Energy transformer</i> | That which yields electrical power |
| <i>Power plant</i> | An electric motor |
| <i>Propulsion effector</i> | Unspecified; usually a propeller or fan unit that is functioning as a result of the rotating motion created by the motor. Alternatively, for example, it could drive a wing-flapping mechanism |
| <i>Control effector</i> | Feedback control loops, current/voltage control |

Table 94.10 Conceptual decomposition of propulsion using stored battery power

| Conceptual unit | Description |
|----------------------------|--|
| <i>Energy source</i> | Electrochemical energy gradient between internal cathode/anode materials |
| <i>Energy transformer</i> | Chemical reaction yielding electron transport, generating an electromotive force |
| <i>Power plant</i> | Electrically driven motor |
| <i>Propulsion effector</i> | Propeller or fan unit driven directly or indirectly (geared) by motor shaft |
| <i>Control effector</i> | Voltage/current regulators, analog/digital feedback control loops |

Table 94.11 Representative cases of UAS using batteries for energy storage (AIAA 2005; Office of the Secretary of Defense 2005; Defense Update International Online Defense Magazine 2006)

| Aircraft | Manufacturer | Subclass |
|-------------|-----------------|-------------|
| Aladin | EMT | Unknown |
| Wasp | AeroVironment | Proprietary |
| Desert Hawk | Lockheed Martin | Unknown |
| BATCAM | ARA | Unknown |

proportional to the current (Hall et al. 2003). A conceptual breakdown of propulsion using electric motors is provided below in Table 94.9.

Batteries For UAS applications, rechargeable batteries are preferred and therefore the focus of discussion. Lithium batteries tend to be lighter and possess higher energy density (Reid et al. 2004). Table 94.10 presents the conceptual decomposition of UAS propulsion using electric motors driven by batteries, and Table 94.11 provides representative examples of UAS using batteries. Unfortunately, manufacturers do not frequently offer the details on what particular chemistry of battery is used.

Fuel Cells Instead of consuming oxygen in the direct combustion of fuel, a fuel cell consumes oxygen (or some other environmentally provided reactant) to

Table 94.12 Conceptual decomposition of a fuel cell based propulsion system

| Conceptual unit | Description |
|----------------------------|--|
| <i>Energy source</i> | Molecular hydrogen internal energy relative to water |
| <i>Energy transformer</i> | PEMFC creating a power source through ionization and electrochemical oxidation of molecular hydrogen |
| <i>Power plant</i> | Electrically driven motor |
| <i>Propulsion effector</i> | Propeller or fan unit driven directly or indirectly (geared) by motor shaft |
| <i>Control effector</i> | Hydrogen flow regulators, voltage/current boost regulators, analog/digital feedback control loops |

Table 94.13 Representative cases of UAS using fuel cells (Protonex Technology Corporation 2006; Office of the Secretary of Defense 2005)

| Aircraft | Manufacturer | Sub class |
|------------|---------------|--------------------------|
| SpiderLion | NRL/Protonex | Proton exchange membrane |
| Hornet | AeroVironment | Proton exchange membrane |

Table 94.14 Conceptual decomposition of a photovoltaic-based propulsion system

| Conceptual Unit | Description |
|----------------------------|--|
| <i>Energy source</i> | Solar radiation |
| <i>Energy transformer</i> | Superpositioned power-generating photovoltaic cells stimulated by the photoelectric effect |
| <i>Power plant</i> | Electrically driven motor |
| <i>Propulsion effector</i> | Propeller or fan unit driven directly or indirectly (geared) by motor shaft |
| <i>Control effector</i> | Voltage/current regulators, analog/digital feedback control loops |

generate electrical power via an electrochemical process. There are a wide variety of fuel cells, including proton exchange membrane fuel cells, phosphoric acid fuel cells, molten carbon fuel cells, solid oxide fuel cells, methanol fuel cells, and alkaline fuel cells (Theiss and Thomas 2000; National Fuel Cell Council 2006). Table 94.12 presents the conceptual decomposition of fuel cell-based propulsion, and Table 94.13 presents two representative cases of fuel cells used in unmanned aircraft.

Photovoltaics Photovoltaic technology obtains usable energy from sunlight. Being effectively cost free and inexhaustible, solar power is attractive for long-endurance systems. Unfortunately, due to weather or daily sunlight availability, UAS operations can be limited unless there is some means of energy storage. A conceptual framework for photovoltaic-based propulsion is shown in Table 94.14 with examples of its usage in Table 94.15.

Table 94.15 Representative cases of UAS using photovoltaics (AC Propulsion 2005; AIAA 2005)

| Aircraft | Manufacturer | Subclass |
|----------|---------------|----------------------|
| SoLong | AC Propulsion | Solar cells on wings |
| Helios | AeroVironment | Solar cells on wings |

Table 94.16 Conceptual decomposition of an ultracapacitor-powered propulsion system

| Conceptual unit | Description |
|----------------------------|---|
| <i>Energy source</i> | Electrostatic potential within capacitor plates, charged by external source |
| <i>Energy transformer</i> | Application of external load, liberating potential energy in the form of electric current |
| <i>Power plant</i> | Electrically driven DC motor |
| <i>Propulsion effector</i> | Propeller or fan unit driven directly or indirectly (geared) by motor shaft |
| <i>Control effector</i> | Voltage/current regulators, analog/digital feedback control loops |

Ultracapacitors An ultracapacitor is a specialized modern capacitor that has an unusually high energy density when compared to common capacitors (Electricity Storage Association 2007). Ultracapacitors are a potential stored energy source for a UAS propulsion system using electric motors. Ultracapacitors can retain much greater energies than standard capacitors for a comparable geometry. While there are no existing applications of ultracapacitors in existing UAS, their use in automotive applications provides a natural transition point into aerospace applications. A conceptual breakdown of propulsion using this technology is shown in Table 94.16.

94.3.2 Regulatory Gap Analysis

Following the technology survey, follow-on research was funded to perform a regulatory gap analysis. The analysis examined Title 14 CFR Part 27 Subpart E, Airworthiness Standards: Normal, Utility, Acrobatic, and Commuter Category Airplanes Power Plans; Title 14 CFR Part 33, Airworthiness Standards: Aircraft Engines; and Title 14 CFR Part 35, Airworthiness Standards: Propellers.

94.3.2.1 Regulatory Gap Analysis Process and Tools

The regulatory gap analysis was performed in two parts. The first part was a local assessment of the regulations to determine where regulatory gaps exist. The second part was a global assessment. The global assessment yields information on the state of the regulations based upon the results of the local assessment.

The local assessment was performed using a spreadsheet to capture a section-by-section analysis of the regulations.

Each row represents a specific section of the regulation part being examined. The section is analyzed based upon its degree of applicability for each conceptual technology and conceptual decomposition framework element. The dimensions of applicability were identified as applies (APP), applies with interpretation (AWI), applies with revision (AWR), or does not apply (DNA). For this study, these terms are defined as follows:

APP The regulatory guideline, as it stands, makes sense for the corresponding technology identified in the spreadsheet.

AWI Understanding the intent of each regulatory guideline, it can be interpreted to cover other areas or technologies that are not explicitly mentioned or addressed, for example, regulations that cover RP technology and also AWI to RO technology.

AWR Employed prudently. Suggests that the regulatory guideline is fine as it stands except with a minor amendment.

DNA The regulation does not apply to any of the propulsion technologies or conceptual aspects of a UAS propulsion system.

94.3.2.2 Results

Upon completion of the local analysis using the spreadsheets, global analysis was performed to derive the final results and recommendations of the study. First, the “fundamental gaps” are identified, which include any fundamental technological differences between the propulsion technologies for UAS versus manned aircraft. Next, the “open set” gaps are identified where regulatory gaps exist because of safety concerns for technologies that fall outside of the existing regulatory framework.

The Fundamental Gap The fundamental gap between the existing regulations and UAS propulsion technologies results from a greater diversity in the types of ET and PP that can be utilized for a UAS. Existing regulations focus upon one of two types of systems, GT and RP. As a result, some regulations simply do not apply to alternative propulsion systems, and in other cases new regulations are required to address the safety concerns of the new technology. A fundamental gap exists in regulation of systems that depend exclusively on EM for propulsion and that there exists no specific regulation addressing the kinds of power supplies that would be driving these propulsion systems. UAS (excluding optionally piloted vehicles) do not need to be concerned with the power requirements to propel the weight of both the pilot and the onboard support and control interfaces, thereby further reducing the electrical carrying requirements of an electrical power source. Technologies such as FC, modernized batteries, UC, and PH have the ability to supply sufficient electricity to provide sufficient endurance for a UAS and must now be considered by regulators.

The regulatory guidelines for thermodynamically driven engines address a set of associated high-level concerns related to the lubrication of moving parts, heat transfer, fuel delivery, air supply, fuel storage, etc. The GT and RP approaches to propulsion both carry flammable liquid petroleum distillates that burn hot with oxygen and have reactants that need to be expelled. These fundamental issues

(among many others) are addressed in the regulatory guidelines in a manner that assures that any implementation of these approaches will be airworthy and have an associated reliability for a given specified period of time. For an electric engine, many of these factors do not apply with the same literal interpretation. Some issues still need to be addressed such as ensuring that the motors remain sufficiently cool and the moving parts remain lubricated. However, a new set of concerns are introduced with the concept of an exclusively EM. Many of the restrictions and regulations in place for guaranteeing the safety of GT and RP in some cases do not make sense and are insufficient when dealing with an EM.

The Open Set Gap From the fundamental gap and the technology survey, it is evident that there exist technologies that exist outside of the regulations. Incremental adjustments to the regulations can be made to accommodate these changes. As new technologies emerge, this could result in regulators continually attempting to patch the regulations rather than create a new mechanism for addressing these new and emergent technologies. In order to close the coverage of the open set gap, the study concluded that the concepts of catch-all regulations should be embraced and extended. This can include extending concepts of abstract regulation, in similar approaches to the conceptual framework. Ideas like those in Parts 23 and 25 Sects. 1301 and 1309 can be complemented with regulation of generic propulsion systems, regulating only the conceptual components, their abstract interfaces, and the conceptualized interactions that each component would have between its interface and another conceptual component interface.

94.4 Case Study #2: Sense-and-Avoid Technologies for UAS

94.4.1 Technology Survey

The sense-and-avoid (SAA) technology survey is guided by a representative framework for sense-and-avoid technologies for UAS. Figure 94.2 graphically depicts this framework. Three major categories of technologies relevant to the topic of SAA for UAS are sense, avoid, and be seen. Each will be discussed in this section.

94.4.1.1 Sense

The first category of SAA technology is *sense*. These technologies allow the UAS to detect other traffic and local terrain features. It is subdivided into airborne sensing and ground-based sensing.

Airborne: Cooperative Cooperative technologies require other NAS users to be equipped with specialized communication equipment, such as a working transponder, such that they can be seen by other aircraft.

Traffic alert and collision avoidance system (TCAS) is a technology used in manned aviation to improve pilot situation awareness to mitigate midair collisions. TCAS I and TCAS II provide traffic advisories to the pilot in the event that

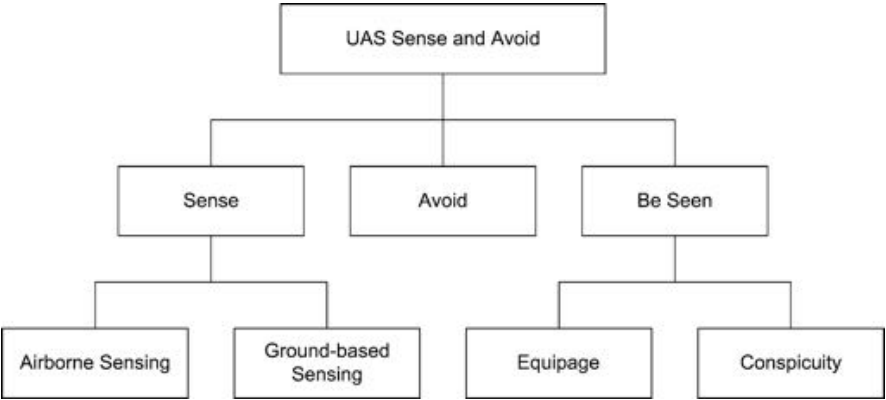


Fig. 94.2 Technology survey framework for UAS sense-and-avoid technologies

there is a risk of collision given the current flight path and oncoming traffic. TCAS II also provides collision avoidance directives to the pilot (Federal Aviation Administration 2012b). A recent FAA study examined TCAS for UAS (Federal Aviation Administration 2011). It concluded that TCAS could be a viable tool to aid situational awareness to the UAS operator/pilot but stated that the technology was never approved as a sole means of replacing the pilot’s role to see and confirm the presence of other air traffic.

Automatic dependent surveillance-broadcast (ADS-B) can also provide an airborne sensing capability to improve situational awareness of local air traffic. Equipped aircraft broadcast their current position and some additional state data via data link. The broadcast message can be received by suitably equipped aircraft. For manned aviation, the aircraft positions received can be displayed in real time to the pilot to enhance local situational awareness. Air traffic controllers can also receive this information via ground-based transceivers (Federal Aviation Administration 2007).

There are two types of ADS-B technology currently in use. Universal Access Transceiver (UAT) technology at 978MHz is most commonly used by general aviation aircraft. The second type utilizes a 1,090MHz Extended Squitter (ES) Mode-S transponder and is most commonly applied to transport category aircraft (Federal Aviation Administration 2007). The MITRE Corporation has produced prototype UAT units for UAS operations including one that is transmit only and another that is capable of both sending and receiving ADS-B messages (Strain et al. 2007).

Airborne: Noncooperative, Active Active non cooperative systems scan ahead of the aircraft to identify local traffic and/or collision threats by emitting and then receiving some form of sound or energy. The technologies vary in size and configuration for airborne sensing of local air traffic and/or terrain features.

Radar can be equipped onboard a UAS for airborne sensing of other aircraft. Smaller non-cooperative radar systems have been employed in robotics applications in the past such as millimeter wave (MMW) radar. *Laser* systems such as LIDAR or laser range finders emit laser light, which is reflected off of the surface of a target. Depending upon the scanning technique, sampling resolution, etc., it is possible to analyze the shape of the target. The system can also track a target over successive scans allowing for a determination of range, bearing, speed, and trajectory. The sensors can be robust enough to support operation in conditions of low human visibility such as fog or smoke. *Sonar* systems utilize the emission of acoustic pulses. The time of flight from transmission to the reception of the reflection is used to determine the approximate range to the target. Due to the limited resolution and range, sonar at this time is not likely a viable option for airborne active sensing.

Airborne: Noncooperative, Passive Passive noncooperative systems do not emit sound or energy in order to detect a target. These sensors often cannot yield as robust of a data set as active sensors including safety critical data such as bearing, range, velocity, or trajectory of the target. *Electro-optical (EO)* cameras convert visible spectrum light into electronic signals and analyze the changes in data. EO cameras can be used with image processing algorithms to detect and analyze visible targets. Video or still images can also be rebroadcast to ground-based operators. Motion detection algorithms (Hottman et al. 2007) can be used to detect potential targets and track features of its motion by using multiple cameras. *Infrared* cameras are similar to EO but capture light frequencies within some subset of the infrared spectrum. These cameras and their processors can produce images for human operators as either black-hot objects or white-hot objects (i.e., with black-hot, the darker a pixel is within the image, the hotter it was perceived by the camera, and vice versa for white-hot). These images can also be examined by computer vision algorithms to do automatic target detection and tracking. One advantage of infrared is that it can operate in conditions where the visual spectrum is occluded such as fog and smoke (Access 5 2004). An *acoustic* system utilizes an array of microphones and processors to analyze the sounds within the airspace around the aircraft and can be capable of detecting and tracking local air traffic (Hottman et al. 2007). The viability of this approach is debated. For instance, a NASA Access 5 study indicates concern about the high signal-to-noise ratio (Access 5 2004). However, SARA Inc. has developed a prototype system utilizing a windscreen to dampen the noise from wind and aircraft vibration (SARA, Inc. 2008).

Ground-based: Cooperative *Traffic information system-broadcast (TIS-B)* is similar to ADS-B. Suitably equipped aircraft or ground stations can receive information regarding local air traffic. TIS-B uses ground-based radar to detect local air traffic. This information is then transmitted via uplink to aircraft equipped with an ADS-B receiver. The ground-based radars can detect non-ADS-B-equipped aircraft so long as they have a Mode-S or Mode-C transponder (Access 5 2004).

Air traffic control (ATC) radar-based separation is another approach that can be taken to aid the sensing of aircraft. Primary radar can be used to detect cooperative and noncooperative aircraft within a limited range. Secondary radar could be used to detect transponder equipped, cooperative aircraft.

Ground-based: Noncooperative *Ground-based radar* separation can be supported for noncooperative aircraft as well as cooperative aircraft. To detect a noncooperative aircraft, primary surveillance radar would be required. Such radar provides limited range and can have difficulty discriminating between local aircraft and other airborne phenomena such as birds.

Spotters can provide noncooperative ground-based sensing of local air traffic so long as the UAS is operating within their visual line of sight. The observer must be able to identify the aircraft and local air traffic. It is the spotter's duty to notify the pilot in command (PIC) if any perceived right-of-way issues exist with local air traffic. A pilot in command (PIC) is the UAS pilot/operator that has legal authority over the flight. Their responsibility from a regulatory point of view is analogous to the pilot in command of a manned flight. It is also common to have airborne spotters that observe the UAS and its local air traffic from a chase aircraft.

94.4.1.2 Avoid

The second category of SAA technology is *avoid*. Collision avoidance can be addressed through one of two primary mechanisms. First, the system can be diverted by a remote operator. Second, the aircraft could choose to autonomously avoid the collision using its autopilot. There can also be a blending of these two approaches. Aircraft autonomy is addressed further in Sect. 94.5.

94.4.1.3 Be Seen

The third category of SAA technology is *be seen*. This represents the ability for the aircraft to be seen by other aircraft both cooperative and noncooperative and ATC. This is divided into UAS equipage for being seen and UAS conspicuity.

Aircraft Equipage *ADS-B* can be equipped on UAS. The MITRE Corporation has developed the UAT Beacon Radio transmit-only and transceiver ADS-B (UBR-TX and UBR-TVR) prototypes for general aviation and UAS operations (Strain et al. 2007). This equipage would allow the UAS to be visible to ATC and suitably equipped local air traffic. *Transponder*-equipped UAS will also be visible to others whenever operating in range of a primary or secondary radar site. Lastly, *TCAS*-equipped UAS could be detected by other TCAS-equipped aircraft so that they could receive collision warnings and/or deviation instructions should a collision risk exist.

Aircraft Conspicuity An RTCA SC-203 report, DO-304 (RTCA SC-2003 2007), states the concept and need for conspicuity quote well:

Conspicuity: Many UA are small and made of materials that provide minimal radar cross sections. Aircraft that are difficult to see by human sight or by systems (e.g., radar or optical) can increase risks of collisions. These could be mitigated by paint schemes, lights or radar reflectors to enhance visibility, but these measures must be appropriate to the flight environment.

In addition to the techniques discussed in the definition above, procedural changes made by the UAS operator can also impact the ability of others to detect the UAS (Hottman et al. 2007).

94.4.2 Regulatory Gap Analysis

The regulatory gap analysis for SAA is divided into two areas of focus. First, FAA Title 14 CFR Part 91 is reviewed to identify regulations related to the PICs role to see and avoid other aircraft. Next, while not law, the AIM (Federal Aviation Administration 2012a) is examined because it provides guidance to pilots regarding their role to see and avoid other aircraft.

94.4.2.1 FAA Title 14 CFR Part 91

Title 14 CFR §91.111: Operating Near Other Aircraft This regulation covers restrictions forbidding the operator to control his/her aircraft in close proximity to other aircraft. The regulation states that operations near another aircraft cannot be made if a collision hazard exists. For manned aviation, this is an instruction to provide adequate self-separation to mitigate risk of collision. For an unmanned aircraft, this regulation points out the necessity of a separation mechanism. The sensing mechanism must have sufficient accuracy and resolution such that the operator and/or vehicle autonomy can determine both the location of the aircraft and the location of the other aircraft. The pilot in command and/or the autonomy must be capable of responding appropriately when a minimum distance of separation is violated. Qualitative and quantitative requirements for avionic components will be necessary to ensure that a sufficient level of safety is maintained.

Title 14 CFR §91.113: Right-of-Way Rules: Except Water Operations This regulation covers the necessary vigilance required by the system (operator and/or autonomy) to see and avoid other aircraft. It also identifies the right-of-way requirements concerning a variety of airborne platforms, as well as the right-of-way requirements among similar aircraft in various situations. The regulation also discusses right of way for landing aircraft. Similar to 91.111, there is a necessity of the sensing technology to determine when the UAS is operating at the same or a sufficiently close altitude to another aircraft and that the two aircraft are converging. There is currently no special rules for UAS. As regulators and engineers evaluate the safety case for passing operations, it is possible that such rules and priorities must be identified.

A potential gap exists with respect to UAS operating around aircraft performing landing operations. A landing aircraft is said to have the right of way. If two aircraft are landing, the lower aircraft has the right of way. This would require that the

aircraft be capable of not only sensing the other aircraft but also deriving its intent to land. Similarly, the regulation also gives right of way to aircraft in distress, which can likely be difficult to identify when the pilot is physically decoupled from the aircraft.

Title 14 CFR §91.115: Right-of-Way Rules: Water Operations This regulation covers right of way similar to 91.113 but includes water operation with interaction can include non-aircraft vessels. This includes head-on, crossing, and overtaking conditions with other vessels. This specifically must be addressed for UAS designed for water operations. Visual observation can be more difficult with water operations. Identifying surface vehicle capabilities and intent can also be more difficult for a UA and/or its operator.

94.4.2.2 Aeronautical Information Manual

Essential to the scope of this project, the AIM identifies the process of fulfilling the see-and-avoid requirements. It also provides a detailed synopsis of how ground operations should be performed. For ERAU's analysis of the AIM, excerpts were extracted that were deemed applicable to see and avoid processes of manned aircraft that would need to be replaced by SAA capabilities of UAS.

AIM §4-4-14: Visual Separation This section discusses visual separation as a mechanism for ensuring safe operation in the terminal area and en route. This includes visual observation provided by ATC in the tower within the terminal area. It also includes active scanning of the airspace around the aircraft to ensure separation. In most flight situations, scanning the sky is a necessary component of flight safety, and it is mentioned in AIM §8-1-6(c)(1) that spotting potential threats increases directly with the amount of time the operator spends scanning the skies. A SAA system would more than provide constant scanning. Unlike the human eye scanning at 10° per second limited by several blind spots in the aircraft, a SAA system may be less limited with regard to the distance, speed, and directions of sensing.

AIM §4-4-15: Use of Clearing Procedures This section discusses clearing procedures used to maintain visual awareness of the airspace around the aircraft. It defines expectations for pre-takeoff, climbs and descents, straight and level, traffic patterns at VOR sites, training operations, etc. It also distinguishes between low-wing and high-wing aircraft. UAS operations also require a clearing of the airspace to maintain assurance of safe separation with other aircraft. However, depending upon the see-and-avoid systems onboard, the need for actual clearing procedures may be reduced or eliminated given the scanning and/or field-of-view capabilities of the sensor(s).

AIM §5-5-8: See and Avoid This section simply restates that under permitting meteorological conditions the pilot has the responsibility to see and avoid other aircraft. The controller can provide local air traffic information as workload permits.

The controller can also issue safety alerts if other unsafe situations are observed. This relates directly to the sections of Part 91 discussed above.

AIM §8-1-6: Vision in Flight This section of the AIM discusses effective use of vision given various levels of illumination as well as techniques that can be employed to scan the sky. It also identifies some quantitative data that can be useful in determining human capabilities for aircraft detection. Some of these details can be relevant to UAS when dealing with ground observers. Other aspects may not be applicable as scanning techniques and the impact of illumination may have little or no impact on particular sensors.

AIM §8-1-8: Judgment Aspects of Collision Avoidance This section of the AIM provides guidance on handling unique collision avoidance situations. It describes determining relative altitude based upon the location of another aircraft with respect to the observer's determination of the horizon. This holds some relevance to visual sensors such as EO or IR cameras. However, some UAS sensors may not be capable of utilizing this technique or have better alternative results for altitude approximation. Visual observers on the ground would be unable to apply this technique.

The section also describes addressing multiple threats simultaneously. This calls on observation of the other aircraft as the pilot performs the avoidance maneuver to ensure that a secondary collision threat does not result. This can be challenging depending upon the SAA technology and the lack of situational awareness for the PIC. Some situations in which visual limitations exist are also discussed. For instance, poor windshield conditions are considered (which could be similar to a fouled sensor on a UAS), which would need to be replaced and/or repaired prior to flight. Similarly, smoke, haze, dust, etc. can reduce visual acuity for manned aircraft visual separation but may or may not have an impact for UAS operators depending upon the sensing techniques employed.

94.5 Case Study #3: Command, Control, and Communication Technologies for UAS

94.5.1 Technology Survey

UAS C3 for this study is defined as:

Command: Aspects related to ensuring the aircraft's progress toward completion of its mission including, but not exclusive to, flight automation, remote piloting, link loss (LL) procedures, situational awareness, flight ATC coordination, etc.

Control: Aspects related specifically to flight control including control surfaces, fly-by-wire(less) systems, and aspects involving flight control at the GCS including cockpit indicators and aircraft controls (yoke, pedals, throttle levers, joystick, touch screens, etc.)

Communication: Aspects of wired and wireless communication necessary to ensure safe operation of the aircraft and communication with ATC regardless of the UA's proximity to the GCS.

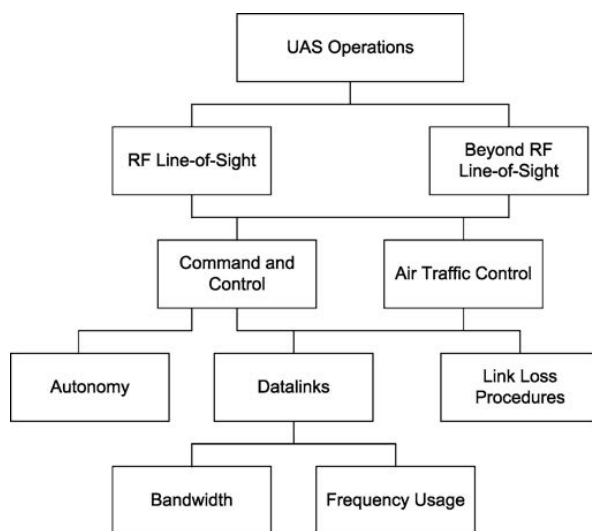
Use of the C3 acronym is not without issue. C3 as originally used by the U.S. Department of Defense (DOD) had “command” meaning ordering tactical elements to carry out some particular mission or goal, “control” as execution of that mission by means of verification of progress and correction toward the goal(s), and “communication” referring to communication between the command and control elements (United States Naval Academy 2008). The Radio Technical Commission for Aeronautics, Inc. (RTCA) Special Committee 203 (SC-203) has condensed this acronym to control and communication (C2), eliminating command. “Control” under their definition combines all aspects of the system necessary to direct the aircraft's execution of its mission; “communication” refers to both ATC communication and communication between the control station, the aircraft, and flight observers (air or ground based). Such a grouping is problematic, in that there are numerous regulations focusing upon control-related aspects, including those related to control surfaces, cockpit layout, pilot flight control requirements, and adequate redundancy.

A technology survey was performed on C3 technologies for current and near-future UAS. A web search yielded a summary of the UAS. From this list, representative aircraft with low, medium, and high endurance, operating within both RF line of site (LOS) and RF beyond line of site (BLOS), was selected. For each, a deeper search was performed to locate data sheets and other information. These data were compiled, and a hierarchical model was produced based on the key variation points among representative systems. Using this model and the data collected, the technology survey section was generated and organized. In this report, the terms LOS and BLOS will refer to line-of-sight capabilities of RF signal propagation. Any cases of visual line of site will be explicitly stated as such.

The C3 framework is shown in Fig. 94.3, organizing surveyed technologies into categories. UAS operations utilize LOS and/or BLOS communications. Under each, technical issues are divided into two categories, C2 and ATC. For C2, the survey explored technologies and issues necessary to safely support flight operations of the UAS from a remote pilot and/or control point of view. For ATC, technologies and issues on the interaction of the aircraft or PIC with ATC were explored. Various data links were examined including their frequencies and data rates. LL procedures are enumerated. For C2 only, the issue of autonomy, remote pilot versus autopilot, is examined.

It is important to note that most BLOS-capable UAS incorporate some LOS technologies. LOS operation may be divided between three classes of UA: low endurance, medium endurance, and high endurance. Low endurance operates almost entirely within LOS. Examples included Advance Ceramic Research's (ACR) Manta B (Advance Ceramic Research 2008), ACR's Silver Fox (Advance Ceramic Research 2008), Meridian (Hale et al. 2007), AeroVironment's (AV) Raven (AeroVironment 2008), and AV's Dragon Eye (AeroVironment 2008). Medium- and high-endurance classes operate in both LOS and BLOS conditions. Medium-endurance examples include Insitu's ScanEagle (Insitu 2008), Insitu's Georanger

Fig. 94.3 UAS C3 hierarchical system model



(Insitu 2008), and AAI Corp.'s Shadow (AAI Corp. 2008). Examples of high endurance include General Atomics' (GA) Predator (General Atomics Aeronautical Systems, Inc. 2008), GA's Mariner (General Atomics Aeronautical Systems, Inc. 2008), Northrup Grumman's (NG) Global Hawk (Northrop Grumman 2008), NG's BAMS (Northrop Grumman 2008), and AV's Global Observer (AeroVironment 2008). Table 94.17 lists some examples of LOS C3 technologies.

BLOS UAS cover primarily high-endurance UAS. Table 94.18 lists some examples of BLOS C3 technologies aboard UAS.

C2 Data Links LOS C2 data links use spectrum from VHF (30–300 MHz) to microwave C band (4–8 GHz) (Neale and Schultz 2007). The most common LOS data link employed for UAS is C band, using 3.7–4.2 GHz for downlink and 5.9–6.4 for uplink. C Band is strategically chosen its frequency less affected by extreme weather.

Some small UA like ScanEagle and Georanger, Meridian, Shadow, Dragon, and Raven use UHF (300 MHz–3 GHz) for LOS C2. It is not uncommon for these aircraft to utilize 72 MHz handheld remote control similar or identical to those used by hobbyists. Some experimental UAS use IEEE 802.11 for C2 link (Brown et al. 2006; Frew et al. 2008), allowing ad hoc networks between UAS. Their range is LOS and directional antennas may be required to ensure signal strength to maintain connectivity.

For BLOS, low Earth orbiting (LEO) and geostationary Earth orbiting (GEO) satellites represent two extremes for satellite communication (SATCOM). LEO satellites operate around 400 km (250 miles). GEO satellites operate around 35,800 km (22,200 miles). Because they are closer to the Earth's surface, LEO satellites can transmit equivalent bit-error-rate messages with lower power.

Table 94.17 Line-of-sight communication for a sample of surveyed unmanned aircraft

| Aircraft | Manufacturer | LOS communication | Characteristics |
|-------------|---|--|---|
| Predator | General Atomics Aeronautical Systems | C band | Wingspan 20.1 m Length 10.9 m Payload 385.5 kg Max altitude 15,240 m Max endurance: 30 h |
| Global Hawk | Northrop Grumman Integrated Systems | CDL (137 Mbps, 274 Mbps); UHF SATCOM | Wingspan 39.9 m Length 14.6 m Payload 1,360.7 kg Max altitude 18,288 m Max endurance 36 h |
| ScanEagle | Insitu Group | 900 MHz of spread spectrum frequency hopping; UHF command/telemetry | Wingspan 3.1 m Length 1.2 m Payload 6.0 kg Max altitude 4,998 m Max endurance 20 h |
| Meridian | University of Kansas | 72 MHz Futaba radio 2.4 GHz microband radio | Wingspan 8 m Length 5.1 m Payload 54.4 kg Max altitude 4,572 m Max endurance 9 h |
| Desert Hawk | Lockheed Martin | Military 15 km data link | Wingspan 1.4 m Length 0.9 m Payload 3.1 kg Max altitude 152 m Max endurance 1 h |
| Dragon Eye | AeroVironment | Military 10 km data link @ 9600 baud | Wingspan 1.2 m Length 0.9 m Payload 2.3 kg Max altitude 152 m Max endurance 1 h |
| Manta B | Advance Ceramic Research | Military band/ISM band radio modem; 24–32 km radio | Wingspan 2.7 m Length 1.9 m Payload 6.8 kg Max altitude 4,876 m Max endurance 6 h |

Since they are not stationary relative to the Earth's surface and narrower field of view, LEO satellite constellations require a larger number of satellites to achieve the same coverage as GEOs. In Peters and Farrell (2003), a constellation of 80 LEO satellites was compared with a six-satellite GEO constellation with equivalent coverage area using Ka band. The LEO constellation outperformed the GEO constellation with reduced latency, lower path losses, and reduced launch cost. A LEO satellite constellation has higher operational costs. Examples of widely used LEO constellations include Iridium (2008) and Globalstar (2008). For both cases,

Table 94.18 BLOS communication for a sample of surveyed unmanned aircraft

| Aircraft | Manufacturer | BLOS communication | Characteristics |
|-------------|---|---|-----------------------|
| Predator | General Atomics Aeronautical Systems | Ku band SATCOM | Wingspan 20.1 m |
| | | | Length 10.9 m |
| | | | Payload 385.5 kg |
| | | | Max altitude 15,240 m |
| | | | Max endurance 30 h |
| Global Hawk | Northrop Grumman Integrated Systems | Ku band SATCOM; Inmarsat | Wing Span 39.9 m |
| | | | Length 14.6 m |
| | | | Payload 1,360.7 kg |
| | | | Max altitude 18,288 m |
| | | | Max endurance 36 h |
| ScanEagle | Insitu Group | Iridium | Wingspan 3.1 m |
| | | | Length 1.2 m |
| | | | Payload 6.0 kg |
| | | | Max altitude 4,998 m |
| | | | Max endurance 20 h |
| Meridian | University of Kansas | Iridium A3LA-D modem 2.4 Kbits/s 1,616–1,626.5 MHz | Wingspan 8 m |
| | | | Length 5.1 m |
| | | | Payload 54.4 kg |
| | | | Max altitude 4,572 m |
| | | | Max endurance 9 h |
| Desert Hawk | Lockheed Martin | No BLOS operations disclosed | Wingspan 1.4 m |
| | | | Length 0.9 m |
| | | | Payload 3.1 kg |
| | | | Max altitude 152 m |
| | | | Max endurance 1 h |
| Dragon Eye | AeroVironment | No BLOS operations disclosed | Wingspan 1.2 m |
| | | | Length 0.9 m |
| | | | Payload 2.3 kg |
| | | | Max altitude 152 m |
| | | | Max endurance 1 h |
| Manta B | Advance Ceramic Research | No BLOS operations disclosed | Wingspan 2.7 m |
| | | | Length 1.9 m |
| | | | Payload 6.8 kg |
| | | | Max altitude 4,876 m |
| | | | Max endurance 6 h |

when the UA moves from one satellite's coverage area to another, service may be temporarily disrupted as communications are handed off.

BLOS C2 data links range from UHF (300 MHz–3 GHz) to Ku band (12–18 GHz) via SATCOM (Neale and Schultz 2007). Ku band is used by high-endurance UAS like Global Hawk, BAMS, and Predator and its derivatives. INMARSAT SATCOM data links, with a frequency range from 1,626.5 to 1,660.5 MHz for uplink and 1,525–1,559 MHz for downlink (INMARSAT 2008), are used by some high-endurance UAS including BAMS, Mariner, and Global

Hawk. L band iridium data links range from 390 MHz to 1.55 GHz (Iridium 2008) and are used by smaller, low- or medium-endurance research UAS such as Georanger and Meridian.

Certain military UAS use Common Data Link (CDL) or Tactical CDL. CDL is a jam-resistant spread spectrum digital link incorporating multiple microwave bands (Global Security 2008). CDL is mostly used for BLOS operations; it can be used for LOS operations to ensure continuously safe and seamless communication when deployed in hostile territory. Data sheets for larger military UAS (e.g., Predator-B, Global Hawk) show identical specifications to CDL without explicitly stating the technology is in use.

Flight Control Technologies and Operation Low-endurance UAS often uses LOS remote control (R/C) for at least part of the flight. For takeoff/landing, an R/C pilot controls the aircraft. Once airborne, the pilot can fly the aircraft manually or allow the autopilot to perform waypoint navigation along a flight path. Representative examples include Manta B, Meridian, Raven A and B, Dragon Eye, Silver Fox, and Shadow. High-endurance UAS often use autopilot for all LOS (and BLOS) flight operations. The flight plan is programmed into the autopilot. Once the mission begins, the aircraft will autonomously takeoff and follow the predefined path. The pilot remains out of the C2 loop but monitors the flight operations and, if need be, interrupts the autopilot and takes over piloting duties. Representative examples include Predator, Mariner, ScanEagle, Georanger, Global Hawk, and BAMS.

For BLOS operations, remote piloting becomes less feasible with SATCOM due to its latency. Most long-endurance BLOS UAS employ autopilots and a pilot on the loop (able to intervene if necessary). The flight plan is programmed in the autopilot via the GCS GUI.

C2 Link Loss Procedures In the original study performed by ERAU, lost link (LL) procedures were addressed as part of the C3 study. One year later, the ERFT study revisited this topic in greater detail. For the sake of conciseness and to avoid redundancy, this topic with respect to C2 will be further addressed within the ERFT case study later in this chapter.

ATC Communication, Coordination, and Lost Link For ATC LL, the objective is to reestablish voice communications between the GCS and the ATC authority. For LL with ATC for LOS operation, a land-based phone line is the only option currently used. Some UA are equipped with multiple VHF transceivers that could be used to establish a ground control to ATC voice communication link using the UA as an intermediary.

BLOS UAS PIC-ATC communications can utilize the UA as a communication relay. The PIC contacts the ATC facility local to the UA via VHF radio onboard the aircraft. For BLOS operations, reestablishing a connection after LL requires redundant voice communication systems onboard. For the Altair LL, the FAA and ATC were provided with detailed flight plans, making sure that the ATC knew

the aircraft's location. Additionally, the missions were planned meticulously with respect to ATC coordination, such that all potential ATC facilities are notified. The mode of notification was not explicitly disclosed (Ambrosia et al. 2007).

Using a UA as a voice relay with ATC has technical issues such as handoff. For manned aircraft, as it transitions from one ATC cell to another, the onboard pilot dials the VHF radio to the appropriate ATC channel as instructed through the handoff procedure. For several existing COAs and aircraft, the aircraft performs a rapid ascent to an altitude above controlled airspace (i.e., above 60,000 ft) and maintains this altitude for the duration of the flight. As a result, interaction along a flight path involving multiple ATC facilities is not common, and proper procedures to handle operations within controlled airspace have not been formally developed. For UAS to operate within ATC-controlled airspace in the NAS BLOS, further protocols must be established regarding the handling of the handoffs and setting of the new frequencies of the aircraft's ground-to-ATC relay. Another potential issue of using UAS as a relay is the spectrum availability to handle additional voice channels (25 kHz bandwidth) to support each UA (Heppe, personal communications, Insitu, Inc., 2008). A proposed alternative is to utilize ground-based telecommunications networks to connect the PIC at the GCS to the ATC facility under which the UA is operating.

94.5.2 Regulatory Gap Analysis

The C3 gap analysis examined 14 CFR Parts 21, 23, 25, 27, and 91 and the AIM. Each section was labeled *does not apply*, *applied as is*, *applies with interpretation*, or *applies with revision* with respect to the categories command, control, and communication. Fundamental gaps were identified as well as gaps associated with particular technologies and/or regulations. Related gaps were grouped as appropriate. Additional regulatory concerns not fitting into the above categories were also identified.

The regulatory gap analysis was performed iteratively. First, each section of the parts was classified using the four labels above for command, control, and communication. During the second pass, the reviewer annotated sections that were labeled as applicable with interpretation or revision. Third, the annotations were organized and merged to produce the regulatory gap analysis report.

94.5.2.1 Fundamental Gaps

Fundamental gaps were defined as issues with the existing regulatory framework resulting in widespread regulatory gaps from the fundamental difference between UAS C3 versus manned aviation C3 and their operating/flight procedures. The most fundamental gap is that UAS are comprised of two physically separate subsystems, the UA and the GCS at which the PIC is located, connected via data links. Secondly, ATC-PIC voice must now be reconsidered. Numerous gaps result from communication being decoupled for UAS. Under some instances, reinterpretation is required such as 14 CFR §§21.{3, 16, 19, 33, 35, 37, 45, 127, 305, 605, and 609}

and 23.1309. It was concluded that UAS should be certified as a whole, and any deviations from current TSOs as regulated in 14 CFR §21.609 should be acceptable only by demonstrating an ELOS for the entire system.

The second fundamental gap is the definition, roles, and responsibilities of the PIC. Because the PIC is no longer onboard the aircraft, regulations and procedures defined with this expectation must be reviewed and reinterpreted to address this change.

The decoupling of the pilot's flight control interfaces and the aircraft control system (autopilot, surfaces, etc.), and the required data link to support this wireless interaction, results in the next fundamental gap. Revisions would be required in 14 CFR §§23.{175, 177, 1329}, 25.{175, 177, 253, 331, 1303, 1329}, and 27.143 to ensure that the failure of certain data links will not result in a system failure. Regulations that previously described the redundancies needed between for flight controls and control surfaces may now be reinterpreted as applying to the data link.

94.5.2.2 Command

In industry and the military, UAS pilots have not necessarily been licensed pilots. 14 CFR §§21.37 and 21.31(b) call for the pilot to be licensed and medically certified in order to assume command of the aircraft. It must be decided whether UAS PICs require the same medical and training standards as licensed pilots. Training can be developed to certify licensed UAS operators, requiring at minimum pilot ground school. Flight instruction regulations are defined in 14 CFR §91.109. For a typical training flight, an instructor pilot sits in the aircraft's copilot position, having complete authority to take command of the aircraft at any time. UAS instructor pilots must similarly be able to override a UAS trainee, requiring GCS with a redundant set of flight controls offering the instructor pilot the same ability to immediately subsume the trainee pilot's commands.

Under 14 CFR §91.7, the PIC of an aircraft has the responsibility to perform the safety-critical preflight check of the aircraft prior to departure. This is suitable for UAS operations in which the takeoff and landing occur at the same site as the UAS PIC. If the takeoff and/or landing site is decoupled from the PIC, an approved designee must be permitted to assume this duty.

FAA UAS Interim Guidelines 08-01 require use of a ground- or chase aircraft-based observer to achieve SAA. Use of an observer produces gaps regarding AIM procedures and guidance. AIM §§4.4.1 and 4.4.12 define guidance stating that the PIC has overall authority regarding the safety of the aircraft regardless of any order issued by ATC. If the observers have greater situational awareness than the PIC, the chain of command in choosing how to respond to a safety-critical situation must be reconsidered. AIM §§4.4.14 and 5.5.1 both discuss the authority of ATC to command the aircraft in visual flight rules (VFR). It can be asked, should similar authority be given to the observer when, regarding vertical clearance, the surveillance capabilities of ATC within the terminal area are vastly superior to those of a ground observer? It may be necessary to set limits upon when a PIC has authority to accept or reject ATC orders and guidance from observers.

The need for pilot situational awareness and the ability to react quickly to adverse conditions conflicts with the physical decoupling of pilot and aircraft in UAS. 14 CFR §§25.253 and 23/25.671 discuss design of the flight control system to provide timely information to the pilot and to handle the commands in the event of a critical warning or sudden loss of control. UAS data link latency increases the time it takes for the pilot to become aware of an adverse situation and react. Maintaining an ELOS under these conditions is challenging. AIM §3.4.6 defines alert areas (e.g., areas of a heavy volume of training flights), and AIM §3.5.4 defines parachute jump areas. Higher situation awareness is also necessary when operating under special use airspace. UAS should be prohibited from both of these environments unless suitable SAA technology is approved. A remote pilot lacks the necessary situation awareness and the reaction time necessary to avoid a potential collision.

AIM §§4.1.19 and 4.4.14 define conditions in which the pilot would have to switch transponder or radio frequencies during transitions from various modes of flight. Current radios and transponders are equipped with knobs that must be turned to change frequencies. Approved alternative mechanisms must be developed to allow this transition to be triggered remotely or automatically.

94.5.2.3 Control

14 CFR §25.397 discusses mechanical loads placed upon the cockpit controls such as stick and wheel controls. Since the cockpit controls are decoupled from the control surfaces, this regulation is no longer applicable. Other mechanical requirements such as those in 14 CFR §§23.395 and 23.405 are no longer necessary as the aircraft is clearly in a fly-by-wire(less) control mode.

The requirement of status indicators to aid the pilots' situational awareness must also be reconsidered. 14 CFR §§25.1303, 23/25.1309, and 91.205 as well as AIM §1.1.19 define the indicators required for the pilot. Under the UAS paradigm, the indicator status information must be transmitted to the GCS and then displayed on the GCS flight control displays.

14 CFR §§23/25.1329 require the capability of the system to avoid becoming stuck in a hard-over. In the event of a GCS or data link failure, the aircraft's autopilot and control system should have sufficient intelligence to detect faults and prevent improper configurations from occurring.

14 CFR §23.679 discusses the need for lockouts of controls, while the aircraft is grounded to prevent accidental bumping of controls, but the lockout must be disabled and cannot be set during flight. For UAS, it may be necessary to include such mechanisms while in flight as well. Consider the need for a UAS operator (other than a PIC) to leave a position temporarily. It may be beneficial to enable a lockout mechanism given the lack of situational awareness to ensure the aircraft is not accidentally controlled by a 3rd party.

94.5.2.4 Communications

Regulatory gaps can be divided into communication segments. GCS-UA communication will focus upon the command data link. GCS-ATC will focus upon

ATC to remote pilot interactions, which may rely upon the UAS as part of the communication link.

GCS-UA Communications 14 CFR §25.143 requires the aircraft to be safely controllable and maneuverable during all phases of flight. The higher latency associated with BLOS operations adds challenges to meeting this regulation with an ELOS. Latency also impacts the remote pilot's use of SAA tools as TCAS and ADS-B, as well as a number of other data services provided by ATC must be rebroadcast from the UA to the remote pilot. The transmission of real-time state data for visual indicators specified in 14 CFR §§23.207, 25.1303, 25.1309, and 91.205 is similarly affected by latency.

There is also a need to address link loss detection and recovery, which is currently not regulated. 14 CFR §23.685 discusses the aircraft's mechanical design to preventing mechanical jamming throughout the aircraft's various linkages as essential to control systems. Loss of data link due to RF range or jamming (malicious or otherwise) represents a different but analogous type of jamming associated with controls. 14 CFR §91.511 requires aircraft to be equipped with sufficiently redundant communication systems for flight over bodies of water exceeding 30 min duration or 100 nautical miles extent. UAS component redundancy can be weight, cost, power, and space prohibitive.

GCS-ATC Communications The technology survey identified techniques to handle communication between the remote pilot located at the GCS and the controlling ATC facility. The major difference is that pilot communication does not originate from the aircraft, but the GCS. 14 CFR §91.185 requires procedures for handling a LL between a pilot and ATC. This regulation may be interpreted as requiring UAS to use a terrestrial telecommunications link from the GCS to the ATC in case the GCS-ATC radio link is lost.

AIM §4.1.15 discusses safety alert messages that require an immediate response by the pilot to avoid eminent damage to the aircraft and/or others; AIM §4.4.10 defines messages in which the word IMMEDIATELY may be added in order to inflect the urgency of the controller's command. As latency may make "immediate" infeasible, "immediate" may need to be replaced with a quantifiable time limit for the pilot to take action. AIM §5.3.1 defines en route procedures for pilots, and AIM §5.3.1.2 defines the controller pilot data link as a supplemental means of providing en route commands from the controller to the pilot. To ensure these en route directives are met, the UA must be equipped with a controller pilot data link, relaying that data back to the PIC at the GCS. Current guidelines do not require this capability and must be revised.

94.5.2.5 Other C3 Gaps

FAA TSOs (Federal Aviation Administration 2012c) are an additional source of regulatory gaps. For the TSO gap analysis, there is less concern regarding whether or not the performance standards are applicable with interpretation versus revision

as new TSOs may be issued with less effort than a change to the FARs. TSO-C9c: Automatic Pilots (Federal Aviation Administration 2012e) defines the requirements for automatic pilots, which must be approved for use in civil aircraft. This TSO is written toward an autopilot on a manned aircraft in which a pilot is in immediate control of the aircraft. The TSO references SAE Aeronautical Standards AS-402A (SAE International 2001). A detailed analysis of AS-402A yielded the following gaps. TSO-C9c §4.2.3 calls for “a controller, if present, it shall operate in the plane and with the sense of direction of motion of the aircraft. The control sensing shall be plainly identified on or adjacent to each control.” This requirement can be reinterpreted toward the requirements of the ground control station.

TSO-C9c §§4.3.1 and 4.3.2 call for “a means by which the pilot can be made cognizant of the condition, including control behavior” and “the direction and relative magnitude of the primary pitch servo present and other two axes.” This requirement may be re-interpreted toward the requirements of the ground control station. It must also be considered that any feedback of information to the GCS controls and indicators from the UA will be susceptible to latency.

TSO-C9c §4.4.1 calls for “corrective control to be: (a) Pitch $\pm 50^\circ$ (b) Roll $\pm 75^\circ$ (c) Yaw $\pm 20^\circ$.” Since the PIC is no longer in immediate control of the aircraft, the corrective controls about these three axes may require greater limitations of corrective controls, while the autopilot is engaged.

TSO-C9c §4.5.1 calls for “a system interlock to prevent the automatic pilot engagement until it has reached a fully operable condition.” For aircraft that handle autonomous takeoff and landing, this requirement is no longer relevant. However, for aircraft in which the PIC may manually remote control the aircraft, this requirement must be reinterpreted toward a requirement for the GCS.

TSO-C52b: Flight Director Equipment (Federal Aviation Administration 2012d) establishes minimum performance standards referencing SAE AS-8008 (SAE International 1984). The regulatory gaps identified are similar if not identical to that of TSO-C9c. TSO-C52b §3.6 calls for identical corrective control capabilities for the pilot, while the autopilot is engaged as TSO-C9c §4.1.1. Likewise, TSO-C52b §3.8 calls for the same lockout mechanism as defined by TSO-C9c §4.5.1. The recommendations for interpretation remain the same for each.

94.6 Case Study #4: Emergency Recovery and Flight Termination Systems for UAS

94.6.1 Technology Survey

To maintain a sufficient level of safety, UAS are equipped with systems to detect faults and failures of onboard components, including electromechanical systems (e.g., control-surface actuators) and avionics systems (e.g., data links, actuator controllers, and sensors). In emergency recovery (ER), the aircraft determines that a fault has occurred, recovering from it while maintaining safe flight. Link loss (LL) procedures, addressing and rectifying failure of command and control data links, are

a subset of ER. UAS experiencing LL or GPS loss may leave a predefined area for operation, the bounding box (BB), and need to be brought down. Flight termination (FT) technologies and procedure end the UA flight while minimizing the risk to the public and property. A flight termination system (FTS) is an onboard system that executes a flight termination, which may be remotely triggered via an independent communications channel by the PIC or automatically performed by the UA on the basis of aircraft conditions.

Information was gathered for UAS systems and related technologies related to emergency recovery, link loss, and flight termination. Data sheets and other materials were collected on a number of UAS and UAS subsystems. A questionnaire was developed and distributed to UAS researchers and industry workers, but very few vendors replied, and fewer were willing to provide data. After conversations with individuals in academia and industry, it was determined that the questions were too concerned with proprietary information and had the potential to expose issues manufacturers may not want made public, including experimental flight test results.

Figure 94.4 presents the developed ERFT conceptual framework. From left to right, the criticality of a vehicle loss and ramifications of such a loss increase. When criticality is low, health-based recovery systems diagnose and correct the problem, and the vehicle continues onward. With greater criticality, mission-level contingency systems handle an emergency recovery, often accompanied by termination of the aircraft’s mission. The final and most extreme response is flight termination. In addition to this framework above, UAS pilot procedures differ from those for manned aircraft with the difference in some cases tied to the technology being used, but in others coming from fundamental differences between UAS and manned aircraft, and their operations.

Table 94.19 summarizes the ERFT capabilities of several surveyed autopilots for small UAS (Vaglienti et al. 2008; Procerus Technologies 2008a; MicroPilot Inc. 2005). Table 94.20 presents the ERFT capabilities of several of the surveyed aircraft (Heppe, personal communications, Insitu, inc., 2008; McDuffy, personal communications, Insitu inc., 2008; Butler and Loney 1995; Flightglobal 2009b; Winstead 2008; Flightglobal 2009d; Flightglobal 2009a; Donaldson and Lake 2007).

Health-Based Recovery Health-based recovery systems handle less extreme aircraft system faults and failures in which the aircraft’s mission continues by adjusting the vehicular attributes. Redundancy is the more common framework for

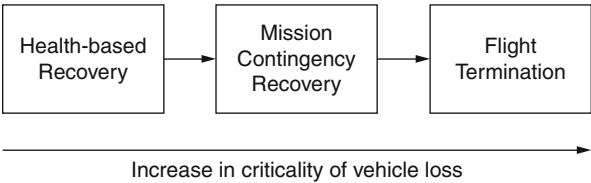


Fig. 94.4 A framework for guidance of ERFT technology survey

Table 94.19 Autopilot ERFT capabilities

| Autopilot | Contingency-based recovery | Flight termination |
|------------------------|--------------------------------------|--|
| Cloudcap Piccolo | Return to waypoint | Mission selectable from close throttle, aerodynamic termination, and/or deploy parachute |
| Procerus Kestrel | Shallow bank until restored | Aerodynamic termination |
| Micropilot MP-2028g | Mission selectable (see next column) | Mission selectable from fly to climb, descend, roll, eject chute, etc. |

Table 94.20 Known ERFT capabilities for surveyed aircraft

| Aircraft | Manufacturer | Contingency-based recovery | Flight termination |
|-------------|--------------------------------|---|--|
| ScanEagle | Insitu, Inc. | Loiter at point for lost link | Aerodynamic termination if departs mission area |
| Predator | General Atomics | Return home for lost link | Optional parachute for early models |
| Global Hawk | Northrop Grumman | Contingency flight paths for various emergency/ contingency modes | Terminate with extreme prejudice |
| Polecat | Lockheed Martin | Unknown | Terminate with extreme prejudice |
| X-48B | Boeing and Cranfield Aerospace | Unknown | Parachute, airbags, and spin parachute (for stall testing) |
| Arrow | Jordon Military | Unknown | Parachute and flotation device |

health-based recovery, while fault detection, identification, and recovery (FDIR) is a recent approach to health-based recovery. With sufficient redundancy, when a component becomes non-functional, it is possible for the control system to transition to a backup system and continue nominal operation. A European UAS research commission recently funded the development of a medium-altitude UA equipped with a redundant engine (Flightglobal 2009c). Redundancy is common in civil and military aviation, with dual or triple redundant systems used for safety-critical hydraulic, electrical, and computational components within aircraft.

For fault detection, residuals representing the error between the expected and actual responses of the aircraft’s systems are calculated. Fault identification analyzes the residuals to identify the cause of the problem. Fault recovery adjusts the control system dynamically, reducing the impact of the failure and restoring nominal operation. If the recovery system is unable to perform such a restoration, then a

mission contingency plan or a flight termination system may be activated (Rotstein et al. 2006; Atkins 2004; Heredia et al. 2004).

Mission Contingency Recovery In mission contingency recovery, the aircraft deviates from its current mission to one of several possible emergency-recovery modes, altering its flight path to mitigate risk due to the failure and/or allow safe recovery of the aircraft. As an example, the Global Hawk UAS possesses a sophisticated contingency management system (CMS) with recovery modes including link loss recovery, return-to-base command, abort landing command, and land now command (Winstead 2008). For each of these, the CMS redirects the aircraft to a flight path appropriate for the current mode, requiring all points along the mission route to have contingency routes that branch off for each contingency mode. Along a contingency route, additional contingency routes can be branched off should additional failures occur. Figure 94.5 presents a primary flight path and a number of contingency branches for Global Hawk. In this figure, only three abstract contingency modes, C1, C2, and C3, are considered. Such an elaborate contingency plan allows ATC along the mission and contingency paths to be aware of the aircraft's potential presence and the circumstances for that presence.

As it is unsafe to operate a UAS when the control data link between the aircraft and the ground control operator is down, lost link procedures are considered contingency-based recovery. The survey revealed that the most common LL procedure is for the aircraft to fly to a predefined location, where it can either loiter until the link is restored, autonomously land, or be remotely piloted via secondary data link (Walker 1997; McMinn and Jackson 2002; National Transportation Safety

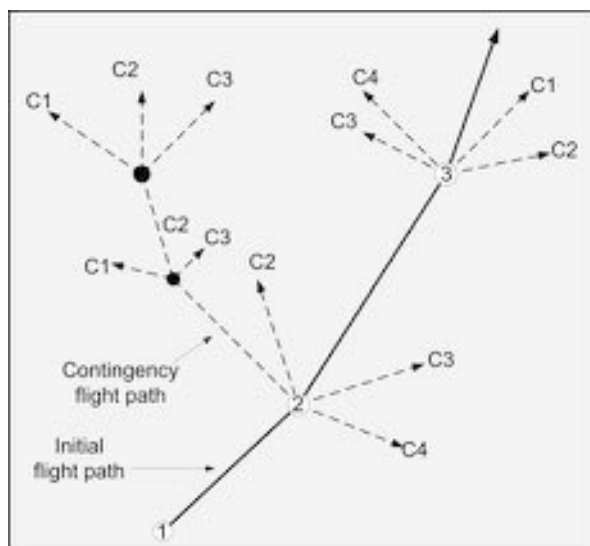


Fig. 94.5 Contingency routes for Global Hawk (Winstead 2008)

Board 2008; Ro et al. 2007). The BAT III LL procedure is a simple return home functionality, flying directly to the last known location of the GCS (Ro et al. 2007). Within sufficient range of the GCS, a remote pilot controls the aircraft to land. NASA and Boeing's PhantomWorks X-36 follows a similar method of returning to base and loitering (Walker 1997), but rather than return to base directly, the aircraft follows a predefined return path. Researchers at NASA Dryden are developing a path-planning algorithm for return-to-base and LL operations ensuring that the UA stays within its authorized flight zone (McMinn and Jackson 2002). LL procedures for BLOS operation in either medium-endurance or high-endurance UA are nearly identical to LOS operations. Altair flew in NAS for Western States Fire Imaging Mission. During one of its missions, the UA had a modem malfunction, resulting in BLOS Ku band C2 LL. The aircraft switched to C band and flew to a predetermined loiter point until the link was reestablished (Ambrosia et al. 2007).

For small UAS, commercial autopilots have contingency management features for link loss. The Piccolo Autopilot (Vaglienti et al. 2008) supports a lost communication timeout in seconds. If after that specified time a message from the GCS has not been received, the aircraft flies to a LL waypoint. The Procerus Kestrel lost link procedure returns the aircraft either to base or an alternate "rally point" (Procerus Technologies 2008b). Micropilot's various autopilots allow users to define the response to the lost link procedure and the criteria for diagnosing the lost link (Micropilot 2008). Its LL procedure supports the return to any waypoint or alternatively to trigger a FTS.

Flight Termination A FTS brings an aircraft down expeditiously while maintaining an appropriate level of safety to public and property. Given sufficient redundancy, a FTS may not be necessary, but two motivating factors for having such a system include insufficient redundancy, often the case for smaller UAS, and the FTS being mandated per conditions of the restricted airspace in which the aircraft is flying (i.e., range safety).

One approach is to aerodynamically terminate a flight by setting the aircraft's control surfaces to a state resulting in the vehicle crashing into the ground or a body of water in a semi-controlled manner. One form of aerodynamic termination is to perform a slow downward spiral, allowing some aircraft damage to be mitigated by a slow descent and the aircraft's final position to be controlled. This mechanism is ideal under airspace violation events, as it prevents deeper intrusion into an unauthorized airspace. This technique is used by the Insitu ScanEagle (McDuffy, personal communications, Insitu inc., 2008) and is also provided as a feature by the Piccolo (Vaglienti et al. 2008), Micropilot autopilots (Micropilot 2008), and the Kestrel autopilots (Procerus Technologies 2008b). In glide-path descents, the aircraft glides from its current altitude to a landing site without engine power. Under a glide-path termination, a suitable landing site may be designated by the UA PIC (Atkins 2004) or autonomously (Atkins 2004; Fitzgerald et al. 2005). Glide-path descents for high-altitude UAS allow the aircraft to terminate in a region in which the risk can be mitigated, as the aircraft can glide out over the ocean or an unoccupied area before impacting with the surface.

Several ballistic recovery systems are available to handle flight termination of an unmanned aircraft. Parachutes have a history of use in manned aircraft, and some technical standard orders exist (Federal Aviation Administration 2009c). Autopilots such as the Piccolo (Vaglianti et al. 2008) and Kestrel (Procerus Technologies 2008b) allow parachute deployment to be part of the FTS if the target aircraft is appropriately equipped. While not a standard feature, parachutes have been installed on the Predator UAS (Butler and Loney 1995), and a number of other UAS are parachute equipped (Flightglobal 2009c; Donaldson and Lake 2007; Procerus Technologies 2008b; Micropilot 2008; Vaglianti et al. 2008). Parafoil parachutes provide additional loft permitting greater control for the aircraft such that it is possible to achieve a glide-path approach (Fitzgerald et al. 2005). This is used on the BAE SkyEye and the IAI I-View (Donaldson and Lake 2007). Some aircraft such as the X-48B also include spin parachutes, which aid in recovery of an aircraft caught in a spin. Airbags or flotation devices may accompany parachute-based FTS. The X-48b is equipped with airbags to reduce the forces at impact (Flightglobal 2009a). The Jordon Arrow is equipped with a foam body in order to remain buoyant if the aircraft is terminated in the water (Donaldson and Lake 2007).

94.6.2 Regulatory Gap Analysis

This section presents the process and results of the ERFT regulatory gap analysis. Title 14 CFR Parts 23, 25, 27, 29, and 91 were reviewed as well as guidance materials, including the Aeronautical Information Manual (AIM) (Federal Aviation Administration 2012a), Airplane Flying Handbook (AFH) (Federal Aviation Administration 2009a), and Helicopter Flying Handbook (Federal Aviation Administration 2009b). Regulatory gaps were organized based upon aspects of pilot/crew procedures, health-based recovery, contingency-based recovery, and flight termination. Both fundamental gaps, regulatory gaps that exist because of the difference between technologies when regulations were written and UAS technologies utilized, as well as open-set gaps, gaps due to UAS technologies that have no analogue with currently regulated technologies, are identified.

The gap analysis was performed by an iterative process. The regulatory and guidance materials to be considered were determined and collected, followed by an initial review employing coarse filtering to identify ERFT-relevant sections of those materials. Rubrics were developed for each of the four aspects to determine the level of applicability of each section, introducing greater transparency and consistency in identifying gaps. The rubrics provided aspect-specific criteria to facilitate consistent classification of the section as applying as is, applying with interpretation, applying with revision, or not applying. Their length precludes their being included here; for the full text of the rubrics, see Stansbury et al. (2009a). A representative example of a rubric is shown in Table 94.21 for assessment of regulations/procedures related to pilot procedures. Using the rubrics, team members analyzed the identified sections, adding annotations to justify each classification. Chapter 16 of AFH was analyzed through a less-formal procedure by deriving the implications of manned emergency

Table 94.21 Rubric for assessing regulations related to pilot procedures

| | |
|-----------------------------|---|
| Does not apply | Regulation or guidance material does not discuss procedures relevant to the emergency recovery/contingency procedures to mitigate risk |
| Applies as is | Regulation or guidance material discusses procedures relevant to the emergency recovery/contingency procedures to mitigate risk. Given current language, applicable as is need for interpretation or revision for UAS paradigm |
| Applies with interpretation | Regulation or guidance material discusses procedures relevant to the emergency recovery/contingency procedures to mitigate risk. Parts of the language of the regulation require interpretation toward equivalent operations for unmanned aircraft |
| Applies with revision | Regulation or guidance material discusses procedures relevant to the emergency recovery/contingency procedures to mitigate risk. Regulation defines procedures of the pilot for safe operation within NAS that are unachievable for UAS given the language as it is written |

procedures for UAS. The results of the several analyses, including the annotations, were discussed to determine a consensus as to the level of applicability of each section. The discussion below focuses upon regulations that required revision or interpretation. Fundamental gaps and open set gaps were also identified. The result of the gap analysis follows.

Pilot and Crew Procedure Gaps Title 14 CFR §91 and the AIM were examined regarding pilot and crew procedures. The AFH was also examined as it indicates expectations of the PIC of a manned aircraft in an emergency. Pilot and crew procedures defined within 14 CFR Part 91 assume the pilot, crew, and passengers are onboard the aircraft. While the pilot of a UAS may no longer be onboard the aircraft, it is possible for a remotely piloted or autonomous aircraft to carry passengers or crew in the not too distant future. These regulations cannot simply be dismissed for “unmanned” aircraft, but rather must be interpreted or revised to be appropriate for cases in which pilots, crew, and/or passengers are or are not onboard. Examples of these regulatory gap include 14 CFR §§91.509, 91.511, and 91.513, which define survival equipment for emergency evacuation for overwater flights, and 14 CFR §91.501, which requires any crewmember onboard the aircraft to be familiar with the emergency equipment and emergency procedures onboard the aircraft before flight. This regulation is also written with the assumption of the PIC being onboard the aircraft and must be reinterpreted.

Procedures for normal flight operations impacted because the pilot is not aboard the aircraft include operations where a pilot would diagnose or respond to an emergency situation that could be handled through health-based recovery, contingency-based recovery system, or flight termination. An example is AIM §§5-4-11, 5-4-14, and 5-4-16, defining arrival procedures, including instrument approaches and simultaneous landing approaches. Under these conditions requiring fast reaction times, the need to abort a landing or deviate from an arrival path could be better handled through contingency-based recovery, which can both restore safety and send a notification to ATC. Similar issues of deviation and notification of ATC exist in AIM §§6-1-1, 6-1-2, and 6-2-1, which define emergency procedures, and AIM §7-1-14, which defines weather avoidance assistance procedures. Other procedure-related guidance includes AIM §6-2-5, which defines requirements for use of the onboard emergency locator during emergencies such as ditching, which under the UAS paradigm ought to be activated by the flight termination system; AIM §6-3-3, which defines procedures for selecting a suitable glide path to ditch the aircraft, which under the UAS paradigm could be performed by the PIC, FT system, or both; and AIM §6-4-2, which defines procedures for a pilot handling the loss of a communication link with ATC. The health-based recovery should be responsible for diagnosing the issue and switching to a redundant communication system if available; if communication is not restored, the contingency management is better capable of handling the procedures for transponder settings to alert ATC of the issue and performing appropriate LL procedure.

Throughout the regulations and the AIM, the pilot is assumed to be capable of issuing a distress call and/or rapidly communicating any sudden deviations with ATC. Given the significant changes to the communication paradigm between pilot, aircraft, and ATC, these procedures need to be revised or significantly reinterpreted. Guidance materials falling under this gap include AIM §5-3-1, which is written assuming traditional communication paradigms for manned aircraft, instead of the aircraft acting as a relay; AIM §§6-1-1, 6-1-2, and 6-2-1, which discuss the ability to deviate from standard procedures in an emergency; AIM §6-3-1, which discusses distress communications where the PIC must provide immediate notification and response to notifications dependent upon the condition and the directives of ATC, with “immediate” being made difficult because of latency; and AIM §6-3-2, which discusses request for emergency assistance when flying under distress.

Additional pilot- and crew-related gaps from Title 14 CFR and AIM include 14 CFR §91.609, which establishes the requirement for flight data recorders and cockpit voice recorders in transport category aircraft; AIM §1-1-19, which assumes that onboard global position system (GPS) would be identical to the currently technical standard order (TSO)-defined units featuring a graphical display for the pilot. Under the UAS paradigm, the GPS can likely be different from these TSOed GPS units and thus may require entirely different procedures for addressing a minor or major GPS failure. Aspects of chapter 16 of the AFH suggest that a new approach will be required regarding emergency situations in UAS. In a traditionally piloted aircraft, the pilot uses visual means to best determine the location in event of an unplanned landing. UAS require either dedicated space in which to fly or technology

to implement determination of the location for a flight-termination event. There is currently no performance specification for such a technological solution. Also, the necessity and means of informing other aircraft in the event the UAS engages in ERFT procedures should be specified. Finally, in the event of a ground-based or airborne observer, procedures should be specified to handle loss of visual contact with the UA, whether due to IMC or some other situation.

Health-based Recovery Gaps For health-based recovery systems, the gap analysis focused upon regulations for equipment that identified potential risks and mitigated them through corrective measures that did not alter the aircraft's current flight plan. Similar to the procedural gaps, the physical disconnect of the pilot from the aircraft lead to situations in which the regulation must be interpreted or revised toward the use of a health-based recovery system to address the situation. 14 CFR §§23/25/27/29.672 mandate that an indicator light notify the pilot if there is a loss in stability control. Under the UAS paradigm, due to data link latency, an indicator light may not be sufficient for notifying the pilot of this situation. The regulation also calls for the aircraft's control to be recoverable by the pilot. However, a health-based recovery system could be capable of dynamically reacting to the fault and recovering stability control. If such a system were onboard the aircraft, it would be necessary that it be demonstrated to provide an ELOS, and the regulation must be reinterpreted to consider such an alternative.

Engine fire suppression systems as mandated by 14 CFR §§23/25.1195 can be considered a health-based recovery system currently onboard some manned aircraft. This regulation requires revision for a number of reasons. The propulsion system of a UAS may not be based upon the use of an internal combustion (e.g., fuel cell, electric motors), which would likely not need a fire suppression system. Based upon the size of an aircraft, requiring such a system may produce a significant burden. For instance, a small hand-launched UAS could likely be incapable of handling such a system and remain airworthy.

Health-based systems may be capable of handling a variety of different failures either by initiating the appropriate transition to a redundant system or by diagnosing and recovering the failure directly. As a result, regulations such as 14 CFR §§27.695 and 29.695, which discuss recovery from power and control failures, ought to be reinterpreted to consider the health-based recovery system operating in place of the pilot in command. Similarly, health-based recovery may be capable of providing the "immediate" response to some emergency situations defined in the AIM such as AIM §5-4-11.

In AIM §§1-1-1, 1-1-12, and 1-1-20, it is assumed that minor failures of navigation aids may be detected by the pilot in command and the pilot can then act appropriately to maintain safe flight. Under the UAS ERFT framework, a health-based recovery system could both detect and recover from a minor fault without having a significant impact on flight. Voice communication loss and recovery is discussed in AIM §6-4-1. A health-based recovery system could be utilized in place of pilot procedures to autonomously change to a redundant communication link.

Contingency-Based Recovery Gaps Because contingency-based recovery results in the UAS deliberately changing course to recover from some emergency condition, the procedural guidelines as defined in the AIM may require some re-interpretation because of the physical separation from the UAS with its contingency-based recovery system and the pilot in command. For instance, AIM §4-3-5 defines the necessity to quickly notify ATC in the event of a land abort. Since such a deviation may be executed by the contingency-based recovery system, the system must either alert ATC directly or communicate the event to the PIC who then notifies ATC resulting in added delay. This gap exists for a number of sections of the AIM including AIM §§4-3-5, 4-4-1, and 5-5-2.

Lost link procedures also generate a number of gaps. Current regulations do not address the command data link, but do address the voice communication channel. Recovery procedures for voice communication loss are defined in AIM §§6-4-1 and 6-4-2. The latter requires the pilot to notify ATC via a change in transponder setting. If the aircraft is used as a communication relay and the loss of communication with ATC occurs as a result in the link from GCS to UA failing, the pilot would be unable to command such setting change. Diagnosis and recovery of communication failures, however, could be handled entirely without the PIC actions as part of a predefined set of lost link procedures onboard the aircraft. It may be useful to define separate transponder alert settings for voice and data communication losses, respectively.

A number of other contingency situations occur in which the contingency-based recovery system would override the actions specified in the AIM. Specific examples include AIM §§1-1-1, 1-1-12, and 1-1-20 for corrective measures in the event of a navigation aide failure; AIM §7-1-14 for utilizing weather avoidance assistance; etc.

Flight Termination Gaps The examination of regulations and procedures related to flight termination focused upon those that addressed the ditching or emergency landing of an aircraft. One major gap that exists with respect to FT is the structural and restraint requirements for passengers and crew onboard aircraft. For UAS, it might be assumed that no pilot is onboard; however, in the future unpowered aircraft may still be used for passenger or cargo transport. In the case in which the intended design and use of the aircraft would not support any humans onboard, these safety regulations would not be applicable. If the intent is for passengers or crew, it should be possible to mandate only those safety elements necessary to support that use. Impacted regulations include 14 CFR §§23/25/27/29.561, 23/25/27/29.562, 23/25/29.803, 23/27/29.805, 23/25/27/29.807, 25/29.809, 25.810, 23/25/29.811, 23/25/29.812, and 23/25/29.813. Similarly, emergency/survival equipment and crew training on procedures to utilize this equipment are mandated in 14 CFR §91.501, 91.509, 91.511, and 91.513.

14 CFR §91.609 establishes the requirement for flight data recorders and cockpit voice recorders in transport category aircraft. This regulation provides no exemption for aircraft without a pilot in the cockpit, such as an unmanned transport-category aircraft. While a voice recorder could be useful to record communication between

ATC and the GCS-based operators, the wording of this requirement will likely require revision for it to fit correctly. 14 CFR §§25.1457, 27.1457, and 29.1457 also address cockpit voice recorders, though it provides the requirements of the device, rather than the requirement to have the device.

The AIM and chapter 16 of the AFH define procedures for the pilot if the aircraft must be ditched. AIM §6-2-5 defines the requirements for triggering the emergency locator upon ditching the aircraft. AIM §6-3-3 discusses the procedures for finding a suitable crash glide path to ditch the aircraft in water. In all of these cases, it may be possible to automate these tasks as part of the onboard flight termination system.

Fundamental Gaps During the gap analysis, fundamental gaps were identifiable because a significant number of regulations were identified as having gaps and the justifications for these gaps were very similar. The physical decoupling of the PIC from the aircraft qualifies as one of the largest fundamental gaps as it results in the largest number of regulatory gaps related to procedures and airworthiness regulations. For instance, any procedure requiring an immediate response must be reevaluated as “immediate” may no longer be achievable as currently understood due to latency and a lack of situational awareness.

A number of regulations and procedures are written regarding the safety of passengers and crew onboard the aircraft; however, one of the fundamental differences between UAS and manned aircraft is that a UAS can be constructed for operation solely without passengers or crew. All existing relevant regulations must be reinterpreted, revised, or eliminated from applicability for UAS. The GCS generates an additional fundamental gap as existing regulations for cockpit layout and equipment must be reinterpreted and/or revised in order to support aircraft control remotely. It may be necessary to revise existing regulations as some traditional cockpit controls have been eliminated in place of a mouse and keyboard being used to define waypoints and the autopilot controlling flight surfaces.

Under the AFH and HFH, procedures existed for the pilot to down the aircraft. A fundamental gap exists in regard to how procedures are written for UAS regarding emergency flight termination. A flight termination system such as a parachute is available to UAS, which is rarely seen or used by manned aircraft. The kinetic energy of a UAS can be dramatically different than a manned aircraft such that terrain considered previously unsuitable for an emergency landing may be adequate.

Open Set Gaps Several pieces of aircraft equipment must be evaluated to determine what regulations must be defined for their usage within an unmanned aircraft, as they are not typically found within a manned aircraft, including command data link; ATC/GCS voice link; GCS components; situational awareness sensors such as any onboard cameras, radars, and auditory sensors; health-based recovery system; contingency management system; and flight termination system (explosives or ballistic recovery system).

The PIC can only communicate with the aircraft via a command data link at the GCS. An open set gap exists because manned aircraft are not equipped with data

links for this purpose, and no existing regulations exist for the command data link as part of 14 CFR. Regulations must exist to define performance, link loss procedures, data frequencies, message sets, etc. In the event of a mishap, UAS technology must be equipped with a contingency management system capable of identifying the failure and then executing one of several possible predefined actions. A number of possible approaches are available. The open set gap occurs because nowhere in the current regulatory framework is such a system mandated nor its minimum performance requirements defined. Additional open set gaps exist for the roles of the UAS flight crew. Ground observer qualifications including training, health, visual acuity, etc. are not currently defined as part of the regulations.

94.7 Conclusion

This chapter has discussed the role of technology surveys and regulatory gap analyses in supporting the FAA and policy makers in understanding how the current regulatory environment must adapt to accept the near-disruptive technology of UAS. Earlier in the report, the concepts of the technology survey and regulatory gap analysis were presented, but without any specific best practices.

Each case study summarized published research conducted by the ERAU team over the course of several years. The first two studies involved propulsion systems for UAS (Griffis et al. 2008; Griffis and Wilson 2009). Next, C3 (Stansbury et al. 2008, 2009d) and SAA (Reynolds and Wilson 2008a, b) were studied. Finally, given the lessons learned from the previous studies, the ERFT study (Stansbury et al. 2009a, b, c) tried to pool together the best practices from the previous studies in order to produce a less subjective analysis of the regulations. It is recommended that the reader seek out these papers for further information and more detailed versions of their respective technology surveys and gap analyses.

This chapter concludes with the authors attempt to convey some recommended practices for conducting technology surveys and regulatory gap analyses. This shall hopefully allow those in the UAS community seeking to analyze other technology areas to start with a solid iterative foundation for conducting their studies.

94.7.1 Guidance on Performing a UAS Technology Survey

An iterative process is recommended for conducting a technology survey for a UAS. First, a web search will provide a summary of the market for a specific UAS technology. The preliminary results will likely be a list of UAS utilizing the technology, manufacturers of that technology, some major categories of the technology, and some high-level information on how the technology is applied to UAS. Second, given these preliminary results, a second survey of literature (both online and print published) should be conducted to derive a deeper understanding of the technology (i.e., specific data sheets, user manuals, guidance materials). The resulting data should be compiled. Third, a framework should be derived for

conceptually representing the technology, its subcategories, and its relationship to UAS. This model is used to help define the key points of variation among representative systems. Lastly, utilizing this model, a final technology survey is conducted in which collected references and additional research results are applied to the model to clearly articulate the representative cases of the various technologies as applied to the context of UAS.

94.7.2 Guidance on Performing a Regulatory Gap Analysis

The gap analysis proceeds in steps. The first step is to review Title 14 of the Code of Federal Regulations to determine putative applicability of the regulation to the particular technology at hand. Putative applicability is determined by as inclusive of an applicability criterion as possible: if the regulation could be construed through even a generous interpretation to apply to the technology, it is included in the putative list.

The second step is to develop rubrics which classify the regulation applied to the technology as one of applies, applies with interpretation, applies with revision, and does not apply. A rubric is developed for each element of the conceptual model or framework for the technologies under review. The language of the rubric should be such that a reasonably well-informed individual using the rubric should come to similar conclusions as to whether any given regulation applies to the technology at hand.

The third step is to apply the appropriate rubric of step two to each of the regulations in 14 CFR 1–199. Multiple individuals should perform step three, such that their results can be compared. Differing results should be discussed until agreement is reached as to the proper classification. Annotations should be provided to identify the rationale for the classifications made whenever necessary.

The fourth and final step is to compare the results of step three with the putative list from step one. Disagreements between the step three list and the putative list should be re-examined to ensure that the classification is appropriate.

References

- AAI Corp, Unmanned aircraft systems (2008), online, <http://www.aaicorp.com/New/UAS/index.htm>
- AC Propulsion, AC propulsion's solar electric powered solong uav (2005), online, http://www.acpropulsion.com/ACP_PDFs/ACP_SoLong_Solar_UAV_2005-06-05.pdf
- Access 5, Cooperative conflict avoidance sensor trade study report V.2. Technical report, NASA Access 5, Edwards, CA, 2004
- Advance Ceramic Research, Unmanned vehicle systems (2008), online, <http://www.acrtucson.com/UAV/index.htm>
- AeroVironment, Unmanned aircraft systems (2008), online, http://www.avinc.com/UAS_products.asp
- AIAA, UAV programs around the world. Aerospace America, issue supplement, 2005
- AIAA, 2011 worldwide UAV roundup, Poster, 2011

- V.G. Ambrosia, B. Cobleigh, C. Jennison, S. Wegener, Recent experiences with operating UAS in the NAS, in *AIAA Infotech Aerospace 2007 Conference and Exhibit*, Rohnert Park, California, AIAA 2007-3007, 2007
- AREN, Mazda Wankel Rotary Engines for aircraft website (2006), online, <http://www.rotaryeng.net>
- Army-Technologycom, Army technology – CL289 – unmanned aerial vehicle (2007), online, <http://www.technology.com/projects/cl289/>
- E.M. Atkins, Dynamic waypoint generation given reduced flight performance, in *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, AIAA 2004-779, 2004
- M. Brian, How rocket engines work (2007), online, <http://science.howstuffworks.com/rocket.htm>
- N. Brown, A. Samuel, S. Bhandar, R. Colgren, D. Schinstock, J. Lookadoo, Modular wireless avonic system for autonomous UAVs, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, AIAA 2006-6683, 2006, pp. 21–24
- M.C. Butler, T. Loney, Design, development and testing of a recovery system for the Predator UAV, in *13th AIAA Aerodynamic Decelerator Systems Technology Conference*, Clearwater Beach, AIAA 95-1573, 1995
- Defense Update International Online Defense Magazine, Desert Hawk Miniature UAV (2006), online, <http://www.defense-update.com/products/d/deserthawk.htm>
- P. Donaldson, D. Lake (eds.), *Unmanned Vehicles Handbook 2008* (Shephard Press, Ltd., Berkshire, UK, 2007)
- Electricity Storage Association, Technologies – supercapacitors (2007), online, http://electricitystorage.org/tech/technologies_technologies_supercapacitor.htm
- Federal Aviation Administration, Automatic dependent surveillance – broadcast (ADS-B) out performance requirements to support air traffic control (ATC) service. Technical report, Docket no. FAA-2007-29305, Department of Transportation: FAA, 2007, http://www.faa.gov/aircraft/air_cert/continued_operation/ad/
- Federal Aviation Administration, Interim operational approval guidance 08-01: Unmanned aircraft systems operations in the national airspace system, Technical report, Federal Aviation Administration. Aviation Safety Unmanned Aircraft Program Office AIR-160, 2008
- Federal Aviation Administration, Airplane flying handbook (2009a), http://www.faa.gov/library/manuals/aircraft/airplane_handbook/
- Federal Aviation Administration, Helicopter flying handbook (2009b), <http://www.faa.gov/library/manuals/aircraft/media/faq-h-8083-21.pdf>
- Federal Aviation Administration, TSO-C23d: Personnel parachute assemblies (2009c), [http://rgl.faa.gov/Regulatory_andGuidanceLibrary/rgTSO.nsf/0/00493ac675eda12e86256da500600ef7/\\$FILE/C23d.pdf](http://rgl.faa.gov/Regulatory_andGuidanceLibrary/rgTSO.nsf/0/00493ac675eda12e86256da500600ef7/$FILE/C23d.pdf)
- Federal Aviation Administration, Evaluation of candidate functions for traffic alert and collision avoidance system II (TCAS II) on unmanned aircraft system (UAS) (2011), online, http://www.faa.gov/about/initiatives/uas/media/TCASonUAS_FinalReport.pdf
- Federal Aviation Administration, Aeronautical information manual (2012a), online, http://www.faa.gov/air_traffic/publications/atpubs/aim/
- Federal Aviation Administration, TCAS home page (2012b), online, <http://adsb.tc.faa.gov/TCAS.htm>
- Federal Aviation Administration, Technical standard order (TSO) (2012c), online, http://www.faa.gov/aircraft/air_cert/design_approvals/tso/
- Federal Aviation Administration, TSO-C52b: Flight director equipment (2012d), online, http://rgl.faa.gov/Regulatory_and_Guidance_Library/rgTSO.nsf/0/56EF54910099134186256DC100600602?OpenDocument
- Federal Aviation Administration, TSO-C9c: Automatic pilot (2012e), http://rgl.faa.gov/Regulatory_and_Guidance_Library/rgTSO.nsf/0/4D729BA5BDF5851286256DA4005DC0AD?OpenDocument
- D. Fitzgerald, R. Walker, D. Campbell, Vision based emergency forced landing system for an autonomous UAV, in *Proceedings of the Australian International Aerospace Congress Conference*, Melbourne, Australia, 2005, pp. 397–402

- Flightglobal, British blend: UAV x-planes help boeing with blended wing concept (2009a), online, <http://www.flightglobal.com/articles/2006/05/30/206893/british-blend-uav-x-planes-help-boeing-with-blended-wing.html>
- Flightglobal, Global Hawk downed by rouge abort signal (2009b), online, <http://www.flightglobal.com/articles/1999/10/06/56882/global-hawk-downed-by-rouge-abort-signal.html>
- Flightglobal, Grand designs (2009c), online, <http://www.flightglobal.com/articles/2005/06/07/198916/grand-designs.html>
- Flightglobal, Lockheed confirms P-175 Polecat UAV crash (2009d), online, <http://www.flightglobal.com/articles/2007/03/20/212700/lockheed-confirms-p-175-polecat-uav-crash.html>
- L. Frater, E. Stokes, R. Lee, T. Oriola, An overview of the framework of current regulation affecting the development and marketing of nanomaterials. Technical report, ESRC Centre for business relationships accountability sustainability and society (BRASS), Cardiff University, 2006
- E.W. Frew, C. Dixon, J. Elston, B. Agrow, T.X. Brown, Networked communication, command, and control of unmanned aircraft systems. *J. Aerosp. Comput. Inf. Commun.* **5**, 84–107 (2008)
- General Atomics Aeronautical Systems, Inc, Aircraft platforms (2008), online, <http://www.ga-asi.com/products/index.php>
- Global Security, Common Data Link (2008), online, <http://www.globalsecurity.org/intell/systems/cdl.htm>
- Globalstar, Globalstar, Inc. – worldwide satellite voice and data products and services for customers around the globe, (2008), online, <http://www.globalstar.com>
- C.L. Griffiths, T.A. Wilson, A conceptual framework for UAS propulsion applied to risk and regulatory gap analyses, in *SAE 2009 AeroTech Congress and Exhibition*, Seattle, WA, 2009
- C.L. Griffiths, T. Wilson, J. Schneider, P. Pierpont, UAS propulsion systems technology survey, Technical report, U.S. Department of Transportation: Federal Aviation Administration, 2007
- C.L. Griffiths, T.A. Wilson, J.A. Schneider, P.S. Pierpont, Framework for the conceptual decomposition of unmanned aircraft propulsion systems, in *Proceedings of the 2008 IEEE Aerospace Conference*, 2008
- R.D. Hale, W.R. Donovan, M. Ewin, K. Siegele, R. Jager, E. Leong, W.B. Liu, The Meridian UAS: detailed design review, Technical report, TR-124, Center for Remote Sensing of Ice Sheets. The University of Kansas. Lawrence, Kansas, 2007
- D. Hall, B. Hosken, R. Wagner Robotics instruction course (2003), online, <http://teamster.usc.edu/~fixture/Robotics/Course.htm>
- G. Heredia, V. Remu, A. Ollero, R. Mahtani, M. Musal, Actuator fault detection in autonomous helicopters, in *Proceedings of the 5th IFAX Symposium on Intelligent Autonomous Vehicles (IAV 2004)*, Lisbon, Portugal, 2004
- S. Hottman, K. Hansen, M. Berry, Review of detect, sense, and avoid technologies for unmanned aircraft systems. Technical report, U.S. Department of Transportation: FAA, 2007
- INMARSAT, Aeronautical services (2008), Online, <http://www.inmarsat.com/Services/Aeronautical/default.aspx?language=EN&textonly=False>
- Insitu, Insitu unmanned aircraft systems (2008), <http://www.insitu.com/uas>
- Iridium, Aviation equipment (2008), online, <http://www.iridium.com/products/product.htm>
- L. Kirk, D. Marshall, B. Trapnell, G. Frushour, Unmanned aircraft system regulatory review. Technical report, US. Department of Transportation: Federal Aviation Administration, 2007
- J.D. McMinn, E.B. Jackson, Autoreturn function for a remotely piloted vehicle, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, AIAA 2002-4673, 2002
- Micropilot, MP2028 series autopilots (2008), Online, <http://micropilot.com/autopilots.htm>
- MicroPilot Inc, *MP2028g Installation and Operation* (MicroPilot Inc., Stony Mountain, MB, 2005)
- National Fuel Cell Council, Fuel cell glossary (2006), <http://www.usfcc.com/Glossary2.pdf>
- National Transportation Safety Board, NTSB Incident CHI06MA121 – full narrative (2008), online, http://www.nts.gov/ntsb/brief2.asp?ev_id=20060509X00531&ntsbno=CHI06MA121&akey=1
- M. Neale, M.J. Schultz, Current and future unmanned aircraft system control and communications datalinks, in *AIAA Infotech Aerospace Conference and Exhibit*, Rohnert Park, CA, AIAA 2007-3001, 2007

- Northrop Grumman, Unmanned systems (2008), online, http://www.is.northropgrumman.com/systems/systems_ums.html
- Office of the Secretary of Defense, Unmanned aircraft systems roadmap 2005–2030, 2005
- A. Parsch, Boeing/Insitu ScanEagle (2006), online, <http://www.designation-systems.net/dusrm/app4/scaneagle.html>
- R.A. Peters, M. Farrell, Comparison of LEO and GEO satellite systems to provide broadband services, in *21st International Communications Satellite Systems Conference and Exhibit*, Yokohama, AIAA 2003-2246, 2003
- Procerus Technologies, Kestrel autopilot system (2008a), online, <http://procerusuav.com/Downloads/DataSheets/Kestrel.2.2x.pdf>
- Procerus Technologies, *Kestrel User Guide* (Procerus Technologies, Vineyard, UT, 2008b)
- Protonex Technology Corporation, *ProCore UAV – Lightweight Propulsion* (Promotional Flyer, 2006), online, <http://www.fuelcellmarkets.com/images/articles/ProtonexProCoreUAV.pdf>
- T.M. Ravich, The integration of unmanned aerial vehicles into the national airspace. *North Dakota Law Rev.* **85**(597), 597–622 (2009)
- RCV Engines, Ltd, News release: RCV awarded engine contract for micro air vehicle (2006), online, http://www.rcvengines.com/pdf_files/pr/mav-contract.pdf
- C. Reid, M. Manzo, M. Logan, Performance characterization of a lithium-ion gel polymer battery power supply system for an unmanned aerial vehicle. Technical report NASA/TM-2004-213401 2004-01-3166, NASA, 2004
- C. Reynolds, T.A. Wilson, Detect, sense, and avoid: Regulatory gap analysis, in *FAA Center for Excellence in General Aviation Conference*, Anchorage, AK, 2008a
- C. Reynolds, T.A. Wilson, Regulatory gap analysis: Detect, sense, and avoid technologies for UAS. Technical report, Pending Release, FAA technical report, 2008b
- K. Ro, J.S. Oh, L. Dong, Lessons learned: Application of small UAV for urban highway traffic monitoring, in *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, AIAA 2007-0596, 2007
- Rolls-Royce, PLC, AE 3007 technical data (2006), online, <http://www.rolls-royce.com/defence-aerospace/products/tactical/ae3007/tech.jsp>
- H.P. Rotstein, R. Ingvalson, T. Keviczky, G.J. Balas, Fault-detection design for uninhabited aerial vehicles. *Journal of Guid. Control Dyn.* Anchorage, AK, **29**(5), 1051–1060 (2006)
- RTCA SC-2003, DO-304: Guidance materials and considerations for unmanned aircraft systems. Technical report, RTCA, Inc. Special Committee SC-203, 2007
- SAE International, Flight Director Equipment, AS-8008, 1984
- SAE International, Automatic pilots, AS-402B, 2001
- SARA, Inc, Uav acoustic collision-alert system (2008), online, http://www.sara.com/ISR/UAV_payloads/PANCAS.html
- R.S. Stansbury, M.A. Vyas, T.A. Wilson, A survey of UAS technologies for command, control, and communication (C3). *J. Robot. Intell. Syst. (JINT)* **54**(1), 61–78 (2008)
- R.S. Stansbury, W. Tanis, J. Davis, T.A. Wilson, A technology survey and regulatory gap analysis of emergency recovery and flight termination for UAS, in *AUVSI North America*, Washington, DC, 2009a
- R.S. Stansbury, W. Tanis, J. Davis, T.A. Wilson, UAS emergency recovery and flight termination: technologies and regulatory gaps, in *SAE 2009 AeroTech Congress and Exhibition*, Seattle, WA, 2009b
- R.S. Stansbury, W. Tanis, T.A. Wilson, A technology survey of emergency recovery and flight termination systems for UAS, in *Infotech@Aerospace/Unmanned Unlimited Conference*, Seattle, WA, 2009c
- R.S. Stansbury, M.A. Vyas, T.A. Wilson, A technology survey and regulatory gap analysis of command, control, and communication (C3) UAS technologies, in *IEEE/AIAA Aerospace Conference*, Big Sky, MT, 2009d
- R.C. Strain, M.T. Degarmo, J.C. Moody, A lightweight, low-cost ADS-B system for UAS applications, in *AIAA Infotech Aerospace 2007 Conference and Exhibit*, Rohnert Park, California, AIAA 2007-2780, 2007

- C. Theiss, A. Thomas, Comparison of prime movers suitable for USMC expeditionary power sources. Technical report, Oak Ridge National Laboratory (ORNL), 2000
- United States Naval Academy, Fundamentals of naval warfare systems (2008), online, <http://www.fas.org/man/dod-101/navy/docs/fun/index.html>
- B. Vaglianti, R. Hoag, M. Niculescu, *Piccolo System User's Guide* (Cloud Cap Technology, Hood River, OR, 2008)
- L.A. Walker, Flight testing the X-36 – the test pilot's perspective. Technical report, NASA contractor report no. 198058, NASA – Dryden Flight Research Center, Edwards, California, 1997
- J.S. Winstead, Transformational isr (RQ-4 GlobalHawk), in *TAAC Conference Proceedings 2009 [cd-rom]*, Albuquerque, NM, 2008

Concept of Operations of Small Unmanned Aerial Systems: Basis for Airworthiness Towards Personal Remote Sensing

95

Brandon Stark, Calvin Coopmans, and YangQuan Chen

Contents

| | | |
|------------|--|------|
| 95.1 | Introduction | 2340 |
| 95.2 | Airworthiness | 2343 |
| 95.2.1 | Aircraft Standards | 2343 |
| 95.2.2 | Ground Control Station | 2345 |
| 95.2.3 | Air Safety | 2346 |
| 95.3 | Flight Operations | 2349 |
| 95.3.1 | Operational Environment | 2349 |
| 95.3.2 | Flight Authorization | 2350 |
| 95.3.3 | Data Mission Assurance | 2351 |
| 95.4 | Operator Certification | 2351 |
| 95.4.1 | Human Factors | 2352 |
| 95.4.2 | Documentation and Logs | 2352 |
| 95.5 | An Application Scenario | 2353 |
| 95.5.1 | The Riparian Application Scenario | 2353 |
| 95.5.2 | Mission Preparation | 2353 |
| 95.5.3 | Mission Success Metrics | 2354 |
| 95.5.4 | Applications for Advanced Payload Development or Human-Automation Interaction | 2355 |
| References | | 2358 |

B. Stark (✉) • Y. Chen

Mechatronics, Embedded Systems and Automation (MESA) Lab, School of Engineering,
University of California, Merced, CA, USA

e-mail: bstark2@ucmerced.edu; yqchen@ieee.org; ychen53@ucmerced.edu

C. Coopmans

The Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State University, Logan,
UT, USA

e-mail: cal.coopmans@usu.edu

Abstract

Unmanned aerial systems (UASs) have rapidly grown into a significant application for civilian operations. However, regulations and guidelines have lagged, leading to uncertainty for domestic development. Within the USA, there has been significant pressure to develop the regulations to utilize UASs within the National Airspace System (NAS), but there remains a significant amount of discussion on UAS operation and its subsequent safe integration. The purpose of this chapter is to discuss the challenges of UAS integration into the NAS for a class of small UASs for personal remote sensing (PRS). This approach is centered on the three pillared foundation for NAS integration presented by the U.S. Department of Defense (DOD) Unmanned Aircraft System Airspace Integration Plan and focuses on the specific challenges that are unique to the PRS UAS platform. This chapter presents a concept of operations for these unmanned aircrafts (UAs) along with an application scenario example and concludes with a discussion of the future use of UASs in the NAS and how it is achievable through adequate regulations and standards.

95.1 Introduction

Civilian and commercial unmanned aerial systems (UASs) regulations have remained largely underdeveloped, despite the anticipated growth of the UAS industry in the aerospace industry (Hill 2012). In the USA in particular, while regulations and standards have been developed for military applications, there has been little progress for domestic UASs to achieve access and integration in the U.S. National Airspace System (NAS) (Kaliardos and Lyall 2013). Within the U.S., exploratory reports such as the Small Unmanned Aircraft System Aviation Rule Committee (SUAS ARC) Comprehensive Set of Recommendations for sUAS Regulatory Development, the Interim Operational Approval Guidance Policy 08-01, and the Department of Defense (DOD) Unmanned Aircraft System Airspace Integration Plan have been released over the past decade attempting to address these issues, but many restrictions and uncertainties still exist, especially for civilian operations (Barnhart et al. 2012; Stark et al. 2012a). Compounding the issue is that to many people, unmanned aircrafts (UAs) are an unknown danger, their misperception due to the high profile of military UASs (Barnhart et al. 2012). While the dangers of a UAS should not be understated, these systems have the potential to be a valuable tool for a wide range of domestic applications. In particular, small UASs are expected to achieve a significant growth over the next decade as their potentials are fully realized.

Typically, UASs have been designed with the intent of completing missions deemed as the “three D’s”: dull, dirty, or dangerous. In these missions, UAs are utilized as a replacement for manned aircrafts in an effort to make these missions safer (DOD 2011). While this effort has been effective for military applications, the concept that UASs are to be a replacement for manned aircraft is a constraint towards the development of domestic uses of UASs.



Fig. 95.1 UAS prototypes (Utah Water Research Lab)

While many people view the primary application for UASs as espionage and warfare, they have a significant potential as a remote sensing platform. Small unmanned aerial systems (SUASs) can be effectively utilized as remote sensing platforms down to a personal level (Chen et al. 2011). While it is unlikely that everyone will own a UAS in the future, many people can utilize a method for data collection. In this capacity, a personal remote sensing (PRS) UAS can be defined as a class of unique application-based small unmanned aerial systems for specific data collection and remote sensing applications, either for scientific or commercial applications (Chao and Chen 2012). An example of a successful application can be found in the AggieAir Flying Circus, a remote sensing service operation that specializes in high-resolution, multispectral aerial imagery for a variety of applications (AggieAir Flying Circus 2012). This service center, sponsored by the Utah Water Research Laboratory (UWRL), has had a great deal of success providing its unique capabilities over several sites across the western USA with its fleet of UAS (Fig. 95.1).

Remote sensing operations come from a variety of scenarios. Geo-referenced aerial imagery can be utilized for rangeland management (Bestelmeyer 2006). Landowners can monitor the growth of crops using sophisticated imaging sensors (Hunt et al. 2010; Berni et al. 2009; Jensen et al. 2009). Wildlife researchers could utilize the imagery to the nesting habits of waterfowl (Jones et al. 2009). Scientific thermal imagery can be utilized for a wide variety of agricultural applications (Sheng et al. 2010) as seen in Fig. 95.2 (Chen et al. 2012). When equipped with thermal imagery sensors, UASs may be utilized in fire-fighting applications as well (Ambrosia et al. 2003; Casbeer et al. 2005). Emergency response personnel may use a PRS UAS equipped with thermal imagery to help in search and rescue operations (Rudol and Doherty 2008). Scientific researchers may use a PRS UAS to aid in wetland management (Jensen et al. 2011). These applications are all enabled through the use of UAS in a manner that manned aircraft could not accomplish at a much lower cost and a much safer level.

However, in utilizing these platforms, a variety of challenges are presented. UA can be a great tool, but when used incorrectly or maliciously, they can become



Fig. 95.2 Thermal imagery collected by AggieAir (Chen et al. 2012)

extremely dangerous. Compounding the issue, PRS UAS platforms are often similar in size to typical RC model aircraft. While safety is an important standard for RC model flying (AMA 2011) and are regulated by the Academy of Model Aeronautics (AMA), RC aircrafts are often incorrectly sold as toys in department stores, with little to no awareness of the danger that they could pose. This misconception can affect the mindset of users of the similarly sized PRS UAS, especially for those who started as an RC hobbyist with no familiarity of the AMA or their safety standards. Do-it-yourself hobbyists have flocked towards building their own UASs with no rules or regulations to follow to ensure their safety. Without proper regulations and guidance, this can present a significant threat. A serious discussion on the future of UASs must address several issues, none of which are easily solved.

- Who should be allowed to use UASs and in what capacity?
- What can be done to ensure that UASs are being used properly?
- How can regulations be utilized to promote air safety?
- Standardized training for commercial operations?
- Is strict regulation necessary the proper solution?
- How can PRS UASs be utilized efficiently and cost effectively in the presence of regulation?

Describing and defining a concept of operations is the first step towards addressing these concerns and understanding how UASs can be utilized effectively and safely in the NAS for everyone. The concept of operations described in this chapter (Fig. 95.3) is designed to help promote the utilization of UASs for personal remote sensing in a manner congruent with integration into the NAS and addresses the unique challenges and questions that these platforms present. This preparation is designed around the three foundations for UAS airspace integration: any UAS must be certified as airworthy, flight operations must be in compliance with any regulatory guidelines, and the UAS operators must be certifiably qualified (DOD 2011). With Fig. 95.3 as the big picture in mind, UAS operations in the NAS will become routine in the near future.

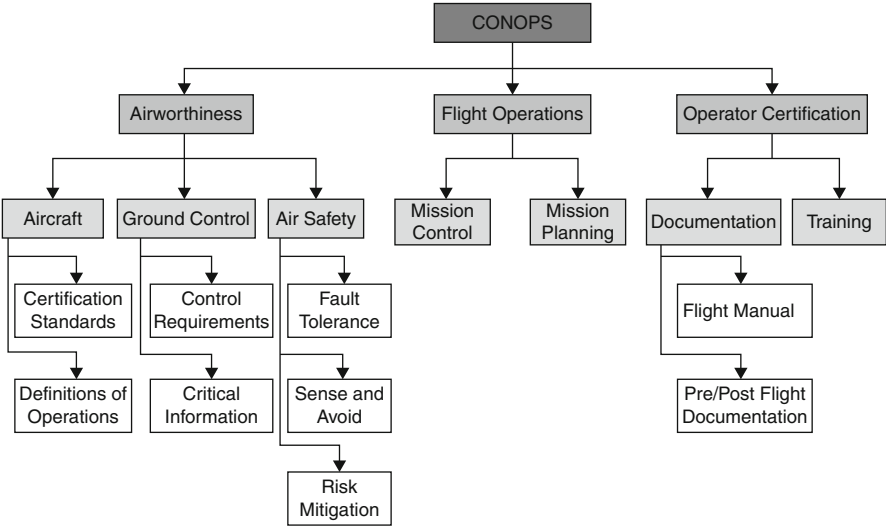


Fig. 95.3 Concept of operations for PRS SUASs (Stark et al. 2012a)

The rest of the chapter is organized as follows. The next section, Airworthiness, discusses the design criteria that are used to demonstrate airworthiness for UAS platforms. The following section, Flight Operations, introduces critical aspects for mission control and planning, with a special emphasis on the acquisition of mission data. An investigation into operator certification is presented in Sect. 95.4 towards a goal of understanding human factors and risk management in UAS systems. In Sect. 95.5, two example application scenarios are briefly examined. The final section concludes with a discussion to provide insight into the questions posed by UAS to NAS integration and comments on the future trends in UAS development for domestic use.

95.2 Airworthiness

Airworthiness is a broad concept that can be defined as containing the necessary criteria for safe and controlled flight (De Florio 2011). It brings together the major components of an airworthy system: proper aircraft standards, ground control criteria, and an emphasis on air safety. In the following section, each of these three aspects is discussed.

95.2.1 Aircraft Standards

Aircraft standards have been utilized to define the necessary requirements to promote an understanding of acceptable means of compliance with air safety requirements. Current regulations are underdeveloped, especially within the USA

While some regulations have been proposed, one area that they are still deficient on is the standards regarding the utilization of autonomous control and its effect on safety standards.

95.2.1.1 Current Regulations

In the USA, the Federal Aviation Authority has yet to release any specific regulations regarding UASs; instead current recommendations point towards the applicable sections of U.S. FAA Federal Aviation Regulations (FAR) 14 CFR Part 23 – Airworthiness Standards: Normal, Utility, Acrobatic, and Commuter Category Airplane (Federal Aviation Authority 2013). While some aspects are applicable, FAR 14 CFR 23 is deficient for many UASs, especially small UASs (SUASs) as defined in Group I (Table 95.1). An international attempt by the NATO Standard Agreement (STANAG) 4671: UAV System Airworthiness Requirement (USAR) was presented in 2010 (STANAG 4671 2010). Based on several existing regulations, including the FAR 14 CFR Part 23 and the European Aviation Safety Agency Certification Specifications – 23 (Normal, Utility, Aerobatic, and Commuter Aeroplanes), STANAG 4671 defines the basis for certification for military UASs. Recently, NATO has begun drafting of STANAG 4703 USAR Light (<150 kg) for defining the airworthiness of smaller UASs, and STANAG 4702 USAR Rotary Wing for airworthiness for rotary-wing UASs.

While the STANAG USAR regulations are focused on military applications, many of the concepts are applicable for PRS UASs. However, the smaller scale,

Table 95.1 UAS groups as classified in Department of Defense (2011)

| UAS groups | Maximum weight (lbs) | Normal operating altitude (ft) | Speed (kts) | Typical operations |
|------------|----------------------|--------------------------------|-------------|--|
| Group 1 | 0–20 | <1,200 AGL | 100 | Small portable systems for short-range surveillance and reconnaissance, within visual range |
| Group 2 | 21–55 | <3,500 AGL | <250 | Small to medium support for intelligence, surveillance, and reconnaissance in unimproved areas |
| Group 3 | <1,320 | <FL 180 | | Medium to long-range endurance for enhanced missions |
| Group 4 | >1,320 | | Any | Large UASs with extended range and endurance for beyond line of sight operations |
| Group 5 | | >FL 180 | | Includes the largest systems for specialized operations |

Table 95.2 Autonomous UAS control levels (Stark et al. 2012a)

| Level | Automation control qualifications |
|---------|--|
| Level 1 | Completely autonomous including autonomous sense and avoid (SAA) |
| Level 2 | Autonomous mission capable; UAS operator required for collision avoidance and mission re-tasking |
| Level 3 | UAS operator specifies desired aircraft attitude; control surface and throttle autonomously controlled |
| Level 4 | Remotely piloted; no augmented control |

lower margins for cost efficiency and mission applications necessitate adjustments, particularly in air safety. These standards would be beneficial in many ways but its effectiveness would be greatest in UASs that are flown regularly. The SUAS ARC, following public opinion, was explicit in exempting recreational use from following airworthiness regulations to provide the greatest latitude for model aircraft operators (SUAS ARC 2009).

For systems that are expected to fly regularly, such those flown by commercial operations, one deficient aspect of the STANAG USAR regulations is the lack of consideration towards the various levels of automation that may be utilized. A PRS UAS may have several levels of autonomous control and may be adjustable in mid-flight (Stark et al. 2012a). Mid-flight adjustments in autonomy are especially useful for mission re-tasking or for emergency contingency actions. Autopilot systems must be expected to act in a deterministic capacity in order for safety protocols or standards to be utilized effectively. A quantification of autonomous control levels is presented in Table 95.2.

95.2.2 Ground Control Station

The GCS plays a critical role in the airworthiness of a UAS. It is composed of two major aspects, the use of the GCS to aid UAS operations and the utilization of a communication standard.

95.2.2.1 Ground Control Station Standards

The GCS provides the UAS operator with a method for communication with the UA. The combination of a GCS and the onboard autopilot must be utilized to fully replace a manned pilot. Where the autopilot is deficient (by design or by automation level), the GCS must be able to provide the operator with sufficient knowledge to make informed and safe actions. Both STANAG 4671 and the SUAS ARC have suggested primary and secondary levels of information requirements. These listings are adequate for many uses, including PRS UASs, though in practice, the data that is presented is less important than how it is presented. Studies of human-automation interaction have utilized metrics such as situational awareness (Drury et al. 2006a,b),

and cognitive load (Ayaz et al. 2010; Bahner et al. 2008) to evaluate their systems (Parasuraman 2010). These metrics can be utilized to great effect in analyzing performance of a single UAS operator controlling multiple aircraft, though this application is currently recommended to be prohibited by SUAS ARC (2009).

95.2.2.2 Communication Standards

The role of communication between the ground and the UA is critical for maintaining safe operations. In typical PRS UAS operations, operations are limited to line of sight (LoS). In addition to safety reasons, this is also a result of utilizing common civilian frequencies. In the USA, unlicensed, industrial, scientific, and medical frequencies are governed by Part 18 of the Federal Communication Commission (FCC). These frequencies include 915 MHz, 2.45 GHz, and 5.8 GHz; the latter two are often significantly crowded with Wi-Fi Internet usage. These commonly used civilian frequencies are susceptible to noise that limits their effectiveness over long ranges. One counter to the overcrowding on these frequencies is to utilize frequency hopping, such as that found in Spektrum 2.4 GHz DSMX RC transmitters and receivers. By changing frequencies in a pre-agreed-upon pattern, a robust link performance can be achieved that limits inadvertent or malicious frequency jamming.

95.2.3 Air Safety

Air safety is a broad term that constitutes a significant portion of defining airworthiness. This can most effectively be accomplished through the use of assessments and risk analysis to address the most difficult aspect of integration of UASs into the NAS. STANAG 4671 proposes a Functional Hazard Assessment (FHA), congruent with existing airworthiness standards for manned aircraft that aims to identify all potential hazards and mitigating actions. In a typical procedure, a preliminary system safety analysis is performed by examining the individual operational phases: planning, staging, launching, flying, and landing. At each phase, the functions of each of the active components are identified and potential fault conditions are listed. Utilizing a common cause analysis metrics, the severity and probability of each of the fault conditions can be quantified (STANAG 4671 2010).

In this section, ideas and procedures are presented to assist in the development of an FHA and methods to improve air safety through fault-tolerant systems, sense-and-avoid systems, and risk mitigation strategies.

95.2.3.1 Fault Tolerance

Fault tolerance in SUASs such as PRS UASs is an important element for air safety. While full-sized aircraft can often utilize several layers of redundancy, smaller aircraft are often limited by cost or weight restrictions. Instead, extra emphasis can be placed on developing fault-tolerant systems in an effort to achieve an acceptable level of safety.

Faults are typically described as abnormal behavior or a complete loss of a component within a flight control system (Ducard 2009). These faults can be separated into two major categories: hardware faults and electronic faults. Hardware faults occur on the aerodynamic and structural components and range from mechanical failures to electromechanical failures, such as control surface ineffectiveness. Electronic faults are typically related to avionics or control systems, such as sensor failures, autopilot failures, or communication failures. It is important to note that not all faults need to be handled singularly by fault tolerance. The more effective strategy is to provide multiple levels of air safety by a combination of fault tolerance and risk mitigation strategies.

95.2.3.2 Sense and Avoid

The most challenging aspect for a UAS to meet is typically referred to as “see and avoid” for traditional manned aircraft. For a UAS, “sense and avoid” is a more appropriate phrase, but the concept remains the same. The ability to sense flight obstacles, including other aircraft and ground hazards, and perform evasive maneuvers is a critical element for UAS integration into the NAS (ASUAPO 2008; SUAS ARC 2009; DOD 2011). Sense-and-avoid systems are typically separated into two categories: airborne Sense and Avoid (ABSAA) and ground-based sense and avoid (GBSAA). In the USA, for both of these types of systems, adequate performance has typically been described as being able to detect a potential collision such that any avoidance maneuver will prevent the aircraft from being less than 500 ft from the object (DOD 2011). This is a significant technical challenge, especially for airborne systems.

However, for PRS UASs, the typical mission is accomplished within direct LoS in remote locations. Protocol for direct LoS operations may include the use of an observer that actively scans for potential hazards (DOD 2011). Unlike larger UA platforms, PRS UAs are typically small, slow but maneuverable aircraft that are capable of effective avoidance maneuvers, especially in the presence of other air traffic. The combination of LoS operations, remote locations, and effective avoidance maneuvers mitigates most other airborne threats and provides a sufficient ability to prevent collisions.

Beyond LoS operations pose a significantly greater challenge for SAA systems. In addition to the distinction between ABSAA and GBSAA, there is a distinction between cooperative and noncooperative SAA technologies. Cooperative SAA such as the traffic collision avoidance System (TCAS) and automatic dependent surveillance -broadcast (ADS-B) rely on the installation of extra equipment on the aircraft that allows for the transmission or broadcast of the UA’s information to other aircraft in the area (Angelov 2012). These technologies have a significant history of success for manned aircraft and are the superior choice for air traffic control. However, their effectiveness is limited to other cooperative aircraft. In the areas that PRS UASs are likely to be conducting missions, such as remote farmland, air traffic is typically not a great threat, but rather ground hazards such as satellite towers or noncooperative air traffic such as ultralight aircraft pose the greatest threat.

Fig. 95.4 AggieAir PRS UA with high-visibility reflective markers



Noncooperative systems, either airborne or ground-borne-based systems, typically rely on sensing technologies such as optical cameras, radar, or laser scanning. These systems have significant technical challenges ahead for their utilization to become common. A recent study of a pilot's ability to detect other air traffic suggests that the minimum resolution for an optical solution is $1,920 \times 1,080$ HD video and is only effective at ranges under 1 mile (Kephart and Braasch 2010). This aspect of SAA systems is an active area of current research (Angelov 2012).

While much work remains to be accomplished regarding SAA technology, more work can be done towards UAS observability. As a typically noncooperative aircraft, a PRS UAS must be designed to ensure an adequate level of observability to ground crew and other aircraft. This includes conforming to a high-visibility high-contrast color scheme as described in other specifications (MIL-STD-2161B 2008). While no standard has been developed specifically for UAS, bright colors such as yellow, orange, and neon green are effective when contrasted with darker blues, blacks, and purples. Illuminating lights such as those suggested in STANAG 4671 are also effective but not sufficient singularly. Reflective markers such as those depicted in Fig. 95.4 can be utilized effectively.

95.2.3.3 Risk Mitigation

Risk mitigation is a broad term that encompasses fail-safe protocols and off-nominal procedures to be performed when appropriate. Many of these protocols are designed to mitigate as much damage as possible in the event of an unforeseen event, considering first the safety of the operators, ground crew, and general public and secondly the recoverability of the UA. An incomplete listing of risk mitigation protocols is listed in Table 95.3.

Table 95.3 Risk mitigation protocols

| Fault | Protocols |
|------------------------------------|---|
| Loss of thrust | UAS operator takes manual control and attempts an emergency landing |
| Loss of GPS | UAS operator takes manual control and attempts an emergency landing |
| Loss of attitude | UAS operator takes manual control and attempts an emergency landing |
| Control surface failure | UAS operator takes manual control and attempts an emergency landing |
| Loss of datalink comm. | Autopilot may enter datalink communication fail-safe and return to a predesignated location, or UAS operator takes manual control and attempts an emergency landing |
| Structural damage | UAS operator takes manual control and attempts an emergency landing |
| Emergency kill situation | UAS operator deactivates all propulsion elements |
| Fail-safe control surface protocol | UAS operator activates control surface lock to place the UAV in a controlled trajectory termination immediately |

95.3 Flight Operations

A PRS UAS is primarily a civilian system. As these systems mature, they will be actively utilized in a large variety of applications. However, unless regulatory standards are fully defined and implemented, they pose significant risk. One of the most important requirements that must be introduced is a clear definition of the flight operations of missions. In this section, the defining characteristics of a flight operation, specifically the operational environment and mission planning, are presented.

95.3.1 Operational Environment

One of the distinct advantages of PRS UAS is the relatively short setup time required to organize for a mission. In an ideal setting, a UAS operator could set up the system, launch and complete a mission, and pack up in a couple of hours. In contrast, a manned aircraft conducting the same mission might require several days of planning and setup. Mission planning for a PRS UAS is still a critical element, ensuring that mission objectives can be accomplished safely.

In many typical PRS UASs, no infrastructure setup is necessary or is sufficiently portable and transportable (Fig. 95.5). This advantage allows missions to be conducted wherever ground access is available, allowing missions to be accomplished without the use of an established airport or runway. Air traffic is typically minimal

Fig. 95.5 Portable UAS launch equipment



and the entire operation can be accomplished within line of sight (LoS), and the terminal area (launch and recovery operations) occupies the same space as the operational area. In addition, most operations can be completely accomplished in low traffic class G airspace (DOD 2011), minimizing potential air traffic situations.

95.3.2 Flight Authorization

Operation of a PRS UAS must also conform to existing regulations on aircrafts. Few options in the USA are currently available for FAA authorization. The most common option is for UAS operators to apply for a Certificate of Waiver or Authorization (COA) that permits the use of a specific or class of UA with clearly defined operations. However, this is currently applicable for UA designated as public aircraft, a designation for aircraft used only for the U.S. Government or U.S. governmental functions such as national defense, law enforcement, search and rescue, and aeronautical, biological, or geological resource management (ASUAPO 2008).

However, there is plenty of confusion of the topic of UAS operation within the U.S. Currently, there are no official FAR that address UAS operations. However, the FAR has published several related documentation, including standards that may eventually become regulatory rules but are currently recommendations. One source of confusion on operating authorization stems from Advisory Circular (AC) 91-57, published in 1981. This circular encouraged voluntary compliance for model aircraft standards presented. Item 3.c is the most commonly referred item on this circular, stating that model aircraft are not to be flown higher than 400 ft. AGL (AC 91-57 1981).

In 2005, the FAA issued the policy statement AFS-400 UAS Policy 05-01 to provide guidance to determine if UAS may be allowed to conduct flight operations in the U.S. NAS (AFS-400 2005). This policy was not designed to be a regulatory,

though it has stood as the first attempt to standardize domestic UAS operations and developed the guidelines used for evaluating each COA application (Barnhart et al. 2012). In regard to the low-flying UAS operations that PRS UAs complete, item 6.13 is significantly applicable. This item lists that any UAs that comply with the guidance in AC 91-57 are considered model aircraft and are not evaluated by the UA criteria described in AFS-400 UAS Policy 05-01 (AFS-400 2005). This however was intended not to sanction civil operations but to continue the nonregulation of model aircraft and recreational use of the NAS (Barnhart et al. 2012).

In 2008, the FAA published the Interim Operational Approval Guidance 08-01 (IOAG 08-01) to replace AFS-400 UAS Policy 05-01. This document provided provisions for civil UAS operation through the use of a Special Airworthiness Certificate in the experimental category as the only certification available to civil operators. This document lists that all applicable regulations of 14 CFR parts 61 and 91 must be observed for UA (ASUPO 2008).

The SUAS ARC published in 2009 provides additional set of recommendations for SUASs (SUAS ARC 2009). This is largely congruent with the specifications described in the IOAG 08-01 with special consideration for the needs of SUAS such as PRS UASs. However, the recommendations listed are not regulatory or standards and are not meant to supersede any previous standards such as IOAG 08-01.

Future documentation and regulations have been promised, but their likely contributions would be similar to the standards and recommendations already in place.

95.3.3 Data Mission Assurance

While many UAS missions are defined by payload implementation, the data acquired is emphasized in a PRS UAS (Coopmans et al. 2012). Calibration, setup, and payload reliability are all aspects that dictate mission parameters. The goal of a PRS UAS is for maximum data quality and this plays a major role in mission design. The mission for a PRS UAS is designed around the data to be sampled rather than when it is convenient to fly. Weather conditions may not always be favorable, placing additional burden on preflight mission risk analyses.

95.4 Operator Certification

The UAS ground crew is a significant aspect of the unmanned aerial system. For UAS integration into the NAS, human factors must be considered (Kaliardos and Lyall 2013). Within the closed-loop UAS system, the decision making and the higher-level control of the system relies on the human element for system resiliency (Stark et al. 2012b). Utilizing the study of human factors in UASs, the human aspects of control, use, and safety operations can be effectively considered during the design of human training, record keeping, and organizational factors that make up the administrative part of a UAS.

95.4.1 Human Factors

A report by the FAA from December 2004 (Williams 2004) indicates the multitude of human factors in several military UASs. In this study, poor training and unfamiliarity of the new technology was described as the major factors in a significant number of UAS incidents. In order to quantify and study the various human factors involved in general aviation accidents, Shappell and Wiegmann presented a comprehensive human factors analysis and classification system (Shappell and Weigmann 2000). In this system, four levels of failure are described: Unsafe acts, preconditions for unsafe acts, unsafe supervision, and organizational influences.

Using HFACS, a similar report on UAS accidents provides a breakdown of the more common human factors for failures (Manning et al. 2004). Organizational influences were present in 14 % of accidents, while decision errors and unsafe supervision problems were present in 11 % of the accidents, respectively. This report also recommends the use of HFACS as a supplement to accident investigations as it provides valuable statistical data on the causes of human errors that help develop training programs. A later report offered similar conclusions (Glussich and Histon 2010), stressing the need for better training strategies as well as the need for more study.

Implementation of the organizational factors plays a significant role in reducing human-related errors (Shappell and Weigmann 2000). Constant repetition of procedures encourages commitment and aid the cognitive load of the humans in the loop, reducing the probability of errors (Bahner et al. 2008).

Utilizing these methods can lead to improving UAS ground crew training (Stark et al. 2012b). While the specific training hours for each skill set depends on the design and complexity of the UAS, the training program can be broken down into five programs: Basic UAS Operator Certification, UAS Aircraft Crew Certification, Mission Certification, Avionic Systems Certification, and Payload System Certification.

95.4.2 Documentation and Logs

An important step for the continued performance of the UAS is proper record keeping. This includes the continued implementation of pre-/postflight inspections, detailed flight logs, and flight reconstruction data collection. This ensures the airworthiness of the aircraft, the continued training of UAS operators, and the reduced human-caused failures.

Preflight analysis documentation and pre-mission documentation account for a high level of documentation but are critical aspects to maintain airworthiness. These documentations include but are not limited to flight-readiness reviews, pre-mission risk analysis, predeparture checklists, preflight checklists, and detailed mission procedures.

A preflight risk analysis is a common part of pre-mission operations. It is a part of military and general aviation procedures and its use for UASs is a natural

progression. This use of quantifiable metrics to analyze fly or no fly conditions removes ambiguity and promotes efficiency and safety (Barnhart et al. 2012).

Like traditional aircraft, the working parts of airframes have maximum flight hours. If these hours are exceeded, failure of these parts is likely. Logging of flight time allows maintenance schedules to be implemented for PRS-level airframes, increasing the safety for everyone in the airspace. RC aircraft typically do not list expected part lifetimes, making it difficult to gauge typical usage life. Thorough testing and verification is recommended for RC aircraft that have been converted towards regular UAS operations. Encouraging operators to log hours is a useful tool to promote a standardized proficiency in the UAS and to verify their experience.

Performance metrics for PRS aircraft designs allow failures to be documented and quantified, allowing designers to improve aircraft designs based on data gathered in the field. Other aircraft subsystems such as payloads, launchers, and ground support equipment can also be addressed in this fashion, enabling a trail of accountability for performance and safety.

95.5 An Application Scenario

There are countless numbers of applications where personal remote sensing enabled by UASs can provide significant benefits over traditional satellite imagery or manned aircraft. While other platforms can be utilized, small UAs can provide high-resolution, multispectral imagery at a higher rate due to their relative low-cost and reduced logistical requirements.

95.5.1 The Riparian Application Scenario

In this application scenario based on ongoing work done by the AggieAir Service Center, multispectral imagery is collected by a PRS UAS for riparian applications (Jensen et al. 2011). The goal of the proposed mission is to provide information that can be utilized to study the fish habitat within the river. Since the river is constantly changing, aerial imagery is to be collected at several times throughout the year. The mapping of the various flow conditions will be utilized to study the health of the river and its effect on the fish habitat. The important aspects of the data collection are that the river channel must be clearly identifiable with accurate georectification to provide sufficient resolution. The quality of the imagery and the timing of the imagery are both equally important for data analysis. Example raw imagery of this type of scenario is depicted in Fig. 95.6.

95.5.2 Mission Preparation

Given the river's remote location, large plots of land are available for launch and landing, as well as any emergency procedures. Neither air traffic nor local population is a major concern. In this particular mission, a target goal is <25 cm

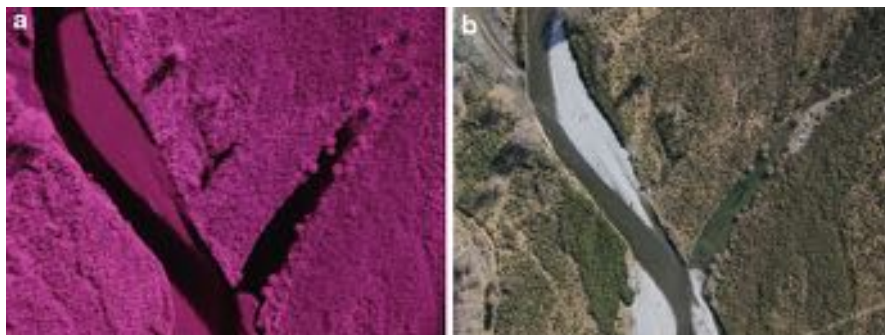


Fig. 95.6 Example raw riparian imagery (a) NIR imagery, (b) RGB imagery

resolution, but the area coverage is large. The target flight altitude is recommended at 1,000 ft. This flight altitude is allowed by the COA issued to the aircraft at this location, but a notice to the air traffic control is issued as well. The preflight simulation estimates the total flight time expected at 30 min, well within the maximum flight time of the aircraft and the allowed flight time at this altitude as listed by the COA. The proposed altitude is within LOS, but binoculars are provided as a precaution. Given the conditions of the aircraft and the mission plan, the pre-mission risk analysis is largely dependent on the atmospheric conditions at the time of the proposed flight. On the day of the proposed flight, the weather conditions are favorable, ensuring that the mission is low risk and would provide the highest quality data conditions.

The flight path for data collection is a critical element. In order to achieve maximum data collection, the UA will follow the path of the river, and then sweep back and forth perpendicular to the river as it returns home. Preflight simulation estimates imagery coverage at 400 %, a sufficient level. Previous flights have indicated that turbulent wind at this altitude negatively effects data collection and thus a higher imagery coverage is necessary to ensure coverage.

In this regard, the understanding of the performance of the payload has affected the mission parameters. The collection of NIR imagery for the accurate mapping of the river necessitates that the mission occurs at the sun's highest apogee to minimize shadows that mask river detection.

95.5.3 Mission Success Metrics

The success metrics of this mission can be defined as the optimality of data collection with respect to flight time and the number of flights necessary to collect adequate information. The quality of the imagery is not known to the ground crew until the end of the mission. If the data is not usable, the mission must be repeated, an inconvenient prospect especially in the presence of time constraints (maximum sun apogee, rapidly changing flow conditions).

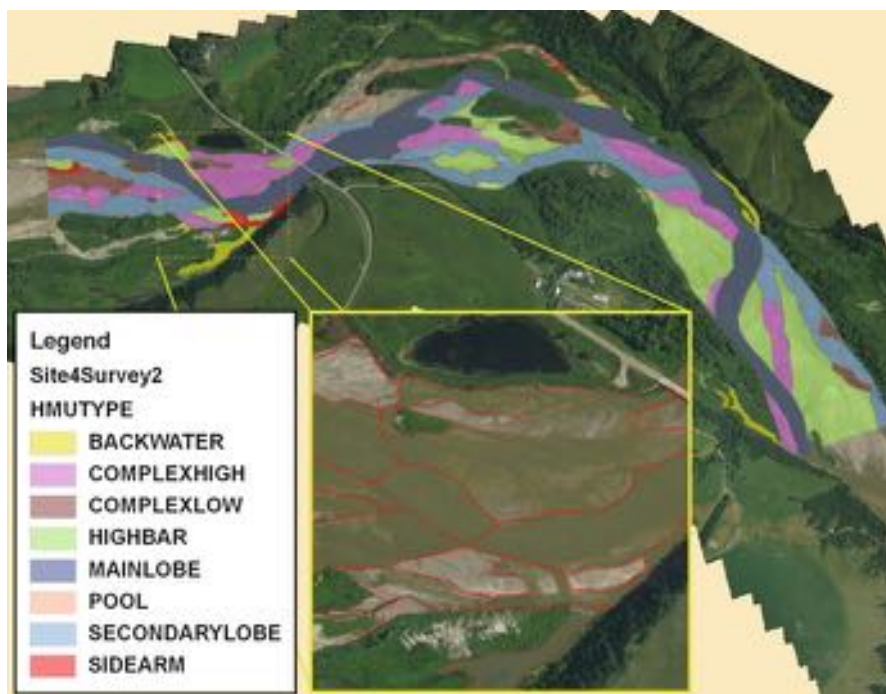


Fig. 95.7 Annotated mosaic a few days after flight with annotation by Rushing Rivers Institute (AggieAir Flying Circus 2012)

An example of the final imagery created through this application is seen in Fig. 95.7. In this image, with final annotations provided by Rushing Rivers Institute, the various flow levels throughout the year are visible in an understandable format that benefits riparian studies.

95.5.4 Applications for Advanced Payload Development or Human-Automation Interaction

Future work of applications of advanced cognitive payloads or human-interface devices can be utilized in several capacities. In future implementations, they may be utilized to improve UAS operation safety or to improve performance by adding additional autonomy to the UA. The development of these advances must consider their effect on safety, but at the same time, their development not be impeded through needless regulation.

The addition of a cognitive payload with an emphasis on data mission assurance can provide invaluable performance to PRS UASs. Utilization of payload management and resiliency aids in the development of reliable systems as well as reduce downtime due to faulty components. Low-cost systems often utilize commercial-of-the-shelf (COTS) equipment which may be at a lower quality standard and are more

prone to failures. Such resilient payload architecture can be utilized to monitor the sensors attached to the payload and initiate recovery operations or alert the ground crew (Coopmans 2012).

In addition, due to the offline nature of the high-resolution image collection, real-time processing such as data quality assurance or real-time autonomous target identification is not possible. An advanced cognitive payload that can process images for autonomous target identification (people, hazards, or areas of interest) or for image quality for data collection can be utilized to reduce the average flight time necessary for a mission. In-flight mission re-tasking could be accomplished with real-time information for an improve data collection (NSF 2011). A proposed system may be able to suggest areas that need to be collected again or may autonomously add these areas to its ongoing flight path. The prospect of adding more autonomy to flight planning would necessitate a further discussion on the safety factors, including ensuring its deterministic behavior and mitigation protocols for autonomous flight planning failures.

Human-automation interaction is a significant aspect that can be improved through the use of more cognitive payloads and advanced human-automation devices such as immersive goggles and more intuitive human-machine interfaces (HMI). These advances may play a significant role in improving the safe operation of UASs by utilizing more intuitive actions and information processing.

Conclusion

The concept of operations developed in this chapter was designed around the three foundations for UAS integration in the NAS described by Department of Defense (2011). For small UASs for PRS applications, these foundations are critical to implement because these are the UAs that are most likely to be the most commonly utilized. While there are many applications for UAs, the larger UAs are cost-prohibitive to the majority who would utilize their advantages. The smaller PRS UASs are presented as a cheaper and more efficient alternative for manned aircraft and larger UASs, especially where large spatial coverage is not necessary. However, the issue arises in answering the questions posed at the beginning in the chapter.

The most pertinent question addresses the users and in what capacity the users should be allowed to operate. Presumably, the most common users of these small UASs would fall into four categories: recreational users, personal usage for regular operations, commercial operations, and government/education agencies for development and scientific operations. Each of these groups have different purposes and reasons for flying UA within the NAS, but all of them have safety considerations that need to be addressed.

Recreational UAS operations may appear as the most dangerous due to the complexity of the systems; however, recreational use of the NAS with remotely controlled model aircraft has had a long history of success without strict regulation. As mentioned previously, the SUAS ARC recommended that this

nonregulation continue, with the AMA organization providing the self-regulation to maintain safety standards. This arrangement has encouraged innovation and development of the model aircraft industry and it would likely continue with the development with recreational UAs. The important aspect in this arrangement is the self-regulation of the AMA to enforce safety codes, especially flight altitude and flight operations that are already in place (AMA 2011).

Government agencies, educational institutions, and commercial operators are all groups that would look to utilize UASs on a regular basis and in operations that would exceed the recommendations of the AMA safety code. Regular usage at this level should be regulated in the methods described in this chapter's concept of operations documentation. Systems in use on a regular basis should be subject to airworthiness certification and regulation, enforced through the local aviation authority. Safety protocols and operator training should be an important part of UAS certification for these groups. This is to create a line of accountability for their actions that have a higher risk of danger or damage to the surroundings and local populace. Development of advanced technology such as SAA or advanced cognitive payload or control would not be impacted by this level of regulation but rather be encouraged by the importance of safety protocols.

The most challenging group to apply regulations towards is the middle ground between recreational use and regular personal use. While the standard safety codes regulated by the AMA may be sufficient for most users, many will want to extend beyond to higher altitudes and farther distances. This is already a contentious area for many in the RC hobbyist community in the frame of reference of first-person view (FPV) flying. In FPV flying, users will utilize long-range wireless cameras and fly their aircraft through a vision system, often beyond the safety code of the AMA. Discussions with AMA members and non-AMA members on the topic of FPV flying will lead to large variety of misinformation on the current regulations on FPV flying and F.C.C. regulations regarding video transmission. This is not an indication of poor management of the AMA but rather of an indication of the ever-changing technology and the slow response of new regulations and safety protocols. It is evident that a new method for the regulation of advanced technology is necessary.

However, there exists a close parallel to UAS operation in high-power rocketry (HPR), currently governed by the independent agencies Tripoli Rocketry Association (TRA) and the National Association of Rocketry in the USA. These agencies have special certifications for users that allow the use of their facilities for high-power rocketry. They are designed to provide a measure of the user's competence to avoid gross violations of good practice and safe operation (NAR 2012). Certification is only available to members of good standing of these groups and for higher levels of certification, an examination is required (NAR 2012). The rules and regulations for HPR is a combination of FAA regulations, the National Fire Protection Association (NFPA) regulations, and the NAR's and TRA's safety codes. The success of HPR suggests that SUAS operations, especially those in use for remote sensing applications, should follow a similar model to promote safety and ensure education among its users.

The future of UAS operations for PRS is bright and its implementation will open up a significant number of new avenues for research and data collection. While many see domestic UAS operations as a detriment to society (Paul 2012) or rife with privacy issues (Calo 2011), their immense potential as a valuable tool for water management, precision agriculture, and search and rescue operations, just to name a few, outweighs their potential downside with the proper rules and regulations in place. Management of UAS operations in the NAS is not an easy topic to address, but incrementally UAS integration into the NAS will become a reality.

Acknowledgments This project is supported in part by Utah Water Research Laboratory MLF 2006–2012 and by NASA UAS2NAS program (Grant # NNX11AO77A). Part of this work was also supported by NSF Robust Intelligence, Collaborative Research, Information Technology Research Program (Award # 1138632).

References

- Academy of Model Aeronautics (AMA), National Model Aircraft Safety Code, 2011
- AggieAir Flying Circus (2012), <http://www.aggieair.usu.edu>. Accessed 28 Sept 2012
- V.G. Ambrosia, S.S. Wegener, D.V. Sullican, S.W. Buechel, S.E. Dunagan, J.A. Brass, J. Stoneburner, S.M. Schoenung, Demonstrating UAV-acquired real-time thermal data over fires. *Photogramm. Eng. Remote Sens.* **69**(4), 391–402 (2003)
- P. Angelov, *Sense and Avoid in UAS: Research and Applications* (Wiley, Hoboken, 2012)
- Aviation Safety Unmanned Aircraft Program Office AIR-160 (ASUAPO), Interim Operational Approval Guidance 08-01 Unmanned Aircraft Systems Operations in the U.S. National Airspace System, Washington, D.C., 2008
- H. Ayaz, B. Willems, S. Bunce, P. Shewokis, K. Izzetoglu, A. Deshmukh, B. Onaral, Cognitive workload assessment of air traffic controllers using optical brain imaging sensors, in *Advances in Understanding Human Performance: Neuroergonomics, Human Factors Design and Special Populations*, ed. by T. Marek, W. Karwowski, V. Rice (CRC/Taylor & Francis Group, Boca Raton, 2010), pp. 21–32
- J.E. Bahner, A.D. Huper, D. Manzey, Misuse of automation decision aids: complacency, automation bias and the impact of training experience. *Int. J. Hum.-Comput. Stud.* **66**, 688–699 (2008)
- R.K. Barnhart, S.B. Hottman, D.M. Marshall, E. Shappee, *Introduction to Unmanned Aircraft Systems* (CRC, Boca Raton, 2012)
- J.A.J. Berni, P.J. Zarco-Tejada, L. Suarez, E. Fereres, Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *IEEE Trans. Geosci. Remote Sens.* **47**(3), 722–738 (2009)
- B.T. Bestelmeyer, Threshold concepts and their use in rangeland management and restoration: the good, the bad and the insidious. *Restor. Ecol.* **14**(3), 325–329 (2006)
- M.R. Calo, The Drone as privacy catalyst. *Stanf. Law Rev. Online* **64**, 29–33 (2011)
- D.W. Casbeer, D.B. Kingston, R.W. Beard, T.W. McLain, Cooperative forest fire surveillance using a team of small unmanned air vehicles. *Int. J. Syst. Sci.* **37**(6), 351–360 (2005)
- H. Chao, Y.Q. Chen, *Remote Sensing and Actuation Using Unmanned Vehicles* (Wiley, Hoboken, 2012)
- Y.Q. Chen, D.L. Dee, A. Jensen, *Multi-UAV Based Multi-Spectrum Collaborative Personal Remote Sensing: Concepts, Platform & Applications, Invited Tutorial at 2011 ICUAS*, Denver, CO, 2011

- Y.Q. Chen, A. Jensen, G. Strahan, C. Coopmans, *Low-Cost UAV-Based Precision Thermal Infrared (TIR) Mapping – A New Personal Remote Sensing Capability: UAV Platform, TIR Payload, In-Flight Calibration and Applications, Invited Tutorial at 2012 ICUAS*, Philadelphia, PA, 2012
- C. Coopmans, B. Stark, C. Coffin, A payload verification and management framework for small UAV-based personal remote sensing systems, in *Proceedings of 2012 5th International Symposium on Resilient Control Systems (ISRCS)*, SLC, UT, USA, 2012, pp 184–189
- F. De Florio, *Airworthiness*, 2nd edn. (Elsevier, Burlington, 2011)
- Department of Defense, *MIL-STD-2161B: paint Schemes and Exterior Markings for U.S. Navy and Marine Corps Aircraft*, Washington, D.C., 2008
- Department of Defense, *Unmanned Aircraft System Airspace Integration Plan*, Washington, D.C., 2011
- J.L. Drury L. Riek, N. Rackliffe, A decomposition of UAV-related situation awareness, in *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, Salt Lake City, 2006a, pp. 88–94
- J.L. Drury, J. Richer, N. Rackliffe, M.A. Goodrich, Comparing situation awareness for two unmanned aerial vehicle human interface approaches, in *Proceeding of IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, Gaithersburg, MD, 2006b
- G.J.J. Ducard, *Fault-Tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles* (Springer, London, 2009)
- Federal Aviation Administration, *AFS-400 UAS Policy 05-01: Unmanned Aircraft Systems Operations in the U.S. National Airspace System – Interim Operational Approval Guidance*, Washington, D.C., 2005
- Federal Aviation Administration, AC 91-57, Washington, D.C., 1981
- Federal Aviation Authority, *Federal Aviation Regulations Title 14 CFR Part 23 – Airworthiness Standards: Normal, Utility, Acrobatic and Commuter Airplanes*, Washington, D.C., 2013
- D. Glussich, J. Histon, Human/automation interaction accidents: implications for UAS operations, in *Proceedings of 2010 IEEE/AIAA 29th Digital Avionics Systems Conference (DASC)*, Moffett Field, CA, 2010, pp. 4.A.3-1–4.A.3.11
- K. Hill, *Congress Welcomes the Drones*. (Forbes, 2012), <http://www.forbes.com/sites/kashmirhill/2012/02/07/congress-welcomes-the-drones/>. Accessed 22 Feb 2012
- E.R. Hunt Jr., W.D. Hively, S.J. Fujikawa, D.S. Linden, C.S.T. Daughtry, G.W. McCarty, Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sens.* **2**, 290–305 (2010)
- A. Jensen, Y.Q. Chen, M. McKee, T. Hardy, S.L. Barfuss, AggieAir – a low-cost autonomous multispectral remote sensing platform: new developments and applications, in *Proceedings of 2009 IEEE International Geoscience and Remote Sensing Symposium, IV*, Cape Town, South Africa, 2009, pp. 995–998
- A. Jensen, T. Hardy, M. McKee, Y.Q. Chen, Using a multispectral autonomous unmanned aerial remote sensing platform (AggieAir) for Riparian and Wetlands applications, in *Proceedings of 2011 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Vancouver, Canada, 2011, pp. 3413–3416
- G.P. Jones, L.G. Pearlstine, H.F. Percival, An assessment of small unmanned aerial vehicles for wildlife research. *Wildl. Soc. Bull.* **34**(3), 750–758 (2009)
- B. Kaliardos, B. Lyall, Human factors of unmanned aircraft system integration in the national airspace system, in *Handbook of Unmanned Aerial Vehicles*. (Springer Science+Business Media, Dordrecht, Netherlands, 2013)
- R.J. Kephart, M.S. Braasch, See-and-avoid comparison of performance in manned and remotely piloted aircraft. *IEEE Aerosp. Electron. Syst. Mag.* **25**(5), 36–42 (2010)
- S.D. Manning, C.E. Rash, P.A. LeDuc, R.K. Noback, J. McKeon, *The Role of Human Causal Factors in U.S. Army Unmanned Aerial Vehicle Accidents*, U.S. Army Medical Research and Material Command, 2004
- National Association of Rocketry (NAR), High Power Rocket Safety Code, Marion, IA, 2012

- NSF Award 1138632, <http://nsf.gov/awardsearch/showAward.do?AwardNumber=1138632> *Low Cost Personal Remote Sensing for Cognitive Disaster Assessment with Enhanced Human-Machine Interface* (2011–2012. PI: YangQuan Chen)
- R. Paul, *Don't Let Drones Invade our Privacy*, CNN. <http://www.cnn.com/2012/06/14/opinion/rand-paul-drones/index.html>. Accessed 28 Sept 2012
- R. Parasuraman, Designing automation for human use: empirical studies and quantitative models. *Ergonomics* **43**(7), 931–951 (2010)
- P. Rudol, P. Doherty, Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery, in *Proceedings of 2008 IEEE Aerospace Conference*, Big Sky, MT, 2008, pp. 1–8
- S.A. Shappell, D.A. Weigmann, *The Human Factors Analysis and Classification System – HFACS*, Office of Aviation Medicine, 2000
- H. Sheng, H. Chao, C. Coopmans, J. Han, M. McKee, Y.Q. Chen, Low-cost UAV-based thermal infrared remote sensing: platform, calibration and applications, in *Proceedings of Mechatronics and Embedded Systems and Applications (MESA) 2010 IEEE/ASME International Conference on*, 2010, pp. 38–43
- Small Unmanned Aircraft System Aviation Rulemaking Committee (SUAS ARC), *Comprehensive Set of Recommendations for sUAS Regulatory Development*, 2009
- STANAG 4671: Unmanned Aerial Vehicles System Airworthiness Requirements, Brussels, Belgium, 2010
- B. Stark, C. Coopmans, Y.Q. Chen, Concept of operations for personal remote sensing unmanned aerial systems. *J. Intell. Robot. Syst.* 2013 **69**(1–4), 5–20 (2012a). doi:10.1007/s10846-012-9710-9
- B. Stark, C. Coopmans, Y.Q. Chen, A framework for analyzing human factors in unmanned aerial systems, in *Proceedings of 2012 5th International Symposium on Resilient Control Systems (ISRCS)*, Salt Lake City, UT, 2012b, pp. 13–18
- K.W. Williams, *A Summary of Unmanned Aircraft Accident/Incident Data: Human Factors Implications*, Office of Aerospace Medicine, 2004

Andrew D. Zeitlin

Contents

96.1 Introduction 2362

96.2 Certification Process 2363

 96.2.1 Project Specific Certification Plan 2363

96.3 Standards 2368

 96.3.1 Requirements for Functions and Subfunctions 2370

 96.3.2 Subfunction Considerations 2372

 96.3.3 Architecture 2374

 96.3.4 Communications 2375

 96.3.5 Algorithms 2375

 96.3.6 Validating Requirements 2376

 96.3.7 Verification Tests 2378

 96.3.8 Software Standard 2379

96.4 Conclusion 2380

References 2380

Abstract

Unmanned aircraft will require a sense and avoid (SAA) capability to mitigate for the lack of a remote pilot’s ability to “see and avoid” other aircraft, which is an operational requirement. Regulators certify the airworthiness of aircraft, assuring their compliance with applicable regulations. Approvals of SAA implementations constitute part of that process. Each applicant will need to develop a Project Specific Certification Plan in close coordination with the regulator. Various tools for safety analysis and configuration control support this effort. Specification of the certification basis for SAA will need close attention, since the system addresses functions traditionally allocated to an onboard pilot. Algorithms and software will receive scrutiny due to their safety roles.

A.D. Zeitlin
The MITRE Corporation, Mc Lean, VA, USA
e-mail: azeitlin@mitre.org

Standards are developed to capture common requirements in support of systems that may use some degree of unique design innovation. A standard should suggest means of demonstrating compliance to its requirements, and so doing will greatly simplify the burden for both applicant and regulator. The closest existing standard to SAA is that for the Traffic Alert and Collision Avoidance System II (TCAS II), but SAA includes the additional function of self-separation. SAA can be implemented with various technologies in a variety of architectures.

SAA should provide two basic functions, each performing prescribed sub-functions. Considerable modeling and simulation will be required to validate system performance and support the safety case. The regulator may offer guidance to applicants through publication of an Advisory Circular.

96.1 Introduction

Airspace regulators approve aircraft and devices for flight in accordance with their responsibility to enforce regulatory requirements. Underlying those requirements is the principle of ensuring public safety. Today, the International Civil Aviation Organization (ICAO) requires airborne collision avoidance for certain manned aircraft (ICAO 2007, 2008) and describes the generic collision avoidance function as one layer in the overall scheme of safety provision (ICAO 2005). However, ICAO cautions that its presently mandated Airborne Collision Avoidance System (TCAS II in the U.S.) is not an appropriate solution for Unmanned Aircraft Systems (UAS) (ICAO 2006) and the same conclusion was reached by subsequent FAA analysis (FAA 2011). It has become widely accepted since UAS lack the ability for the remote pilot to comply with the operational regulation to see-and-avoid, that instead, a Sense and Avoid (SAA) system that includes collision avoidance needs to be developed for unmanned aircraft.

The UAS community generally recognizes an SAA function, also called Detect and Avoid (DAA), as necessary for obtaining routine access to non-segregated civil airspace. Some exceptions may be granted, such as where safety can be demonstrated by alternate means (e.g., using a manned chase aircraft, a ground observer, or relying on ATC separation or other procedures in special circumstances); or for very small UAS operations in visual range of the pilot on the ground. The U.S. Federal Aviation Administration (FAA) currently defines operational limits and mitigations when granting Certificates of Authorization (COA) for public and some civil operators and when issuing experimental certificates. These approaches are not considered feasible for scaling up to grant widespread approvals.

Whereas manned aircraft collision avoidance has been focused upon the avoidance of traffic only, SAA likely would need to add avoidance of terrain, obstacles, and possibly weather. The domain of SAA may be restricted to airborne use, or may extend to the surface, where it could be used to detect and avoid other aircraft, vehicles, people, and obstacles.

This chapter discusses the Certification process and some of its artifacts normally used to gain approval of avionics and discusses how these might be applied to a capability such as SAA. It also discusses the Standards process, identifies issues unique to UAS SAA, and discusses the role Standards play in certification.

96.2 Certification Process

This chapter discusses the certification process with reference to documents used by the FAA. Elsewhere, regulators should follow generally similar practices, although differences will undoubtedly prevail. In Europe, the Joint Aviation Authorities (JAA) develop analogous processes and regulations that frequently are similar to FAA's; these organizations seek harmonization. A difference is that the various European national authorities, rather than JAA, are the regulators there.

The discussion here is focused on approval of the SAA system, which is only part of the process of achieving airworthiness approval for the unmanned aircraft. The usual process is first to obtain certification of the aircraft's airworthiness compliance from the aircraft certification service. The second major step is to obtain operational approval from the flight standards service.

96.2.1 Project Specific Certification Plan

The applicant should not expect to find a fixed path that specifies the exact steps and documentation required. Industry and FAA have developed a guide for applicants to plan and manage the process, document those plans and their progress, and develop a successful relationship with the regulator (AIA 2004). Since systems have unique characteristics and different regimes of operation, the process must be tailored to those details. The FAA advises applicants to work closely with them, starting with the development of a Project Specific Certification Plan (PSCP). That plan would document the agreed-upon certification approach and describe processes as well as deliverable products. These normally consist of the following elements:

96.2.1.1 System Safety Assessment

Safety assessment is described in FAA (2008). This process presents a standard series of steps that provide assurance that the system's safety has been comprehensively studied and that likelihood levels of identified hazards are sufficiently low, corresponding to the severity of each. The applicant will need to compile all of the system's potential aircraft-level hazards and document them through a functional hazard assessment that assigns a severity level to each hazard, representing the worst credible operational effects. This approach has become standard methodology for safety management systems, which are recommended for airspace regulators worldwide.

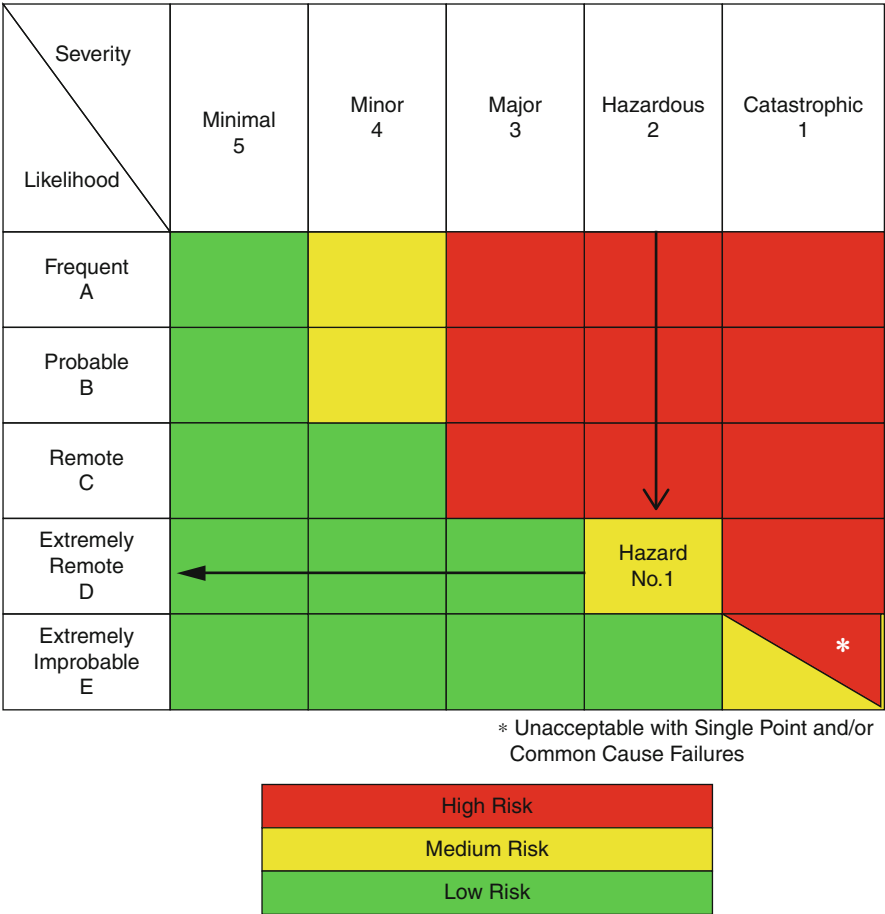


Fig. 96.1 Risk matrix

Figure 96.1 presents a version of the hazard risk matrix depicting an example hazard whose severity has been determined to be “hazardous.” The matrix then shows that this hazard’s safety objective needs to be “extremely remote.” This combination of severity and likelihood would be considered a “medium” risk. Standard practice is to identify and recommend safety requirements to assure that all risks fall within the medium- or low-risk zones of the table. The high-risk zone is not acceptable.

For SAA, a different treatment is appropriate for the residual collision risk that remains when the SAA system operates as designed. It is well accepted that a collision avoidance system (CAS) cannot operate perfectly due to several factors, particularly surveillance limitations, own aircraft maneuver limitations, and intruder aircraft maneuvers. If CAS reduces collision risk by two orders of magnitude, this is

usually considered to be a well-performing system (McLaughlin and Zeitlin 1992; McLaughlin 1997). However, the acceptable level cannot be prescribed for CAS in isolation – both self-separation and collision avoidance together will need to meet a safety target.

Since collision risk already exists throughout the airspace and CAS is introduced as one of the layers of protection to mitigate that risk, the usual SMS severity-based likelihood targets should not be applied to this area. Instead, metrics are used that can be termed “performance” measures of the SAA. When collision risk does arise from actual failure, e.g., from hardware, software, communication, or human failures, the SMS approach is appropriate.

Those hazards that could affect safe flight will require further investigation and documentation using safety tools such as those described in the next three subsections.

96.2.1.2 Fault Tree Analysis

This tool collects events, including pertinent conditions and failures, to show the structure of their contribution towards a “top event,” which is a hazard of interest. For example, a near midair collision (NMAC) event might occur if two aircraft are either on a collision course or in close proximity and SAA events occurred that might fail to prevent a collision in the first case or would cause a collision in the latter case. Those events must be decomposed into the subfunction outcomes of interest that could cause each higher event. Quantifying the constituent events and then combining them in a logic expression would enable the calculation of the top event. Figure 96.2 depicts a portion of an SAA fault tree.

Examples of a collision avoidance fault tree and quantified hazards populating that tree were produced in the early safety study reports for TCAS (Lebron et al. 1983; McLaughlin and Zeitlin 1992). The hazards included surveillance-related errors, of which altimetry was central to TCAS, as it advises avoidance maneuvers in the vertical dimension. SAA systems will need to address surveillance and tracking errors according to the sensors used, communication hazards, and display and human hazards affecting ground-based decisions and commands.

96.2.1.3 Failure Modes and Effects Analysis

This tool is often used in aircraft systems to systematically examine potential failure modes and explore the worst credible consequences of those failures. In most cases, the design contains sufficient redundancy or other mitigation so that the effects are acceptable, i.e., not too frequent for the associated severity. If this is not the case, safety recommendations would be needed to provide sufficient mitigation.

96.2.1.4 Common Cause Analysis

Safety analysis tools often examine one failure event at a time, and the analysis of complex systems is facilitated by the initial presumption of independence among failures. Sometimes this is not a valid premise, and the common cause analysis seeks

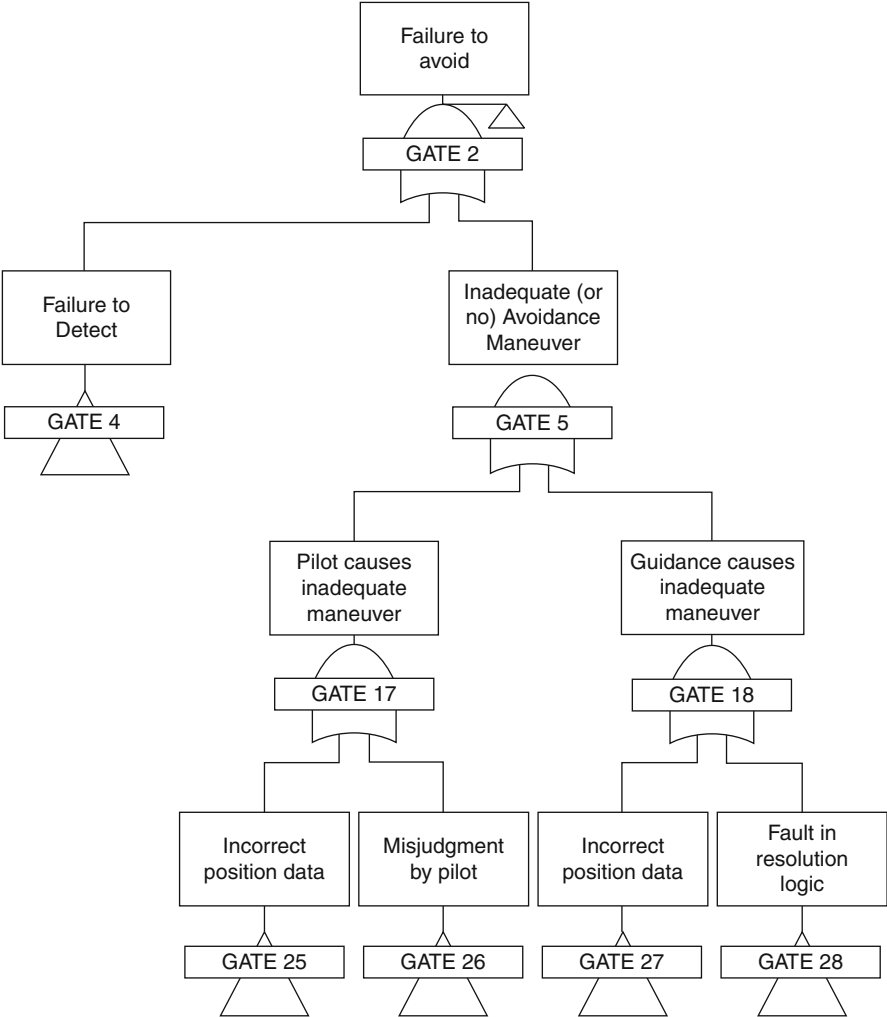


Fig. 96.2 SAA fault tree fragment

out vulnerabilities to common failures arising from a single cause. Performing this analysis and updating other safety analyses in accordance with its results would bolster the confidence in the overall safety case.

96.2.1.5 Certification Basis and Means of Compliance

The certification basis specifies the applicable regulations to which the applicant must comply in order to obtain the type certificate. SAA is novel in many respects, and it may not be obvious how to fit all of its aspects into the regulatory framework. The primary roles of SAA are to remain well clear from, and avoid collisions with,

other airborne traffic. SAA would operate in a manner analogous to the visual functions invoked by regulations 14 CFR §91.111, 113, and 181.

It is likely that the certification basis will include elements of safety, performance, and interoperability. Performance in this context refers to the extent that SAA is able to mitigate the existing risks of separation loss and of midair collision. This is distinct from safety risks arising from system or operator failures. Interoperability refers to SAA coexisting with other systems aboard own or other aircraft and exchanging data with them as required and also not causing any degradation to their function. The obvious example would be the interactions with a collision avoidance system aboard another proximate aircraft, where evidence would need to be provided that the maneuver advisories generated by the respective aircraft would be compatible, with extremely high probability.

96.2.1.6 Software

Although software does not break or wear out, it may contain hidden (or “latent”) failures that could cause hazards to occur. The section below on software standards discusses accepted methods for assurance of quality software, so that the likelihood of latent failure is commensurate with the criticality of the system functions.

96.2.1.7 Human Factors

If the UAS remote pilot is to play any role in SAA, he must be given the relevant data and advisory information in a presentation format that supports timely, necessary, and safe decision making. Possible functions for the pilot are the evaluation of risk, deciding that action is needed, and determining the specific action. These are discussed in Sect. 96.3. The pilot also should be supported in correctly deciding when action is not required, choosing a safe course after necessary SAA action is complete, and determining how far to deviate from course or from a flight level during an avoidance maneuver. These concerns all apply, whether the remote pilot is responsible for the decision (termed “in the loop”) or is monitoring and able to override an automated decision (“on the loop”). Besides providing correct and complete information to support the intended functions, the applicant will need to demonstrate a sufficiently low likelihood of providing the pilot with misleading information that might induce hazardous choices. The pilot’s training, workload, and potential distractions all enter into the evaluation. The pilot actions will need to be translated into system requirements in connection with the development of the safety case for SAA.

An avionics system advisory circular normally provides examples of relevant considerations that should parallel many SAA topics. The TCAS II circular (FAA 2009a) was written to complement the TCAS standard. It gives some description of the system and its components and suggests implementations of traffic and resolution advisory displays and pilot controls whose designs were backed by considerable research and consensus. Standards organizations with expertise in the human factors discipline were cited, but it is mentioned that alternative implementations may be approved with sufficient research and validation. A discussion of a certification

approach includes both verification tests contained in the TCAS standard and also related tests of aircraft maneuver performance. Ground and flight tests of the installed system are discussed.

96.2.1.8 Others (Operational Concept, Equipment Compatibility, Envisioned Safety Enhancements)

It is recommended that applicants work with the regulator to document the identification and resolution of important technical, regulatory, and administrative matters in issue papers. Such papers are updated as the status and resolution of issues evolve.

The PSCP also would contain an agreed-upon testing plan and specify the documentation to be produced showing compliance. The familiarity or novelty of the system bears a great deal upon the extent of both testing and documentation to be expected. As long as UAS and SAA are new to the airspace, more rigor and detail could be expected; later, familiar approaches may rely more upon experience and focus more on marginal differences.

The ultimate certification product for avionics is normally the Technical Standard Order (TSO), which is a design and production approval that designates the relevant equipment standards. The aircraft using the avionics would reference it in its application for its type certification (TC), supplemental type certification (STC), or amended type certificate. The latter two types could apply to alterations to an aircraft through installation of a system (FAA 2007). These certifications establish the regulator's approval of the aircraft design. An advisory circular also may be issued to provide guidance on complying with applicable airworthiness regulations. The circular may provide an example of an acceptable means of compliance that is well understood by the regulator, but also would state that this is not the only means of compliance that could be approved. The applicant may choose any means of compliance, but then must demonstrate compliance to the satisfaction of the regulator. Standards, which are discussed in the next section, facilitate that demonstration. FAA (2013) provides a partial draft of such a circular for SAA systems.

96.3 Standards

The purpose of a standard is to document a consensus on the requirements for a system and describe a means for demonstrating that those requirements are met. While the standard itself conveys no regulatory authority, its publication provides a service to applicants and regulators alike. Applicants, by invoking the standard and demonstrating compliance, would not need to justify its set of requirements, and regulators would not have to judge the set of requirements within each implementation. If, however, an applicant chose to depart from the standard, the resulting certification process would need to include a justification for accepting the alternate approach.

A standard would be most likely to be respected if it was written with broad input from all affected parties, particularly government, manufacturers, and user groups. In the USA, most avionics standards are developed by the aviation standards organization RTCA, Inc., and in Europe by the European Organisation for Civil Aviation Equipment (EUROCAE). These two organizations sometimes collaborate to produce joint standards, and at times a standard developed by one is endorsed by the other organization. That was the case for TCAS, but the organizations appear to be moving separately for UAS.

The contents of a standard typically include at least a description of the subject and its scope; overall system requirements; requirements for each subsystem; a means of demonstrating compliance, such as specific testing procedures; references to applicable documents; and supporting information, sometimes placed in Appendices.

RTCA, Inc., generally prepares two types of standards: Minimum Aviation System Performance Standards (MASPS) that address functional performance but allow some latitude regarding system architecture and technology selections and Minimum Operational Performance Standards (MOPS) that are far more specific to the architecture and technology.

Many standards tend towards a minimalist approach, specifying only those aspects judged necessary to be performed in a defined, consistent manner. This approach holds some appeal for manufacturers, who may exploit more flexibility in utilizing proprietary architectures and technologies or in making performance trade-offs among subsystems. Some of these standards may describe a specific implementation as an example but would still leave great flexibility to users. MASPS follow this approach to some extent.

An opposite approach to standards would specify many elements of the system in some detail. This would provide the regulator with greater assurance that supporting data for one implementation would apply to all others, as little variation would be allowed. In certain respects, commonality may be required for safety and interoperability. Examples might include physical or electrical interfaces, radio messaging, or software-intensive systems with critical functions. An instance of the latter is the MOPS for the Traffic Alert and Collision Avoidance System II (RTCA 2008), the aircraft collision avoidance system. Its algorithms were intensively simulated and tuned, and a method of maneuver coordination between aircraft was detailed. Its standard specified a complete algorithm, thus assuring that all aircraft using it would receive a known level of protection and would be able to coordinate maneuvers in the prescribed fashion.

The decision regarding minimum versus specific standards does affect the specification of compliance demonstration. A standard may provide a list of tests and associated conditions, which if passed would provide assurance that the system meets the standard's requirements. Such tests may be more difficult to devise if implementation methods are not constrained.

The First SAA Workshop Report (FAA 2009b) established two basic system functions: self-separation and collision avoidance. These two functions each need to perform the following eight subfunctions:

1. Detect – Determine the presence of aircraft or other potential hazards.
2. Track – Estimate the state (position and velocity) of each intruder based on one or more surveillance reports.
3. Evaluate – Assess collision risk based on intruder and UA states.
4. Prioritize – Determine which intruder tracks have met a collision risk threshold.
5. Declare – Decide that action is needed.
6. Determine Action – Decide on what action is required.
7. Command – Communicate determined action.
8. Execute – Respond to the commanded action.

The degree of independence of the basic functions affects the safety case, since collision avoidance is expected to act once self-separation fails to ensure separation. While there may be programmatic motivations to use the same or similar subfunction implementations (e.g., sharing some or all of the sensors supporting detect) for both functions, the safety analysis needs to give close attention to the effects of common failures. Any credible failure that would impair both self-separation and collision avoidance would be a concern and might necessitate a more robust implementation in order to meet a safety level. For sensors, this concern over a common failure might be addressed by deploying multiple technologies to improve the prospects of sensing any threat aircraft by at least one of the sensors.

This list of subfunctions is independent of system architecture. If a standard is written for SAA using a specific architecture, that needs to be specified as well as to which system elements (e.g., aircraft, ground control station, pilot) each subfunction is allocated. In some instantiations, elements from this list can be incorporated in system functions external to SAA. For example, command is likely to be allocated to the communications function and execute to the aircraft flight control system.

96.3.1 Requirements for Functions and Subfunctions

This section presents considerations that may lead to requirements for functions and subfunctions. The discussion of the two functions draws upon Fig. 96.3 taken from FAA (2013). It depicts notional boundaries, not drawn to scale, that are explained in the following paragraphs.

96.3.1.1 Self-Separation

The best insight into this system function appears in the 2nd SAA workshop report (FAA 2013). It is meant to implement the compliance with 14 CFR 91.111 to “remain well clear” of other aircraft by providing an automated substitute for see and avoid. The report acknowledges that those UAS receiving separation services from air traffic control would not use this function unless authority to do so had been delegated to them. For uncontrolled UAS, though, this function would provide the only means to prevent collision avoidance situations from developing. A specific risk-based approach to defining a self-separation volume was presented in the SAA workshop report; standards committees will need to consider operational and technical factors (such as performance aspects

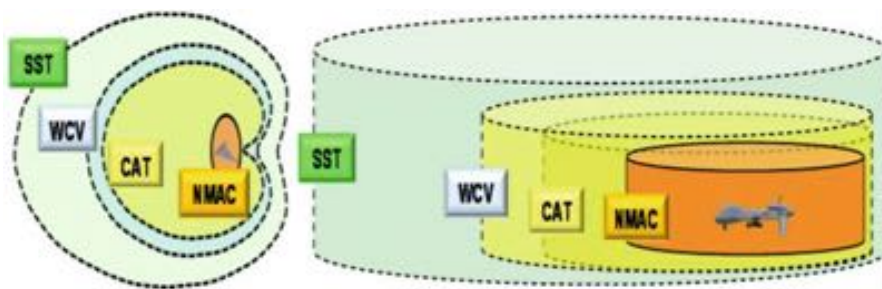


Fig. 96.3 SAA function boundaries

of the surveillance capability) when specifying requirements on the volume to avoid and the associated maneuver decision time.

In Fig. 96.3, SST depicts a self-separation threshold that is defined with the purpose of achieving a selected probability of avoiding penetration of the boundary denoted WCV, standing for a well-clear violation. The WCV is chosen in accordance with calculation of collision risk in the airspace (Weibel et al. 2011), based upon observed historical encounter data (Kochenderfer et al. 2010).

96.3.1.2 Collision Avoidance

The TCAS system provides a well-established definition for a collision volume: 200 ft vertical limit and a circle of 500 ft diameter, both centered about the UAS. This volume is labeled NMAC in Fig. 96.3. The maneuver to avoid this volume, which is associated with a “near midair collision” event, needs to occur no later than the last point at which the UAS could begin a feasible maneuver. That last-chance maneuvering boundary is denoted CAT for collision avoidance threshold. Design of the associated decision time needs to account for both the latencies of a command link (if the command is communicated from a control station) and of the aircraft flight control system.

It would be desirable to jointly design SST and CAT for an implementation, as any gaps or overlaps between the WCV and CAT boundaries could be minimized; but this may not be practical for all implementations. In any event, the following considerations are essential:

- (a) The two functions together must achieve the requisite level of collision risk reduction.
- (b) The CAT will be smaller than the SST.
- (c) When the aircraft penetrate the WCV, the self-separation function must continue to operate and attempt to restore the well-clear state until the collision avoidance function activates.
- (d) While collision avoidance is issuing maneuver advisories, the self-separation function must cease to command maneuvers, as these could dangerously conflict with those of collision avoidance or, at the least, could be distracting.

96.3.2 Subfunction Considerations

96.3.2.1 Detect

Various sensor technologies are under consideration for SAA. For an airborne SAA, these include “cooperative” technologies, which exploit electronic identification provided by the target aircraft avionics, and “noncooperative.” Cooperative technologies include the ATC transponder, which already is used as the basis for TCAS (RTCA 2008), and automatic dependent surveillance – broadcast (ADS-B), which provides the target’s position, usually derived from a satellite-based estimate, its velocity, and other information (RTCA 2009). The noncooperative technologies include electro-optic (camera), infrared, and primary radar not requiring cooperative avionics on the other aircraft.

In some airspace, it can be expected that all targets will be cooperative due to regulatory requirements; in others, noncooperative sensing is essential. It is anticipated that most SAA implementations would combine more than one sensor in order to detect both kinds of traffic. Also, the noncooperative technologies have various strengths and weaknesses (Zeitlin et al. 2009). Combining sensors could exploit the respective strengths.

The various sensors differ in many measurement parameters, including their detection range, angular field of view (FOV), bearing accuracy, and update rate. They also differ in mechanical properties including weight and electrical power requirements, which is important for the smaller UAS and may lead to trades that limit the FOV or range capability. Requirements for FOV coverage will likely consider the probabilities of encountering a threat over different angular bearing sectors, considering feasible relative speeds. Clearly this would vary with the speed of the UAS as well as the characteristics of the airspace population. Another issue related to FOV ties into the conventional formulation of right-of-way rules. Currently, an overtaking aircraft must give way to the aircraft ahead, according to 14 CFR 91.113. This has led some to speculate that SAA need not detect traffic approaching it from behind. While cooperative sensing technologies generally have a 360° lateral FOV, others, such as electro-optic and primary radar, do not. Extending their FOV would require multiple sensors, with associated weight, power, and cost implications. The SAA workshop (FAA 2013) recommended that SAA must not rely on right-of-way requirements to reduce their compliance with see-and-avoid requirements. This implies that SAA must equip with an appropriate FOV so as to detect targets behind own aircraft.

A standard could reference other existing standards for the cooperative sensor technologies and could provide detection requirements for noncooperative targets. An associated requirement may be a limit on the rate of false target detection.

96.3.2.2 Track

The track function associates successive measurements for each target and performs smoothing to derive an estimate for the target’s current position and velocity.

It needs to accommodate varying update rates from different sensors; different times of measurement and missed measurements; and different accuracies. Also, some sensor technologies provide different coordinates than others or from different frames of reference. This function must account for all the differences in producing its best estimates. These estimates may also provide the track's estimated error to the subsequent functions, allowing them to put in a safety buffer that would guard against worst-case errors.

Associating measurements (or "reports") from different sensors is an important consideration, as the cooperative technologies provide a unique identifier, while the others must attempt to associate noisy measurements.

96.3.2.3 Evaluate

This function would use the track variables to calculate other variables more suitable to threat evaluation. An example of such a variable from the TCAS algorithm that is likely to be useful in a SAA context is called "tau." This is an estimate of the time remaining until reaching the closest point of approach to the target. The estimate works over any approach angle and speed. The TCAS tau only uses range and range rate, as those variables were the most accurate measurements using the TCAS surveillance. When other sensors are available for SAA, a different formulation of tau may prove to be more accurate and thus preferable. Other evaluation variables would be the calculated vertical and horizontal miss distance estimates at the closest point of approach (CPA). These would be used by the subsequent function in deciding the priority of the threat situation.

96.3.2.4 Prioritize

The prioritize function would compare track and/or calculated variables against thresholds to determine the threat status and urgency of taking action. Even if no action is indicated by the subsequent function, the priority may be communicated to the pilot on the traffic display, as TCAS does for various target symbols types. This could focus pilot attention on one or more threats potentially needing action soon.

96.3.2.5 Declare

This function combines the priority of a track's status with other rules that determine whether immediate action is required. These rules may be geometry dependent or may account for different criteria, such as confidence in the track quality or the status of coordinating maneuvers with the threat's own collision avoidance maneuver.

96.3.2.6 Determine Action

Numerous considerations play a role in determining the specific SAA action. Foremost should be the choice of a maneuver to avoid penetrating the particular volume for either self-separation or collision. (If the declaration decision was based on a risk criterion rather than upon a physical threshold, it may be difficult to represent as a physical volume.) Since more than one maneuver choice may be capable of meeting this first criterion, other rules are involved in the decision.

For example, it might be found best to defer action as long as possible, either because the track quality improves with decreasing range or because encounter data shows that potential encounters are often resolved without requiring a maneuver. In this case, the rule might defer action until only one feasible maneuver remains.

An essential rule takes precedence when the threat aircraft is also equipped with collision avoidance (as are air carriers worldwide). This rule requires the explicit coordination of collision avoidance maneuvers between the two aircraft in order to assure mutual compatibility. For example, both aircraft cannot be allowed to climb nor both to descend. One must choose a different sense for its resolution. The existing TCAS standards dictate coordination messages and protocols to ensure this compatibility between two TCAS aircraft, and these can be used or adapted for SAA. An adaptation would be needed, for example, if policy were to determine that the UAS should be given lower priority in tie-breaking situations. This direction may be justified on the basis that SAA is less mature, or on lower confidence in its sensing or tracking, or on the basis of a UAS's lesser maneuver capability.

Another criterion might consider rules of the air, for increasing the likelihood of the maneuver's compatibility with the other aircraft's (uncoordinated) visual avoidance actions. Such a rule might be considered a means of implicit coordination.

Operational criteria might consider proximity to other nonthreatening traffic or to restricted airspace areas, in order to avoid choosing a maneuver that could create a new problem with that traffic or cause an airspace violation.

96.3.3 Architecture

Several options exist for the physical architecture of the SAA. The sensors could be located on the UAS, on the ground, or even both of these in combination. Likewise, the decision making could be on the UAS, at a ground control station, or divided. Unless all functions are performed onboard the UAS, a communications link is required. It may downlink surveillance data to the ground control; it may uplink maneuver commands to the UAS; one or both of these could be implemented.

An example of divided decision making might allocate self-separation decisions to the ground control, where ample time remains to resolve a conflict. For this function, the criticality of a short-term communication outage is low. The collision avoidance function might be performed automatically onboard the UAS in order to avoid implementing an extremely robust uplink capability. For this function, little time remains before a potential collision, and the execution of the maneuver is more critical.

Many trade-offs can affect the choice of airborne or ground-based sensing. A larger UAS could equip with multiple or heavier sensors. This might provide it with a broader FOV than a single sensor. Multiple sensor technologies offer the possibility of combining their respective advantages, such as accuracy in different dimensions. However, such a SAA system would need to perform fusion of the data from the several sensors. Trade-offs might risk sacrificing some of the desired

surveillance parameters. Instead, using ground-based radar could provide coverage that surrounds the UAS and also would not require data fusion. A limitation of this approach is the fixed geographic area of the radar, whereas an airborne surveillance system travels anywhere the UAS flies. Ground-based radars also may not provide adequate accuracy and update rate for collision avoidance. However, their ability to support self-separation is more credible, as such radars are routinely used for air traffic control separation services.

96.3.4 Communications

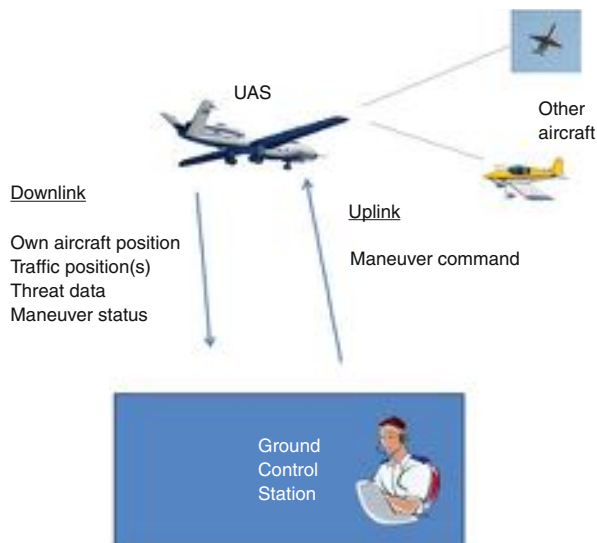
A UAS communications function has many roles. Besides navigation, the mission payload control, and monitoring the aircraft, there are SAA functions that will rely on communication to varying degrees, depending upon the SAA system architecture (Fig. 96.4). The *downlink* is important for aircraft-based SAA designs. A ground control station will need current data for situational awareness of traffic and other hazards. It also will need to obtain data for any potentially threatening aircraft if the pilot is involved in threat declaration and/or a maneuvering decision. Threat data can include that aircraft's equipage for collision avoidance and whether it has announced a resolution of its own. A design challenge may arise in providing sufficient information to the pilot with low latency and sufficient update rate. These elements place requirements on the communications downlink. If threat declaration or maneuver choices are made by onboard system elements, the ground control still must monitor these choices and be prepared to recover the flight profile after the maneuver ends. It needs to be determined whether any maneuvers would be communicated to air traffic control directly from the aircraft or by the pilot. Finally, if a maneuver is selected, or is approved, on the ground, the *uplink* is time critical and must be reliable. The consequences of a late or erroneous maneuver command could be a collision.

96.3.5 Algorithms

An SAA system is expected to use a threat algorithm to make decisions regarding the status of each tracked aircraft. It needs to determine when a maneuver may be necessary and determine the specific maneuver, unless the system leaves the choice entirely to the pilot. The algorithm needs to function for all encounter geometries and should be capable of acting when multiple aircraft are near the UAS.

There is interest in building systems that would maneuver automatically. Some developers seek to overcome communications latency, and others wish to provide collision protection even when the communications link from the pilot is lost. Removing the pilot from the decision loop forces a higher degree of reliance on the algorithm always providing a correct maneuver. In addition, a higher software assurance level may be required.

Fig. 96.4 SAA communications



The primary SAA requirement is to contribute to safety by effectively executing the self-separation and collision avoidance functions. However, of nearly equal importance is the requirement to operate in an acceptable manner to both air traffic control and other airspace users. This would involve maneuvering in predictable fashion, not flying so near to other aircraft as to endanger them, and not resolving an encounter with one aircraft in a manner that creates a new encounter with another.

A new development being pursued for collision avoidance replaces the traditional procedural algorithm with a look-up table developed by an optimized Markov decision process (Kochenderfer et al. 2010). The optimization is based upon a cost function that incorporates both the safety criteria and operational criteria discussed in this paper. It also incorporates probabilistic distributions of aircraft maneuvers and surveillance errors.

In addition to the technical challenges of developing and standardizing SAA algorithms, attention must be given to the issue of product liability. The issue was averted for TCAS when that government-funded program developed its algorithm and the standard mandated its use. It has not been determined whether any similar government involvement will transpire for SAA, nor whether standards will attempt to specify a single algorithm, nor how industry may choose to proceed knowing their unique efforts might become subject to legal action following an accident.

96.3.6 Validating Requirements

Several methods contribute to the validation of requirements contained in a standard. As mentioned earlier, conformance to the standard is expected to make a major contribution to the safety case for the system. For collision avoidance

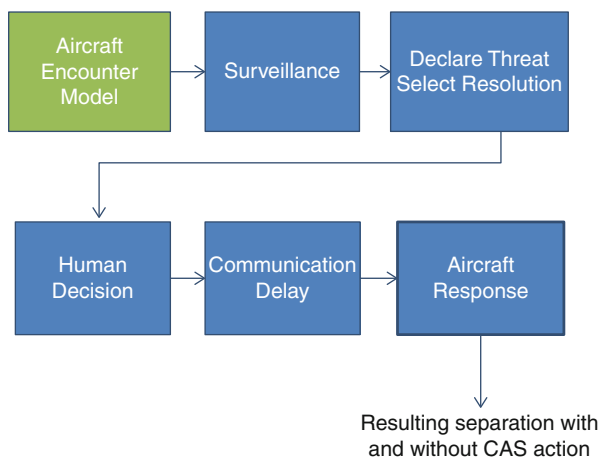


Fig. 96.5 Monte Carlo simulation technique

systems, experience shows that a substantial program of methodical encounter simulation would exercise modeled system elements and enable a calculation of the system's overall safety performance. TCAS simulations (McLaughlin and Zeitlin 1992) have used millions of simulated encounters in order to achieve an agreed statistical basis for performance.

Such a simulation is no better than the models that it contains. Encounter models are considered more trustworthy when derived from actual observed data. They can be validated by comparing separate samples of the data as well as by comparing separate implementations of the model (Zeitlin et al. 2006). Figure 96.5 depicts a system simulation that combines models of aircraft geometric encounters, SAA sensing and tracking, threat detection and advisory selection, and resulting maneuvers. It also shows an optional insertion of human performance in the “command” leg; human performance also might be incorporated in threat detection and/or advisory selection.

Monte Carlo simulation is an established technique that uses separate models like those in Fig. 96.5, whose parameters undergo independent stochastic variation, each with their appropriate distribution. While the same simulation might be used to study a single encounter of interest, the overall safety case depends on frequencies of events and resulting performance. For example, straight-and-level geometries might have a fairly high frequency, while encounters with both aircraft making very late maneuvers might have lower frequency. The latter might prove more difficult to resolve, so weighting each type by their observed frequency gives an estimate of overall performance. The TCAS model (McLaughlin and Zeitlin 1992) used 19 geometric classes, and newer correlated and uncorrelated models (Kochenderfer et al. 2010) use more classes. Typically, millions of encounters are run to model the overall airspace performance. From run to run, the separate variables change independently. For example, the encounter angle, vertical profile, and speed would

vary, while measurement noise would separately vary, and response fidelity and latency again would separately vary.

For collision avoidance, the established practice involves running each encounter twice with identical input – once with response to advisories and once without. This produces data to show the effectiveness of the CAS system, particularly the risk ratio that compares the number of NMAC events with and without CAS. Maneuver magnitudes and resulting displacements also can be obtained as well as advisory types and transitions during an encounter.

A necessity for SAA simulation is to obtain appropriate models of the various sensor technologies and integrate them into this simulation. It may be challenging to model emerging technologies in detail, and some of their performance measures could vary substantially depending upon the physical characteristics of the target aircraft (e.g., size, radar cross section, aspect angle during an encounter). Performance also might vary depending upon visual conditions, particularly for electro-optic sensing. It may be appealing to simplify the model and use conservative performance measures; evidence must be provided that actual performance would not be worse than that modeled.

Other aspects for consideration are probabilities of missed sensor reports and of false returns that could be correlated incorrectly with existing tracks. Flight trials are desirable to validate estimates of these parameters.

Likewise, appropriate models of the communications latency and aircraft maneuver response are crucial to the validity of the results.

A different technique for system validation is termed “stress testing” and also has been used for TCAS. This method uses end-to-end models like those shown above. Rather than running many trials with stochastic variations to replicate true frequencies across the airspace, this method samples key variables at intervals meant to exercise the full range of possible inputs and helps analysts identify patterns of poor performance. Even if those encounter situations were rare in the context of overall airspace exposure, stress testing identifies shortcomings that must be considered for correction before approving the system. It is accepted that SAA will not succeed in every encounter, but both its overall safety and specific limitations must be understood.

96.3.7 Verification Tests

Since the goal of the applicant is to demonstrate compliance with the requirements specified in the standard, there needs to be a process to perform that demonstration. Rather than expecting every applicant to invent such a process and convince the regulator that it is complete, the standard should provide an agreed set of test procedures. This set should be sufficient to demonstrate coverage of each requirement. Inputs, initial conditions, and outputs should be stated. Tolerances should be recommended to allow for reasonable variations within the intent of the standard.

Depending upon the system, the process may require a full end-to-end test or may test subsystems. Software might be tested in part using fast-time or state transition testing, while a full system may require RF signals or other inputs. Some degree of flight testing is likely to be required for at least a minimal demonstration of full system operation. However, flight testing is time consuming and expensive, so its extent should be limited with greater confidence built up from the other methods.

96.3.8 Software Standard

As software is the means of implementing threat algorithms and certain other parts of SAA (i.e., data fusion, tracking), the trust in those aspects is central to the approval of the system. Software is unique in its safety aspects and is covered in a standard of its own. The document (RTCA 2011), entitled “Software Considerations in Airborne Systems and Equipment Certification,” addresses the role of software in a complete system and its life cycle. It provides advice concerning the software planning, development, verification, and configuration management processes. It gives some insight into the certification process and recommends numerous artifacts, or data, to be recorded at various stages of the life cycle. In recognition of many unique approaches both known today and to be developed in the future, a section on other considerations includes advice on alternative methods, as well as changes made to approved software, and other topics.

At the system level, early consideration must be given to the relationship between software errors and failure conditions. The severity of those failures drives all the subsequent stages in terms of rigor of development and verification. It will be much easier to begin the life cycle at the correct level rather than attempt to restore inadequate rigor after development is complete.

The planning process includes such topics as the software development environment and software standards. The latter would document more detailed rules governing the specific aspects of requirements, design, and coding of the software.

The development process would document details of specific processes for the same requirements, design, and coding elements. It must be stressed that the modern approach to high-quality software is built upon the use of well-documented processes.

The verification process must detect and report errors introduced during development. This involves far more than simply testing the software. The process verifies that high-level system requirements flow down through lower level requirements levels and that all of the executable code ultimately is traceable to requirements. The process also ensures robustness to abnormal conditions and inputs.

Configuration control is described with multiple objectives designed to ensure an orderly and well-documented process, including traceability and change control.

The overview of the certification process explains that systems and equipment, including software, need to be “approved” in order to become part of a certified aircraft. Development tools can become “qualified” to an appropriate level of assurance.

The various data items described support either the planning process where the approach to certification is detailed or the compliance process where evidence is presented to show that requirements are fully satisfied.

96.4 Conclusion

Certification of SAA actually is an approval of a system's compliance with all regulatory requirements. Each certification plan is unique and should be coordinated closely with the regulator. Both standards and guidance documents such as advisory circulars should greatly simplify the demonstration of compliance. The standard contains requirements that would be validated through analysis and simulation. The standard also should present the testing requirements to demonstrate compliance by means of producing specified outputs.

References

- AIA, *The FAA and Industry Guide to Product Certification*, 2nd edn. Aerospace Industries Association, General Aviation Manufacturers Association, Federal Aviation Administration, Sept 2004
- FAA, *Guide for Obtaining a Supplemental Type Certificate*, AC 21-40A, Sept 2007
- FAA, *Safety Management System Manual v2.1*, May 2008
- FAA, *Airworthiness Approval of Traffic Alert and Collision Avoidance System II (TCAS II) Versions 7.0 & 7.1 and Associated Mode S Transponders*, AC 20-151A, Sept 2009
- FAA, *Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS)*, Oct 2009
- FAA, *Evaluation of Candidate Functions for Traffic Alert and Collision Avoidance System II (TCAS II) on Unmanned Aircraft System (UAS)*, RTCA 092-11/SC203-045, Mar 2011
- FAA, *Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS)*, Second Caucus Workshop Report, Jan 2013
- ICAO, *Global Air Traffic Management Operational Concept*, Doc 9854, AN/458, 2005
- ICAO, *Airborne Collision Avoidance System (ACAS) Manual*, Doc 9863, AN/461, 2006
- ICAO, *Annex 6: Operation of Aircraft, Part I, International Commercial Air Transport-Aeroplanes*, 2007
- ICAO, *Regional and Supplementary Procedures*, Doc 7030/4, 2008
- M. Kochenderfer et al., *J. Guidance Control Dyn.* **33**(2), 487–499 (2010). AIAA
- J. Lebron et al., *System Safety Study of Minimum TCAS II*, DOT/FAA/PM-83/36, The MITRE Corporation, Dec 1983
- M. McLaughlin, *Safety Study of the Traffic Alert and Collision Avoidance System (TCAS II) Final Version*, MTR-97W32, The MITRE Corporation, June 1997
- M. McLaughlin, A. Zeitlin, *Safety Study of TCAS II for Logic Version 6.04*, MTR-92W102, The MITRE Corporation, July 1992
- RTCA Inc., *Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II) Version 7.1*, DO-185B, June 2008
- RTCA Inc., *Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)*, DO-260B, Dec 2009

-
- RTCA Inc., *Software Considerations in Airborne Systems and Equipment Certification*, DO-178C, Dec 2011
- R. Weibel, M. Edwards, C. Fernandes, *Establishing a Risk-Based Separation Standard for Unmanned Aircraft Self-Separation*, MIT Lincoln Laboratory, Mar 2011
- A. Zeitlin et al., *Collision Avoidance for Unmanned Aircraft: Proving the Safety Case*, MP060219 and 42-1017, The MITRE Corporation and MIT Lincoln Laboratory, Oct 2006
- A. Zeitlin et al., *Sense & Avoid Technology – Not Ready for UAS?* AUVSI Unmanned Systems North America, The MITRE Corporation, 2009

Section XVII

UAV - Human Interfaces and Decision Support Systems

Paul Oh and Michael Goodrich

Kimon P. Valavanis and George J. Vachtsevanos

UAV: Human Interfaces and Decision Support Systems focuses on research findings and efforts that are underway by the UAV user community to reduce the current UAV operator workload levels. Until a satisfactory level of autonomy is achieved, the human will continue to function as an integral part of the UAV ground control station (GCS) loop. Moreover, intelligence acquired by UAVs must be communicated in an appropriate format to humans on the ground or to a mother ship for further analysis, assessment, and action. The human must intervene when new intelligence suggests that the operational profile must be changed.

► **Human Interfaces in Micro and Small Unmanned Aerial Systems** by Peschel and Murphy surveys the various aspects of user interfaces for a Mission Specialist role on micro and small UAS human-robot teams. Three formal human team member roles are presented, Flight Operations Director, Pilot and Navigator, and the Mission Specialist, in order to identify distinct team member role differences. For each category of UAV, the Mission Specialist role is evaluated to determine what type of user interface technologies are present and/or available, how the Mission Specialist interacts or could interact with the user interface technology, and what are the perceived consequences of this user interface technology in the context of the three identified micro and small UAS human-robot team roles. The findings of this survey suggest that current Mission Specialist performance in micro and small

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

UAS may be suboptimal due to the sharing of a single Pilot and Navigator-oriented user interface or a reuse of the Pilot and Navigator user interface.

► **Human Factors Perspective on Next Generation Unmanned Aerial Systems** by Goodrich and Cummings presents a comprehensive view on UAS autonomy and user interface design, motivating emerging themes in near-term and far-term UAS developments. Issues include user interface design, UAS autonomy, UAS teaming, operator workload, and payload management.

► **Cognitive Task Analysis for Unmanned Aerial System Design** by Adams provides an overview of cognitive analysis methods that can be integrated into the early system design process to account for the human element. It specifically provides an overview of task analysis, work analysis, and information flow analysis methods along with an overview of how individual methods from the different analysis classes have been combined. It is argued, and rightly so, that although UAS are complex tools requiring thoughtful designs that account for the platform itself, payloads, and human users as a holistic system, often the focus is first placed on designing the platform or payload, while the human element is considered at a later stage. As such, system design approaches do not properly consider the human cognitive limitations and do not design the system to support the human's tasks and decision making. Specific examples of applying these analysis methods to unmanned aerial systems, ranging from potential futuristic systems, to research systems, to deployed systems, are also reviewed.

► **Display and Control Concepts for Multi-UAV Applications** by Calhoun and Draper introduces a variety of display and control concepts that may be especially applicable to multi-UAV applications. Many such technologies are being investigated at the U.S. Air Force Research Laboratory and include multimodal interfaces, temporal displays, and integrated visualization concepts. The thesis is that for any candidate technology, rigorous user-centered design and evaluation needs to be conducted, which focuses on the tasks and requirements of the specific multi-UAV application being targeted. It is also important to ensure that the operator has adequate means to first observe and then direct the automation's functioning in order to be responsive to changes in the mission, vehicle status, or operational environment. What is important to capture is that advances in automation technology are leading to the development of UAV concepts in which a single pilot will be responsible for multiple UAVs. Thus, the operator's role will become more supervisory in nature, resulting in a new interface paradigm required for effective supervisory control of multiple UAVs by a single operator.

► **UAV Operators Workload Assessment by Optical Brain Imaging Technology (fNIR)** by Izzetoglu, Ayaz, Hing, Shewokis, Bunce, Oh, and Onaral discusses the utilization of functional near-infrared-based brain imaging systems, fNIRs, in the monitoring of a cognitive workload during UAV operation and as an objective measure of expertise development, that is, the transition from novice to expert during operator training. The challenge relates to the fact that UAV ground operators are required to acquire skills quickly and completely, with a level of expertise that builds the operator's confidence in his/her ability to control the UAV under adverse conditions. As UAVs are held to increasingly higher standards of efficiency

and safety, operators are routinely required to perform more information-dense and cognitively demanding tasks, resulting in increased cognitive workloads during operation. Functional brain monitoring offers the potential to help UAV operators meet these challenges, and research has demonstrated the utility of fNIRs for the purpose of monitoring frontal cortical areas that support executive functions (attention, working memory, response monitoring). fNIR technology allows for continuous monitoring of operators during training as they develop expertise, as well as the capacity to monitor their cognitive workload under operational conditions while controlling UAVs in critical missions.

Collectively, this section of the handbook introduces fundamental concepts associated with the still essential UAV-human interface, workload requirements, communication and display means, information processing, dynamic scheduling of UAV operations, assessment, and action issues, aiming at improving and optimizing decision making, mission effectiveness, and success, reducing at the same time the human-to-one-UAV ratio, which is currently close to 4:1. Consequently, long term, this will lead to mature technologies, the implementation of which will change the current human-in-the-loop requirement to the future human-on-the-loop concept.

Joshua M. Peschel and Robin R. Murphy

Contents

98.1 Introduction 2390

98.2 UAS Human-Robot Teams 2391

 98.2.1 Categories of UAVs 2391

 98.2.2 UAS Human Team Role Descriptions 2392

98.3 UAS Mission Specialist HMI 2394

 98.3.1 Micro UAS Mission Specialist HMI 2394

 98.3.2 Small UAS Mission Specialist HMI 2396

98.4 UAS Mission Specialist HMI Findings 2397

98.5 Conclusions 2401

References 2402

Abstract

This chapter surveys the various aspects of user interfaces for a *Mission Specialist* role on micro and small unmanned aerial system (UAS) human-robot teams. An operational characterization of micro and small unmanned aerial vehicles (UAVs) that are presently in use is first categorized into two size groups. Following this, three formal human team member roles are presented that trend from the micro and small UAS-related literature, *Flight Operations Director*, *Pilot and Navigator*, and the *Mission Specialist*, in order to identify distinct team member role differences. For each category of UAV, the *Mission Specialist* role

J.M. Peschel (✉)
Human-Infrastructure Interaction Lab, Department of Civil and Environmental Engineering,
University of Illinois at Urbana-Champaign, Urbana, IL, USA
e-mail: jmpeschel@gmail.com; peschel@illinois.edu

R.R. Murphy
Center for Robot-Assisted Search and Rescue, Department of Computer Science
and Engineering, Texas A&M University, College Station, TX, USA
e-mail: murphy@cse.tamu.edu

is evaluated to determine (1) what type of user interface technologies are present and/or available, (2) how the *Mission Specialist* currently or could interact with the user interface technology, and (3) what the perceived consequences of this user interface technology are in the context of the three identified micro and small UAS human-robot team roles. The findings of this survey suggest that current *Mission Specialist* performance in micro and small UAS may be suboptimal due to the sharing of a single *Pilot and Navigator*-oriented user interface or a reuse of the *Pilot and Navigator* user interface.

98.1 Introduction

UAS operations involve a human-robot team (Barnes et al. 2000; Cooke et al. 2006; Goodrich et al. 2007; Murphy et al. 2008; Rash et al. 2006). For the purposes of this discussion, the UAS human-robot team is defined as the human personnel primarily responsible for UAS flight, navigation, and acquisition of mission-related information and excludes consumers of information without direct control over the payload or platform (referred to as “knowledge workers” in Murphy and Burke (2008)). Human team members may be colocated with the UAV or at a remote location and, depending on the type of UAV and mission, can vary in number. Additionally, human team member functional roles may overlap.

Murphy et al. (2008) identified three roles in a micro or small UAS team: *Pilot*, *Mission Specialist*, and *Flight Director (Safety Officer)*. The bulk of human-robot interaction literature appears to be on the *Pilot* and reducing or combining that role with that of the *Mission Specialist*. Murphy et al. also suggest that more *Mission Specialists* will be added to the team as networking access becomes more pervasive. However, the *Mission Specialist* role is not well documented.

HMI research for specific UAS human team roles does not appear in the literature, presenting a challenge for designers and developers working with current and future unmanned systems. The HMI knowledge void becomes manifest as research efforts attempt to improve UAS capabilities by instead focusing on, among other things, reducing the human-robot crewing ratio through merging human team roles and increasing UAV autonomy (Cooper and Goodrich 2008; Cummings and Mitchell 2005; Cummings et al. 2007), increasing the number of UAVs in a single UAS (Burkle et al. 2011; Manathara et al. 2011; Tsourdos et al. 2011), and making UAS smaller, more mobile, and available to more diverse domains (Parrot 2011), without first understanding how human team roles are actually interacting. As Hobbs (2010) points out, there have been no human factors analyses published on any mobile interfaces for unmanned systems. The present lack of HMI understanding inhibits researchable improvements in UAS capabilities that may be possible by considering individual (and team) interactions within a UAS human-robot team.

This work surveys the hardware and software interfaces currently used by the *Mission Specialist* role on 10 micro or small UAS human-robot teams. Understanding human-machine interfaces is a first step towards understanding how a human fulfills the *Mission Specialist* role through human-machine interaction (HMI), which is critical for investigating general human-robot interaction in UAS, reducing the human-robot crewing ratio, and improving the individual role and team performance. However, research and development to improve the HMI experience of UAS interfaces has largely focused on UAS flight and navigation (Tvaryanas 2006; Walters and Barnes 2000). A HMI approach to support the acquisition of data and mission-related information remains historically less well developed (Cooke and Chadwick 2010), particularly for micro and small UAVs, as does an understanding of the HMI aspects of the *Mission Specialist* as the human team member responsible for such acquisition.

The chapter begins by providing an overview of two categories of UAS (micro and small) and the three roles common to all systems (*Flight Operations Director*, *Pilot and Navigator*, and *Mission Specialist*). It then produces a comprehensive summary of hardware and software HMI by UAS category based on over 20 papers and systems. The chapter discusses the implications for the *Mission Specialist* role and shows, especially in micro and small UAS operations, that there exist more user interface limitations present and, ultimately, that there may be a user interface interaction conflict between the *Mission Specialist* role and the team member responsible for flight and navigation. The work is expected to (i) serve as a reference document for unmanned system designers and developers and (ii) contribute to a better understanding of human-robot interaction in UAS (and in any robot being used for real-time data collection) by identifying the human-computer interfaces used in practice.

98.2 UAS Human-Robot Teams

In this section, background information is presented on two different categories of UAVs (based on size, payload weight capacity, range, altitude, and endurance) that may be employed for a mission, and the three roles present in the human team trending from the literature (*Flight Operations Director*, *Pilot and Navigator*, and *Mission Specialist*) that perform the actual UAV operations.

98.2.1 Categories of UAVs

The composition of a UAS human-robot team depends largely on the complexity of the UAS itself (Hopcroft et al. 2006). It is, therefore, necessarily cogent to the discussion to first provide a summary classification framework of selected UAVs that a human team may operate. A two-category classification system is employed, micro and small, which is consistent with the size categorization of the U.S. Air Force (Bierbaum 1995; United States Air Force (Online)),

Table 98.1 Classifications of selected unmanned aerial vehicles (UAVs) currently in operation^a

| Group | UAV platform name | Size ^b (m) | Weight ^c (kg) | Range (km) | Altitude (km) | Endurance (h) |
|-------|-----------------------------|-----------------------|--------------------------|------------|---------------|---------------|
| Micro | AirRobot AR100B® | 1.0 × 1.0 | 0.2 | 0.5–1.4 | 0.9 | 0.2–0.5 |
| | Aeryon Scout | 0.8 × 0.8 | 0.3 | 3.1 | 0.5 | 0.2–0.3 |
| | Draganflyer X6 | 0.9 × 0.9 | 0.5 | 0.5 | 2.4 | 0.2–0.3 |
| Small | AeroVironment Raven® | 1.1 × 1.3 | 0.2 | 10.0 | 4.6 | 1.3 |
| | AAI Shadow 600 | 4.8 × 6.8 | 41.3 | 200 | 4.9 | 12–14 |
| | Northrop Grumman Fire Scout | 9.1 × 8.4 | 272 | 127 | 6.1 | 5–8 |

^aMaximum operational parameters are reported and referenced from manufacturer specification sheets – normal operational parameter values will usually be lower and domain dependent

^bDimensions given are length × wingspan

^cThe maximum payload weight the vehicle can carry

Army (United States Army (Online)), and Navy and Marine Corps (Office of the United States Secretary of Defense (Online)) (Table 98.1). It is noted that for the purposes of this work, focus is restricted to subsonic and suborbital UAVs.

The first group consists of micro UAVs. This category of vehicle represents the smallest physical size, operational range (distance of travel), altitude (elevation above ground or sea level), and endurance (time of operation) of all UAVs, and it is the vehicle type most commonly available for commercial and civilian operations, such as wilderness and urban search and rescue. Micro UAVs allow a human team, which is usually colocated, to remotely navigate and visualize information in environments where, for example, humans or other ground-based robots are not practical. UAVs in the micro-category are traditionally of a rotor- or fixed-wing design.

The second group consists of small UAVs. Small UAVs expand upon the operational range, altitude, and endurance of the human-robot team without a significant change in the physical size of the vehicle. This would be important, for example, to on-site military combat units who will colocate with the UAV but need to maintain a large displacement distance for reconnaissance operations. Increased levels of autonomy are also found in small UAVs. One of the main differences between micro and small UAVs, besides an improvement in operational characteristics, is the dominance of fixed-wing vehicles and the increased payload weight capacity in the small category; very few rotor-based vehicles have been developed with small UAV operational parameters.

98.2.2 UAS Human Team Role Descriptions

There exists a diversity of human team role nomenclature in the literature across the various UAVs that have been developed and studied (Murphy et al. 2008;

Cooper and Goodrich 2008; Oron-Gilad and Minkov 2010; Cooke et al. 2006; Hopcroft et al. 2006). However, certain human team member role trends are identifiable and, generally, fall into one of three role categories: *Flight Operations Director*, *Pilot and Navigator*, and the main focus of this work – the *Mission Specialist*. These three role labels represent a synthesis from the literature and are given in this work to describe general role function rather than preferred role identification within any specific UAS. A human team member role survey from the current literature is provided for each of the UAV classifications shown in Table 98.1.

98.2.2.1 Flight Operations Director

Proper supervisory planning, coordination, and control of any operation are critical to its success, especially for a UAS human-robot team. Across the UAS literature, one or more human team member responsible for directing the mission was found to be a reoccurring role.

Murphy et al. (2008) define the role of a micro UAV team *Flight Director* as the individual responsible for overall safety of the team members (human and UAV). The *Flight Director* is in charge of mission situational awareness and has the authority to terminate the operation at any point. Cooper and Goodrich (2008) describe for a micro UAV team an Incident Commander, who is solely in charge of managing the search effort. For the small category, Oron-Gilad and Minkov (2010) ethnographically describe a UAS *Team Commander* role. Serving as the head of the human-robot team, the *Team Commander* may communicate with other UAS human-robot teams in the field or control stations and, in addition, monitor the technical condition of the vehicle. More complex situations described did arise requiring an additional individual, a *Mission Commander*, to join the team in order to focus only on strategy and coordination.

98.2.2.2 Pilot and Navigator

Piloting the vehicle is an essential human role and common to all UAS human-robot teams. However, the degree to which one or more individuals are solely responsible for flight control activity varies.

The role of a micro UAV *Pilot* defined by Murphy et al. (2008) is the human team member responsible for teleoperating the UAV within line of sight. They further indicate that the *Pilot* is responsible for the general airworthiness of the UAV prior to and during flight and addresses maintenance issues of the vehicle. According to Cooper and Goodrich (2008) who also address a micro class UAV, the *Pilot* is responsible for both aviation and navigation. For the small category, higher levels of UAV autonomy come into play, and the formal *Pilot* role transitions to a separate offsite operations control center. Oron-Gilad and Minkov (2010) provide detail on a *Field Operator* role that gives input as needed regarding where the UAV should fly; however, this role appears to, at best, have limited UAV flight control and navigation input capabilities.

98.2.2.3 Mission Specialist

UAV operational capabilities allow a human-robot team to insert themselves remotely for the main purposes of visual investigation and recording and, in more advanced vehicle systems, delivery of an onboard payload. It is therefore incumbent that one or more members of the human team be responsible for carrying out these kinds of activities. This role is referred to as the *Mission Specialist*.

Murphy et al. (2008) define the role of a micro *Mission Specialist* as a single human team member solely in charge of collecting reconnaissance data. Specific responsibilities include viewing the real-time video output from the UAV camera, directing the *Pilot* for reconnaissance, and adjusting the UAV camera settings for optimal image capture. Also for micro UAVs, Cooper and Goodrich (2008), who denote their human team member role as the *Sensor Operator*, assign the responsibility of directing a gimbaled camera for scanning and imagery analysis. In the case of a small class UAV, Oron-Gilad and Minkov (2010) describe an *Operator* role that is responsible for looking at specific areas and targets to evaluate the occupancy status of enemy troops. In their study, the operator focused on reconnaissance and the tactical aspects of the UAV mission.

98.3 UAS Mission Specialist HMI

As a subcategory of traditional human factors research, human-machine interaction (HMI) is a design science that has developed through experimental psychology methods being applied to techniques and tools available from within the discipline of computer science (Schneiderman and Plaisant 2010). In the context of this work, the interest is on how the human team member in the *Mission Specialist* role specifically interacts with the computer technology available during a UAV mission. The attempt is to address, where available, both the hardware and software aspects of human interfacing in each of the classifications of UAVs shown in Table 98.1.

98.3.1 Micro UAS Mission Specialist HMI

The details of *Mission Specialist* HMI for micro UAVs are the least well documented among the two UAV categories, which may likely be due to the often non-domain-specific nature of use in commercial and civilian applications.

Murphy et al. (2008) used a Like 90 T-Rex rotary-wing UAV in order to survey damage in post-Hurricane Katrina and post-Hurricane Wilma operations. The *Mission Specialist* role observed the real-time video feed from the T-Rex camera on a separate display screen and used independent radio control hardware for camera positioning. A second study by Murphy et al. (2008) during a separate post-Hurricane Katrina operation involved the use of an iSENSYS IP-3 rotary-wing UAV. Here the *Mission Specialist* role wore a heads-up display (HUD) for real-time visualization and utilized radio control hardware for positioning of the payload camera. In a study on wilderness search and rescue exercises, Cooper and

Fig. 98.1 Example of a micro UAS *Mission Specialist* interacting with an AirRobot AR100B® UAV (Courtesy of Center for Robot-Assisted Search and Rescue)



Goodrich (2008) employed the use of experimental fixed-wing UAVs fitted with a gimballed camera. The *Mission Specialist* visualized the video feeds from the UAV on a display screen and controlled the camera settings using independent radio control hardware. Figure 98.1 provides an example of a *Mission Specialist* interacting with a micro UAV.

Though not formally studied in the literature, there are several commercially available micro UAVs. User interaction with these UAVs ranges from simple hardware-based radio control to more sophisticated software-based control interfaces. Skybotix Technologies offers the CoaX®, a coaxial helicopter capable of general surveillance through a fixed-mounted onboard camera. An open-source application programming interface (API) is available to allow for flight control customization by one or more team members; however, the onboard camera is not controllable (Skybotix Technologies 2011). The Parrot AR. Drone is a quadrotor UAV that has both fixed forward- and vertical-facing cameras. An open-source API is also available. The AR. Drone is unique in that it is controllable only with Apple iOS devices (Parrot 2011).

Larger micro UAVs include the AirRobot AR100B®, which is a quadrotor micro UAV that includes an interchangeable payload. The *Pilot* for flight operations uses a hardware control interface that also contains a small display screen that can project real-time video when a camera is used as a payload. An API is available for the AR100B® for control (both flight and camera) customization; therefore, a *Mission Specialist* role could separately interact with the vehicle for data-gathering purposes on a separate laptop device (AirRobot 2011). The Draganflyer X series of rotor-based micro UAVs, produced by Draganfly Innovations, Inc., is controlled primarily by a hardware interface with limited touch screen interaction for flight and navigation. An onboard camera is also controllable using the same hardware interface, but video can be broadcast wirelessly to an HUD or a separate

display station, thereby allowing a *Mission Specialist* role the ability to complete reconnaissance tasks (Draganfly Innovations, Inc. 2011). Aeryon Labs has designed the Scout, a quadrotor vehicle with a hot-swappable payload that may include a gimbaled camera. The Aeryon Scout is capable of beyond line-of-sight operations and uses exclusively a touch-based software interface for flight and navigation control. Real-time video and image data transmission during the flight is available (to any wireless display device), and a *Mission Specialist* role could independently interact with the system to control the camera and complete reconnaissance tasks using a customized version of the touch screen interface (Aeryon Labs, Inc. 2011). Similarly, the Honeywell T-Hawk is capable of real-time video streaming from a controllable gimbaled camera to a custom pen-based laptop device (Honeywell 2011).

98.3.2 Small UAS Mission Specialist HMI

UAV human-robot teams in the small category often utilize coordination similar to that of micro UAV teams; however, there is not always colocation. As such, interface technologies for the *Mission Specialist* role appears in different varieties.

Oron-Gilad and Minkov (2010) provide two investigations of combat units utilizing a small UAV during the Second Lebanon War of 2006. Both studies indicated that the *Mission Specialist* role interacted with a handheld touch screen device. Additionally, there was a dedicated tablet laptop docked to the handheld device. The control panel had traditional hardware setup for interfacing, including a keyboard, trackball, and combination mouse/joystick. It was implied that both the *Pilot* and *Mission Specialist* role had to share the same handheld device to interact with the UAV. Figure 98.2 provides an example of a *Mission Specialist* interacting with a small UAV.

Commercially available small UAVs typically come with proprietary user interface technology. The fixed-wing Raven[®], produced by AeroVironment, has a common ground control system (GCS) that interfaces with all of the manufacturer's UAV systems. The control device consists of small handheld unit with dual joysticks for flight and navigation. A single small display screen is located in the center of the handheld device, allowing for visualization from onboard cameras. A hardware button-driven menu system allows for camera control. Due to the increased level of autonomy in this system, the *Pilot* and *Mission Specialist* roles overlap, leading to a single operator being essentially responsible for both roles (AeroVironment, Inc. 2011). AAI Corporation produces the fixed-wing Shadow line of Tier II UAVs. As with the Raven[®], the Shadow has a portable GCS. Two operators visualize the mission activities using independent large display screens. A full keyboard, joystick, and button pad allow for a *Mission Specialist* role to interface with the Shadow vehicle (AAI Textron Systems, Inc. 2011). Northrop Grumman produces the Fire Scout, which is uniquely a large rotor-based UAV. Flight and navigation by two operators is accomplished through a proprietary control system (CS). Consisting of four large display screens – two per operator – the CS displays payload imagery

Fig. 98.2 Example of a small UAS *Mission Specialist* interacting with an Aero Vironment Raven® UAV (Courtesy of the U.S. Department of Defense)



for reconnaissance that would fall under the purview of responsibilities for the *Mission Specialist* role. Interfacing is done through full keyboard, joystick, and button controls (Northrop Grumman 2011).

98.4 UAS Mission Specialist HMI Findings

In this section, a comparative analysis is provided of the *Mission Specialist* role and user interface technology found across both classes of UAVs. From the findings, it is observed that for micro and smaller UAS, there may exist a technology interaction conflict that could develop between the *Mission Specialist* role and the team member role responsible for flight and navigation.

For HMI purposes and though there are always exceptions, there are two distinct categories of human teams involved with a given micro or small UAS mission: (i) active and (ii) passive. Active human teams control all aspects of the UAV in the field. Micro UAVs, which are uniquely on-site, can be transported with the human team for complete launch, control, and landing by the human team. On-site active team roles may overlap. Passive human teams in the field (more common for a small UAS mission) tend to only interact with the UAV for activities such as taking reconnaissance imagery and adjusting navigation. For a passive UAS, the UAV may or may not be transported with the human team; transport, launch, control, and landing could be done via other persons at a separate location. For passive human teams, roles tend not to overlap. In terms of the UAS interaction technology (Tables 98.2 and 98.3), it is observed that there are two categories present for UAS human teams: (i) *Pilot and Navigator* centered (active) and (ii) *Mission Specialist* centered (passive). *Pilot and Navigator*-centered technology trends towards active control of the UAV. *Mission Specialist*-centered technology trends towards passive controlling of the camera, taking reconnaissance imagery, etc.

Table 98.2 Summary of hardware-based HMI used by a mission specialist role on UAS human-robot teams

| UAV | | platform name | Heads-up display | Isotonic joystick | Keyboard (full) | Mouse | Trackball | Pushbutton panel | Touch screen | Video display |
|-------|--|-----------------------------|------------------------------------|------------------------------------|----------------------------------|------------------------------|------------------------------|------------------------------------|------------------------------------|------------------------------------|
| Micro | | AirRobot ARI100B® | | AirRobot (2011) | | | | AirRobot (2011) | | AirRobot (2011) |
| | | Cooper & Goodrich Custom | | Cooper and Goodrich (2008) | | | | Cooper and Goodrich (2008) | | Cooper and Goodrich (2008) |
| | | Draganflyer X Series | Draganfly Innovations, Inc. (2011) | Draganfly Innovations, Inc. (2011) | | | | Draganfly Innovations, Inc. (2011) | Draganfly Innovations, Inc. (2011) | Draganfly Innovations, Inc. (2011) |
| | | Honeywell T-Hawk | | | Honeywell (2011) | Honeywell (2011) | | | Honeywell (2011) | Honeywell (2011) |
| | | iSENSYS IP-3 | Murphy et al. (2008) | Murphy et al. (2008) | | | | Murphy et al. (2008) | | |
| Small | | Like 90 T-Rex | | Murphy et al. (2008) | | | | Murphy et al. (2008) | | |
| | | Parrot AR. Drone | | | | | | | Parrot (2011) | Parrot (2011) |
| | | Skybotix Technologies CoaX® | | Skybotix Technologies (2011) | | | | Skybotix Technologies (2011) | | |
| | | AAI Shadow | | AAI Textron Systems, Inc. (2011) | AAI Textron Systems, Inc. (2011) | | | AAI Textron Systems, Inc. (2011) | | AAI Textron Systems, Inc. (2011) |
| | | AeroVironment Raven® | | AeroVironment, Inc. (2011) | | | | AeroVironment, Inc. (2011) | | AeroVironment, Inc. (2011) |
| | | Elbit Systems Skylark® | | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) |
| | | Northrop Grumman Fire Scout | | Northrop Grumman (2011) | Northrop Grumman (2011) | | | Northrop Grumman (2011) | | Northrop Grumman (2011) |

Table 98.3 Summary of software-based HMI used by a mission specialist role on UAS human-robot teams

| Group | UAV | | | | | | Synthetic overlay |
|-------|--|------------------------------------|--|--|------------------------------------|--|----------------------------------|
| | platform name | Aerial images | API customization | Menus (simple) | Menus (complex) | Real-time video | |
| Micro | AirRobot AR100B® | AirRobot (2011) | AirRobot (2011) | AirRobot (2011) | AirRobot (2011) | AirRobot (2011) | |
| | Cooper & Goodrich Custom | Cooper and Goodrich (2008) | | | | Cooper and Goodrich (2008) | |
| | Draganflyer X Series | Draganfly Innovations, Inc. (2011) | Draganfly Innovations, Inc. (2011) | Draganfly Innovations, Inc. (2011) | Draganfly Innovations, Inc. (2011) | Draganfly Innovations, Inc. (2011) | |
| | Honeywell T-Hawk | Honeywell (2011) | | | Honeywell (2011) | Honeywell (2011) | |
| | iSENSYS IP-3 | | | Murphy et al. (2008) | | Murphy et al. (2008) | |
| | Like 90 T-Rex | | | Murphy et al. (2008) | | Murphy et al. (2008) | |
| | Parrot AR. Drone Skybotix Technologies CoaX® | Parrot (2011) | Parrot (2011) Skybotix Technologies (2011) | Parrot (2011) Skybotix Technologies (2011) | Skybotix Technologies (2011) | Parrot (2011) Skybotix Technologies (2011) | Parrot (2011) |
| | AAI Shadow | AAI Textron Systems, Inc. (2011) | | | AAI Textron Systems, Inc. (2011) | AAI Textron Systems, Inc. (2011) | AAI Textron Systems, Inc. (2011) |
| | AeroVironment Raven® | AeroVironment, Inc. (2011) | AeroVironment, Inc. (2011) | AeroVironment, Inc. (2011) | AeroVironment, Inc. (2011) | AeroVironment, Inc. (2011) | AeroVironment, Inc. (2011) |
| | Elbit Systems Skylark® | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | Oron-Gilad and Minkov (2010) | |
| Small | Northrop Grumman Fire Scout | Northrop Grumman (2011) | Northrop Grumman (2011) | Northrop Grumman (2011) | Northrop Grumman (2011) | Northrop Grumman (2011) | Northrop Grumman (2011) |
| | | | | | | | |

Both categories of UAS involve some sort of hardware-based interface. In micro UAVs, hardware interfaces dominate. Active human team *Mission Specialists* tend to interface more with hardware-based technology that is designed predominantly for UAV control and navigation. Micro UAV *Mission Specialists* mostly use handheld controllers that control every aspect of the UAV. In both categories, the *Mission Specialist* was more likely to interact with a limited set of hardware-based input, including only a joystick and/or a predefined pushbutton panel. The actual technology involved for the active human team *Mission Specialist*, therefore, may take the form of a simple designated pushbutton system for camera angling, zoom, and image capture (e.g., the AirRobot AR100B® and DraganFlyer X series). Passive human team *Mission Specialists* tended to interface more with both hardware-based and software-based interaction technologies that are designed more for *Mission Specialist*-centered activities. *Mission Specialists* in this category tended to use some sort of standard computer (i.e., desktop or laptop) that has a design focus on taking reconnaissance imagery and, possibly, the ability to make navigation adjustments. In the small UAV category, the *Mission Specialist* was more likely to use a full suite of hardware input devices, including a joystick, keyboard, mouse, trackball, and some type of ancillary pushbutton panel. The reasoning behind these trends in micro and small UAV technology interaction is that in a field situation, where micro and small UAVs are colocated, mobility is a highly desired characteristic and this will obviously reflect in the interface technology.

Some form of visual display, whether integrated directly into the hardware controller or located at an adjacent or remote base station, was present in both of the UAV categories. For *Mission Specialists* in field-type scenarios, displays were generally similar in dimension (less than 14 in.). Some UAV systems found (e.g., the iSENSYS IP-3 and the Aeryon Scout) had HUD capabilities for real-time video feed visualization (although, presumably, any of the UAVs with a standard video signal output could, in theory, be integrated with commercially available HUD or other ancillary display technologies).

In terms of the software-based interactions of the *Mission Specialist* role, both categories were found to have at least one use of static imagery, real-time video, and synthetic overlaying of mission-related ancillary data, as part of the interfacing. Simple menu interaction (i.e., hardware-driven menus) was found to be more prevalent in both categories, while some complex menu interaction appeared in the small UAV categories. This is due to the greater complexity of interaction required by the larger systems and also ties to the greater variety of hardware input devices available to the *Mission Specialist* role. It should also be noted that platform-independent software was observed to take into account a greater variety of hardware inputs.

A logical consequence of these findings is that active human team *Mission Specialists* tend to have to fit within the technology design scope for a *Pilot and Navigator*. One primary concern with this type of user interfacing is that there would likely exist a direct interface interaction conflict with a separate *Pilot and Navigator* role if there was only a single controller device. An interface of this design almost necessitates that the Pilot and *Mission Specialist* roles overlap, which

is not necessarily the most recommended scenario in domains that may likely utilize micro and/or small UAV technology. In close proximity operations, such as urban or wilderness search and rescue, for example, a Pilot may need to focus solely on the flight controls rather than on *Mission Specialist* tasks for collision avoidance (Murphy et al. 2008; Cooper and Goodrich 2008).

98.5 Conclusions

In this survey, three human team member roles (*Flight Operations Director*, *Pilot and Navigator*, and *Mission Specialist*) were observed to trend across both operational categories of small and micro UAS. Consistent for both UAS categories was the unique responsibility of the *Mission Specialist* for data acquisition and interpretation during UAV operation. The survey offers three findings:

The HMI technology used by the *Mission Specialist* had the three primary characteristics: mobile, small, and visual. Mobility was observed in all of the interfaces that the *Mission Specialist* interacted with. Handheld controllers that could be carried and handled by one individual were the most common form of interaction device. Inputs were accomplished by the *Mission Specialist* through interactions with small controls, most commonly isotonic joysticks and pushbuttons on the handheld controllers. Interactive feedback to the *Mission Specialist* was visual and took the form of small video displays, graphical menus, and real-time video.

In the literature, the *Mission Specialist* either shared the same interface with the *Pilot and Navigator* role, was given a duplicate of the *Pilot and Navigator* interface, or was a passive viewer. No UAV system in the literature had a dedicated *Mission Specialist* interface; however, the availability of API customization in commercial UAV systems does allow for that possibility. Given that the *Mission Specialist* is a unique human team role, a distinct or different modality of HMI technology from that of the *Pilot and Navigator* would be expected. Therefore, existing interfaces, in general, do not support the functionality of the *Mission Specialist* role.

The responsibility of the *Mission Specialist* is for data acquisition and, often, interpretation. The possibility for direct manipulation of the imagery for verification, including extracting single static images and a video series for real-time playback while the flight continued to be recorded, appeared present in only one (Oron-Gilad and Minkov 2010) of the ten surveyed UAV systems (the full extent of which, however, was not clear). It should also be noted that this interaction was accomplished through the UAV interface with the broadest array of available HMI technology (isotonic joystick, keyboard, mouse, trackball, pushbutton panel, and touch screen). Interpretation support (such as synthetic overlays) and other software-initiated functions were present on five of the ten systems surveyed, but these options were only commercially documented. These observations suggest that there is a heavily reliance on the hardware-oriented interaction by current *Mission Specialists*. Current HMI for the *Pilot and Navigator*-centered interface could be

limiting the software capabilities that may improve *Mission Specialist* performance in small and micro UAS missions.

These three findings of this survey suggest that current *Mission Specialist* performance in small and micro UAS may be suboptimal due to the sharing of a single *Pilot and Navigator*-oriented interface or a reuse of the *Pilot and Navigator* interface. Further research is warranted to better understand the distinctions between the *Mission Specialist* and *Pilot and Navigator* roles and, specifically, what user interfaces are best suited for supporting the *Mission Specialist* role on the UAS human team.

Acknowledgments This material is based upon the work supported by the National Science Foundation under Grant No. IIS-1143713, EAGER: Shared Visual Common Ground in Human-Robot Interaction for Small Unmanned Aerial Systems.

References

- AAI Textron Systems, Inc., Shadow[®] Tactical Unmanned Aircraft Systems (TUAS) (2011) [Online]. Available http://www.aaicorp.com/products/uas/shadow_family.html
- AeroVironment, Inc., UAS: Raven[®] (2011) [Online]. Available http://www.avinc.com/uas/small_uas/raven
- Aeryon Labs, Inc., Scout product information sheet (2011) [Online]. Available <http://www.aeryon.com/products/avs.html>
- AirRobot, AirRobot AR100B[®] product information sheet (2011) [Online]. Available <http://www.airrobot-uk.com/air-robot-products.htm>
- M.J. Barnes, B.G. Knapp, B.W. Tillman, B.A. Walters, D. Velicki, Crew systems analysis of unmanned aerial vehicle (UAV) future job and tasking environments. United States Army Research Laboratory Report TR-2081, 2000
- W. Bierbaum, UAVs. Air and Space Power Journal Archives (1995) [Online]. Available <http://www.airpower.maxwell.af.mil/airchronicles/cc/uav.html>
- A. Burkle, F. Segor, M. Kollmann, Toward autonomous micro UAV swarms. *J. Intell. Robot. Syst.* **61**(1–4), 339–353 (2011)
- N.J. Cooke, R.A. Chadwick, Lessons learned from human-robotic interactions on the ground and in the air, in *Human-Robot Interaction in Future Military Operations*, ed. by M. Barnes, F. Jentsch (Ashgate, Farnham/Surrey/Burlington, 2010), pp. 355–372
- N.J. Cooke, H.K. Pederson, O. Connor, J.C. Gorman, D. Andrews, Acquiring team-level command and control skill for UAV operation, in *Human Factors of Remotely Operated Vehicles*, vol. 7, ed. by N.J. Cooke, H.L. Pringle, H.K. Pedersen, O. Connor (Elsevier, Amsterdam/Boston, 2006), pp. 285–297
- J. Cooper, M.A. Goodrich, Towards combining uav and sensor operator roles in UAV-enabled visual search, in *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction*, Amsterdam, 2008, pp. 351–358
- M.L. Cummings, P.J. Mitchell, Managing multiple UAVs through a timeline display, in *Proceedings of the American Institute of Aeronautics and Astronautics Infotech@Aerospace*, Arlington, Sept 2005
- M.L. Cummings, S. Bruni, S. Mercier, P.J. Mitchell, Automation architecture for single operator-multiple UAV command and control. *Int. C2 J.* **1**(2), 1–24 (2007)
- Draganfly Innovations, Inc., Draganflyer X series product information (2011) [Online]. Available <http://www.draganfly.com/industrial/products.php>

- M.A. Goodrich, J. Cooper, J.A. Adams, C. Humphrey, R. Zeeman, B.G. Buss, Using a mini-UAV to support wilderness search and rescue: practices for human-robot teaming, in *Proceedings of the IEEE International Workshop on Safety, Security, and Rescue Robots*, Rome, 2007, pp. 1–6
- A. Hobbs, Unmanned aircraft systems, in *Human Factors in Aviation*, ed. by E. Salas, D. Maurino, 2nd edn. (Elsevier, Amsterdam/Boston, 2010), pp. 505–531
- Honeywell, T-Hawk micro air vehicle product information (2011) [Online]. Available http://www.thawkmav.com/downloads/T-Hawk_Data_Sheet.pdf
- R. Hopcroft, E. Burchat, J. Vince, Unmanned aerial vehicles for maritime patrol: human factors issues. Australian Defense Science and Technology Organisation Report DSTO-GD-0463, 2006
- J.G. Manathara, P.B. Sujit, R.W. Beard, Multiple UAV coalitions for a search and prosecute mission. *J. Intell. Robot. Syst.* **62**(1), 125–358 (2011)
- R.R. Murphy, J.L. Burke, From remote tool to shared roles. *IEEE Robot. Autom. Mag.* (Special issue on New Vistas and Challenges for Teleoperation) **15**(4), 39–49 (2008)
- R.R. Murphy, K.S. Pratt, J.L. Burke, Crew roles and operational protocols for rotary-wing micro-UAVs in close urban environments, in *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction*, Amsterdam, 2008, pp. 73–80
- Northrop Grumman, Fire scout product information sheet (2011) [Online]. Available <http://www.as.northropgrumman.com/products/p6mq8bfirescout/index.html>
- Office of the United States Secretary of Defense, Unmanned aircraft system roadmap 2005–2030 [Online]. Available http://www.fas.org/irp/program/collect/uav_roadmap2005.pdf
- T. Oron-Gilad, Y. Minkov, Remotely operated vehicles (ROVs) from the bottom-up operational perspective, in *Human-Robot Interaction in Future Military Operations*, ed. by M. Barnes, F. Jentsch (Ashgate, Farnham/Surrey/Burlington, 2010), pp. 211–227
- Parrot, Parrot AR. Drone User Guide (2011) [Online]. Available <http://www.parrot.com/>
- C.E. Rash, P.A. LeDuc, S.D. Manning, Human factors in U.S. military unmanned aerial vehicle accidents, in *Human Factors of Remotely Operated Vehicles*, vol. 7, ed. by N.J. Cooke, H.L. Pringle, H.K. Pedersen, O. Connor (Elsevier, Amsterdam/Boston, 2006), pp. 117–131
- B. Schneiderman, C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5th edn. (Addison-Wesley, Boston, 2010), p. 4
- Skybotix Technologies, CoaX[®] coaxial helicopter product information sheet (2011) [Online]. Available <http://www.skybotix.com>
- A. Tsourdos, B.A. White, M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles* (Wiley, Chichester/Hoboken, 2011)
- A.P. Tsvaryanas, Human systems integration in remotely piloted aircraft operations. *Aviat. Space Environ. Med.* **77**(12), 1278–1282 (2006)
- United States Air Force, United States Air Force unmanned aircraft systems flight plan 2009–2047 [Online]. Available <http://www.defense.gov/dodcmssshare/briefingslide/339/090723-D-6570C-001.pdf>
- United States Army, U.S. Army unmanned aircraft systems roadmap 2010–2035 [Online]. Available <http://www.fas.org/irp/program/collect/uas-army.pdf>
- B. Walters, M.J. Barnes, Modeling the effects of crew size and crew fatigue on the control of tactical unmanned aerial vehicles (UAVs), in *Proceedings of the Winter Simulation Conference*, Orlando, 2000, pp. 920–924

M. A. Goodrich and M. L. Cummings

Contents

99.1 Introduction 2406

99.2 Supervisory Control of Multiple UASs 2406

99.3 Single Operator Management of Multiple UASs 2409

 99.3.1 A Meta-analysis of Previous Multiple UAV Studies 2410

99.4 The Knowledge-Based Loop: Managing Payloads 2415

 99.4.1 Reactive Navigation and the Chase Perspective 2417

 99.4.2 Systematic Navigation and the Map-Based Perspective 2419

99.5 Multi-operator, Multi-UAV Supervisory Control and Payload Management 2420

References 2421

Abstract

This chapter presents a synthesis of past work on Unmanned Aerial System (UAS) autonomy and user interface design and uses this synthesis to motivate emerging themes in near-term and far-term UAS developments. Issues include user interface design, UAS autonomy, UAS teaming, operator workload, and payload management.

This chapter is a fusion of work from two previously published papers by the authors, integrated and extended to give a fresh perspective on the issues:

- ML Cummings, S Bruni, S Mercier, PJ Mitchel (2007) Automation architecture for single operator, multiple UAV command and control. *Int C2 J* 1(2):1–24.
- M Draper, T Barry, G Calhoun, J Clark, M Draper, M Goodrich, C Jansen, J Kessens, F Kooi, A Lefebvre, S Murray, J Nelson, C Nielsen, G Osga,

M.A. Goodrich (✉)
Computer Science Department, Brigham Young University, Provo, UT, USA
e-mail: mike@cs.byu.edu

M.L. Cummings
Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: MissyC@mit.edu

A Oudenhuijzen, M Quigley, R Shively, B Simpson, R Stone, L van Breda, J van Delft, J van Erp (2006) Advanced operator interfaces. In: Uninhabited military vehicles (UMVs): human factors issues in augmenting the force. Chapter 6. NATO-RTO-TR-HFM-078. Neuilly-sur-Seine: NATO-RTO. An excerpt written by M. Goodrich from this chapter.

99.1 Introduction

Accomplishing a dangerous and complicated mission, while minimizing risks and costs, is a primary goal in many military situations. Because Unmanned Aerial Vehicles (UAVs) can provide information and/or munitions deployment without exposing a human to harm, they naturally support such a goal and are becoming ubiquitous. For example, many military intelligence, search, and reconnaissance (ISR) missions can benefit from using Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs). Civilian applications can also benefit from using UAVs including wilderness search (Goodrich et al. 2008), disaster damage assessment (Murphy et al. 2008), and infrastructure inspection.

With reduced radar signatures, increased endurance, and the removal of humans from immediate threat, unmanned (also known as uninhabited) aerial vehicles have become indispensable assets to militarized forces around the world, as proven by the extensive use of the Shadow and Predator in recent conflicts. Despite the absence of a crew onboard in any of these UAVs, human operators are still needed for supervisory control. Although less mature, civilian uses of UAVs also require supervisory control for complex operations like border patrol, agriculture monitoring, and disaster response.

UAVs require human guidance to varying degrees and often through several operators, which is what essentially defines a UAS (Unmanned Aerial System). For example, the Predator and Shadow each require a crew of two to be fully operational. However, it is often desirable to design systems such that the current many-to-one ratio of operators to vehicles can be inverted.

In order to develop UAV technologies to effectively support single operator, multiple UAV control, the focus should be on the creation of an organization of human operators, software agents, and UAVs such that mission effectiveness is maximized at a minimum cost. The remainder of this chapter examines the role of autonomy and user interface design on UAS design, with an emphasis on not only placing UAVs at the right locations at the right time but also on managing mission-critical payloads of those UAVs. The chapter begins by identifying key attributes of the supervisory control problem for UASs. Since military UAVs represent state of the art, the first half of this chapter use military systems to provide context.

99.2 Supervisory Control of Multiple UASs

The move from platform-centric warfare to network-centric warfare (NCW) represents a shift in the role of humans both in mission planning and actual operation.

As has already been evidenced in the development of fly-by-wire, highly automated aircraft and missile systems (such as Tomahawk and Patriot), military operators are less in direct manual control of systems but more involved in the higher levels of planning and decision making and remote operations.

This shift in control from lower-level skill-based behaviors to higher-level knowledge-based behaviors is known as human supervisory control (HSC). HSC is the process by which a human operator intermittently interacts with a computer, receiving feedback from and providing commands to a controlled process or task environment, which is connected to that computer (Fig. 99.1) (Sheridan 1992). All UAVs in the U.S. Department of Defense inventory operate at some level of supervisory control as depicted in Fig. 99.1.

Human supervisory control in UAV operation is hierarchical, as represented in Fig. 99.2. The innermost loop of Fig. 99.2 represents the basic guidance and motion control, which is the most critical loop that must obey physical laws of nature such as aerodynamic constraints for UAVs. In this loop, operator actions are focused only on the short term and local control (keeping the aircraft in stable flight), and generally human control in this loop requires skill-based behaviors that rely on automaticity (Rasmussen 1983).

The second loop in Fig. 99.2, the navigation loop, represents the actions that some agent, whether human or computer-driven, must execute to meet mission constraints such as routes to waypoints, time on targets, and avoidance of threat areas and no-fly zones. The outermost loop represents the highest levels of control, that of mission and payload management. In this loop, sensors must be

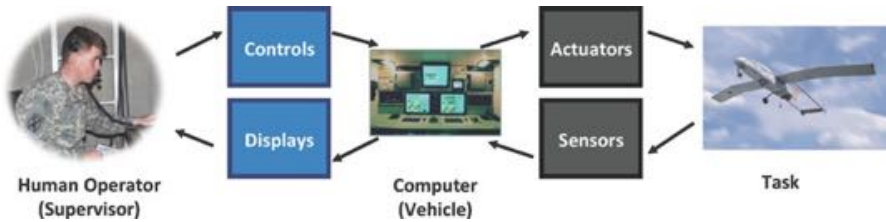


Fig. 99.1 Human supervisory control of a UAV

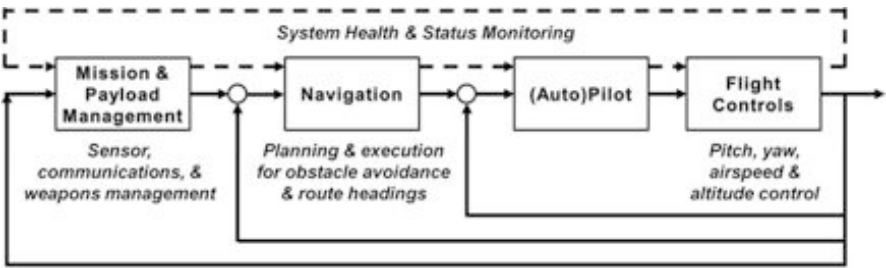


Fig. 99.2 Notional supervisory control loops for unmanned aerial vehicles

monitored and decisions made based on the incoming information to meet overall mission requirements. In this loop, decisions require knowledge-based reasoning that includes judgment, experience, and abstract reasoning that in general cannot be performed by automation.

Finally, the system health and status monitoring loop in Fig. 99.2 represents the continual supervision that must occur, either by a human or automation or both, to ensure that all systems are operating within normal limits. The control loop line is dashed as it represents a highly intermittent loop in terms of the human, i.e., if the human is engaged in another task, with the highest priority given to the innermost loop, health and status monitoring becomes a distant, secondary task.

From the human-in-the-loop perspective, if the inner loops fail, then the higher (outer) loops will also fail. The dependency of higher loop control on the successful control of the lower loops drives human limitations in control of a single and especially multiple UAVs. If humans must interact in the guidance and motion control loop (hand fly a UAV), the cost is high because this effort requires significant cognitive resources. What little spare mental capacity is available must be divided between the navigation and mission management control loops. Violations of the priority scheme represented in Fig. 99.2 have led to serious problems exemplified by numerous Predator crashes. When operators become cognitively saturated or do not correctly allocate their cognitive resources to the appropriate control loops in the correct priorities, they violate the control loop constraints, potentially causing catastrophic failure.

While Fig. 99.2 demonstrates supervisory control at the single vehicle level, Fig. 99.3 represents a notional system architecture that will be required for single operator control of multiple UASs. In order to achieve this futuristic system, operators will need to interact with an overall mission and payload manager while relegating routine navigation and motion control tasks to automation. The challenge in achieving effective management of multiple UAVs in the future is not only to determine if automation can be used to reduce workload but how and to what degree in each of the control loops in Figs. 99.2 and 99.3, as well as what kinds of decision support will be needed by operators given the high-workload environment.

One important consideration in developing the architecture depicted in Fig. 99.3 is the degree of centralization/decentralization of both the UAVs and the human. For example, in current single operator, multiple UAV operational paradigms, the operator independently controls all the vehicles in a serial fashion, and the vehicles do not work collaboratively to solve problems. Such an approach necessarily limits the number of vehicles the single operator can control since effectively the incoming tasks per vehicle represent a queuing network with a single server (the human) who has limited capacity (Cummings et al. 2007).

On the other end of the control spectrum are decentralized networks of UAVS with distributed autonomy across the vehicles such that the UAVs can locally reason independent of the network, while sharing information between the vehicles and a human supervisor at opportunistic times. Such control paradigms are task-based for the operator as opposed to vehicle-based for the centralized case (Clare and Cummings 2011). These architectures with significant advanced and increased

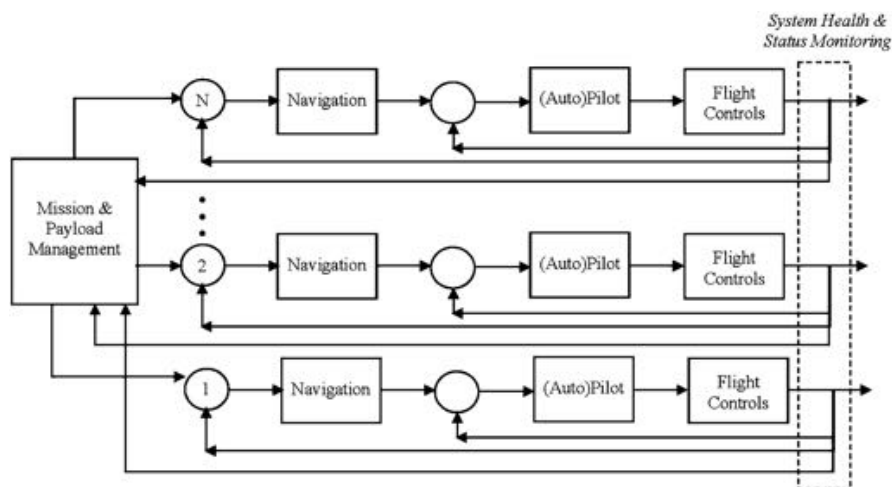


Fig. 99.3 Notional supervisory control loops for single operator, multiple UAV control

autonomy are what will enable significantly more UAVs to be controlled by one person, often referenced as swarm control.

99.3 Single Operator Management of Multiple UASs

Increasing the autonomy across the three control loops discussed previously (Figs. 99.2 and 99.3) is the critical architecture component for allowing one or a small team of operators to effectively control multiple UAVs. By increasing UAS autonomy, operator workload will theoretically be reduced as it could reduce the number of tasks for the operator, and it should reduce the level of interaction even at the highest levels of control in Figs. 99.2 and 99.3. For example, those UAVs that are flown in an autopilot mode relieve the operator from the manual flying tasks that require significant cognitive resources. This frees the operator to perform other critical tasks like mission planning and imagery analysis.

Higher levels of automation across the control loops depicted in Figs. 99.2 and 99.3 will be critical in achieving the single operator, multiple UAV control vision; but how, when, where, and what level of automation should be introduced are still difficult problems. For example, delegating navigation is necessary for organizations where a human must manage multiple UAVs. When the human must sequentially manage payloads, such delegation becomes even more critical.

While workload mitigation can occur through increasing automation, it can inadvertently cause higher workload as well as loss of situational awareness, complacency, and skill degradation (Parasuraman et al. 2000). For example, some UAV researchers have found that intermediate levels of management by consent (automation as an assistant to the operator) are preferable to manual or more fully automated

control (Ruff et al. 2002). However, management by consent means that the number of tasks could be high since operators must always be in the loop, potentially saturating operators, especially in the multiple UAV domain. Moreover, as has been shown in multiple UAV control research, operator performance can dramatically decrease under management by consent given increasing workload and various decision aids (Cummings and Guerlain 2007; Cummings and Mitchell 2008).

Given that an increasing number of tasks will have to be automated to achieve single operator control of multiple UAVs, particularly for swarm management, the question then becomes what to allocate to automation. Previous research has demonstrated that in the scheduling and execution of high-level tasks, of multiple UAVs, management by exception can improve operator performance (Cummings and Mitchell 2008). Management by exception occurs when automation decides to take an action based on some set of predetermined criteria and only gives operators a chance to veto the automation's decision.

While this control scheme can be effective in time-critical, high-risk domains like shutting down a near-critical nuclear reactor, in intentional, highly uncertain domains like command and control, it can be dangerous. Under this control scheme, operators are more likely to exhibit automation bias, a decision bias that occurs when operators become overreliant on the automation and do not check to ensure automated recommendations are correct (Mosier and Skitka 1996).

Automation bias was operationally seen in the 2004 war in Iraq when the U.S. Army's Patriot missile system, operating in a management-by-exception mode, engaged in fratricide, shooting down a British Tornado and an American F/A-18, killing three. The system was designed to operate under management by exception, and operators were given approximately 15 s to veto a computer solution. Unfortunately, the displays were confusing and often incorrect, and operators admittedly lacked training in the highly complex system (32nd Army 2003). Given the laboratory evidence that given an unreliable system, humans are still likely to approve computer-generated recommendations (Cummings 2004), it is not surprising that under the added stress of combat, Patriot operators did not veto the computer's solution. Automation bias is a significant concern for command and control systems, so it will be critical to ensure that when higher levels of automation are used, especially at the management-by-exception level, this effect is minimized.

99.3.1 A Meta-analysis of Previous Multiple UAV Studies

There have been numerous research studies published that have examined various aspects of multiple UAV control. A meta-analysis was performed across those studies that focused either explicitly on operator capacity or human supervisory control aspects of multiple vehicle control in order to determine any significant trends or lessons learned, particularly in regard to levels of automation and the control loops discussed above.

99.3.1.1 Previous Studies

One solution investigated by Dixon et al., to reduce UAV operator workload in the control of one or more small- to medium-sized UAVs, such as the Shadow, consisted of adding auditory and automation aids to support the potential single operator (Dixon et al. 2004). They showed that a single operator could theoretically fully control a single UAV both in terms of navigation and payload if appropriate offloading strategies were provided. For example, aural alerts improved performance in the tasks related to the alerts, but not others.

Conversely, it was also shown that adding automation benefited both tasks related to automation (e.g., navigation, path planning, or target recognition) and non-related tasks. However, their results demonstrate that human operators may be limited in their ability to control multiple vehicles that need navigation and payload assistance, especially with unreliable automation. These results are concordant with the single-channel theory, stating that humans alone cannot perform high-speed tasks concurrently (Welford 1952; Broadbent 1958). However, Dixon et al. propose that reliable automation could allow a single operator to fully control two UAVs in a centralized setting.

Reliability and the related component of trust is a significant issue in the control of multiple uninhabited vehicles. Ruff et al. (2002) found that if system reliability decreased in the control of multiple UAVs, trust declined with increasing numbers of vehicles but improved when the human was actively involved in planning and executing decisions. These results are similar to those found by Dixon et al. (2004) in that systems that cause distrust reduce operator capacity.

Ruff et al. (2002, 2004) determined that higher levels of automation actually degraded performance when operators attempted to control up to four UAVs. Results showed that in centralized settings, management by consent (in which a human must approve an automated solution before execution) was superior to management by exception (where the automation gives the operator a period of time to reject the solution). Management by consent appeared to provide the best situation awareness ratings, the best performance scores, and the most trust for controlling up to four UAVs.

Dunlap (2006) also subscribe to management by consent in their development of a distributed architecture to control multiple unmanned combat aerial vehicles (UCAVs). In this system, a UCAV plan is proposed by the automation, and the operator can either accept or reject the plan or submit an alternative. This recommendation can include both target assignments and routing. While they tested four, six, and eight UCAVs with increasing levels of environmental complexity, their final design limited the UCAV loadout at four. In one experiment, they noted that automation bias was a prevalent problem, stating the operators “had become attenuated to automatically accepting the usually correct proposals from the UCAVs,” which resulted in an increased kill rate for no-targets under the higher levels of automation.

In terms of actually predicting how many UAVs a single operator can control, there are only a few studies that examine this question, and they all focus on

the centralized architecture. Cummings and Guerlain (2007) showed that operators could experimentally control up to 12 Tactical Tomahawk Land Attack Missiles (TLAM) given significant missile autonomy. Operators only had to interact in the mission management loop, and all other loops were highly automated. In a UCAV setting, Cummings and Mitchell (2008) demonstrated that the number of UCAVs that a single operator can control is not just a function of the level of decision support automation but also the operational tempo and demands. Operators under low workload performed well regardless of the level of decision support, but under high workload, performance degraded. When considering operational and workload demands for a suppression of enemy air defenses mission, operator capacity was estimated at five UCAVs.

In a demonstration of the capabilities of a single operator attempting to control multiple Wide Area Search Munitions (WASMs), given high levels of autonomy across all control loops in Fig. 99.3 with only higher-order goal tasking for mission management, Lewis et al. (2006) posit that an operator can effectively control up to eight WASMs. The assumption is that the automation embedded in the vehicle's coordinates, without human intervention, specific tasks such as target detection and choice of the most appropriate member to execute the mission, which are capabilities that are not yet operational. The WASM study is similar to the Tactical Tomahawk study in that all flight-control and navigation functions are allocated to the automation alone, and the human intervenes for very high-level goal management.

Thus, there have been a cross section of studies that have examined operator performance and capacity in the control of multiple UAVs; however, it is not clear how many meaningful comparisons can be made across the different domains primarily because of two parameters: (1) what constitutes *control* and (2) what level of automation was used to aid the operators. In order to more directly compare these studies, the following section will discuss the scale on which comparisons can be made.

99.3.1.2 Level of Automation Trends

In this meta-analysis, the maximum number of UAVs that an operator effectively controlled in each study was extracted. It should be noted that in all of these reported studies, the control occurred in simulated test beds of medium fidelity and that all of these studies represent centralized control architectures. Approximate levels of automation (LOAs) across the control loop(s) from Fig. 99.3 were subjectively identified. While numerous levels and scales of automation and autonomy have been proposed (Parasuraman et al. 2000; Endsley and Kaber 1999; Wickens et al. 1998; Endsley 1995), the ten-level scale originally proposed by Sheridan and Verplank (1978) (SV-LOA) was chosen, as this is a commonly referenced taxonomy. Some of the categories in Table 99.1 were combined to reflect functional similarities. For example, levels 7–10 were combined since the human can take no action. Recognizing that different stages of information processing can be supported by automation (Parasuraman et al. 2000), the decision and action selection stage is represented in the assessment.

Table 99.1 Levels of automation

| SV-LOA | Our LOA | Automation description |
|----------|---------|---|
| 1 | I | The computer offers no assistance: human must take all decision and actions |
| 2 | II | The computer offers a complete set of decision/action alternatives |
| 3 | III | The computer offers a selection of decisions/actions |
| 4/5 | IV | The computer suggests one alternative and executes that suggestion if the human approves (management by consent) |
| 6 | V | The computer suggests one alternative and allows the human a restricted time to veto before automatic execution (management by exception) |
| 7/8/9/10 | VI | The human is not involved in the decision-making process, the computer decides and executes autonomously |

In Table 99.2, which presents a summary of these findings, the numbers of UAVs potentially controllable by a single human operator are referenced along with estimated levels of automation for each of the three control loops (MC – motion control inner loop, N – navigation, MM – mission management outer loop). It should be emphasized that the LOAs selected were approximate since they were both subject to interpretation and assigned post hoc from studies not originally intended to answer the research question. In addition, in many simulations, the LOA was not fixed so the range of LOAs was identified in these cases. In this comparison, an air traffic control (ATC) study was also included since it embodies many of the same principles of human supervisory control that are relevant to the control of multiple UAVs (Hilburn et al. 1997). Since air traffic controllers’ primary focus is safe navigation of aircraft, there is no associated mission management control loop.

Table 99.2 reveals interesting trends. Without explicitly discussing it in their respective studies, all researchers automated the inner motion control loop as depicted in Figs. 99.2 and 99.3. Thus, some form of autopilot was needed to relieve operator workload and free cognitive resources for higher loop control. To achieve the goal of one person controlling many UAVs, operators should only monitor the piloting/maneuvering of the vehicle, not do it themselves. However, this is a cultural problem more than it is a technological problem, as this technology is available today in all UAVs, but resisted in some communities, i.e., some organizations still insist that a human “fly” the vehicle instead of commanding it using various flight profiles.

Figure 99.4a demonstrates a general increasing trend in the number of vehicles an operator can control as a function of increasing automation in the navigation control loop of Fig. 99.2. Thus, given increasing navigation support and a fully autonomous flight-control system, operators can handle more UAVs when they do not have to attend to local and even global navigation concerns. The highest operator capacity was seen in the Tomahawk missile and WASM domains because, as they are one-way UAVs traveling at high speeds, there is little time for human intervention.

Table 99.2 Multiple UAV study comparison

| | Experiment | LOA | | | Max UV# |
|---|-----------------------------------|-----|------|--------|---------|
| | | MC | N | MM | |
| 1 | Dixon et al. (2005) (baseline) | VI | I | I | 1 |
| 2 | Dixon et al. (2005) (autopilot) | VI | IV | I | 2 |
| 3 | Dixon et al. (2005) (autoalert) | VI | I | IV | 2 |
| 4 | Ruff et al. (2002, 2004) | VI | IV–V | IV | 4 |
| 5 | Dunlap (2006) | VI | IV | IV | 4 |
| 6 | Cummings et al. (2007) | VI | IV | III–IV | 5 |
| 7 | Lewis et al. (2006) | VI | VI | IV–V | 8 |
| 8 | Cummings and Guerlain (2007) | VI | VI | IV | 12 |
| 9 | Hilburn et al. (1997) (ATC study) | VI | V | N/A | 11 |

When examining the mission and payload management control loop, [Fig. 99.4b](#) demonstrates the ability of operators to control more vehicles as they are provided with increasing automated decision support. It is interesting to note that given some automated navigation assistance and management-by-consent automation in the mission management loop, there is a convergence of operator capacity at 4–5 vehicles per operator. The next remarkable increase in operator capacity in centralized control settings (8–12 vehicles) is not seen until management by exception is introduced in the mission management or navigation loops. These increased levels of automation will be critical for increased operator capacity since, as previously discussed, if operators are required to attend to local navigation functions, they simply do not have the cognitive resources to successfully attend to all of the tasks in the mission and payload management loop.

As mentioned previously, to truly allow one operator the ability to control significantly more vehicles than the studies represented in [Table 99.2](#), the control architecture must shift from centralized, vehicle-based to decentralized, task-based. The decentralized approach leaves the human just in the knowledge-based mission management loop and offloading the lower two loops in [Figs. 99.2](#) and [99.3](#) to the automation. Thus, the limiting factor is not the number of vehicles an operator is controlling, but rather the number of tasks generated from each vehicle. The higher the autonomy and reasoning across a network of UAVs, the fewer tasks the team of vehicles generates.

In recent experiments, it was demonstrated that with a task-based, decentralized control system, doubling operator task load led to an operator workload increase of only ~50 %. Also in these studies, operator workload never approached unmanageable thresholds despite the doubling of task load. Thus, a task-based, decentralized control system may be robust to high task load situations, in effect allowing an operator to control theoretically limitless numbers of UAVs as long as they are generating, as a group, a manageable set of tasks (Clare and Cummings [2011](#)).

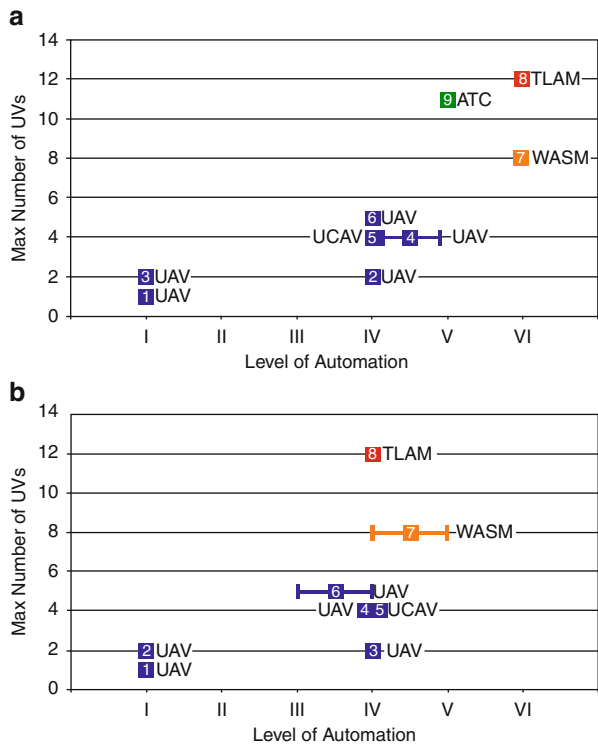


Fig. 99.4 Max number of UAVs vs. LOAs for the (a) navigation loop and (b) mission management loop

99.4 The Knowledge-Based Loop: Managing Payloads

It can be difficult to operate a remote UAS when an operator is required to spend most of his or her time in the mission and payload management outer loop in Fig. 99.2. Such tasks are typically knowledge-based and require judgment and reasoning, not easily codified in an algorithm. One such task that causes high workload in today's UAV operations is the surveillance task such as watching video feed. The perceptual problem associated with this task is the "limited angular view associated with many remote vision platforms creates a sense of trying to understand the environment through what remote observers often call a 'soda straw'" (Woods et al. 2004).

Conventional, workstation-based UAS interfaces do not integrate multiple sources of information into a coherent representation. Many interfaces include a map, UAS platform operations, and flight-control parameters in separate locations on the screen. Most systems deployed today show the footprint of the camera, but this is

often the extent of the integration. For operators using conventional displays, the cognitive workload associated with just getting enough awareness to control flight and navigation (i.e., the inner loops in Fig. 99.2) can be high enough that it is difficult to interpret imagery, plan strategies, and do other higher-level tasks that are relevant for the mission. This problem is frequently solved by adding more humans to the team who are responsible for mission-level issues (Burke and Murphy 2004) although, as noted in Sect. 99.3.1.1, offloading tasks to automation can theoretically enable an operator to manage both navigation and payload management.

A better approach to this problem is to represent real sensor information in an ecological way (Vicente et al. 1995). In the robotic domain, one important and useful ecological technique is the use of mixed-reality interfaces. Such interfaces combine real data with virtual elements and range from augmented reality to augmented virtuality interfaces (Milgram and Kishino 1994). An example of such interfaces, taken from Cooper (2007), is shown in Fig. 99.5. In this interface, satellite imagery and terrain maps are fused to create a virtual representation of the world. The UAV is projected into this virtual representation, and waypoints, compass, and other flight information are displayed within this virtual representation. Perhaps most importantly, video information is projected into the virtual world, giving context and stability for information detected in the camera. Most systems currently deployed project a sensor footprint (though perhaps without video) in this way, and most also provide separate video feeds on larger, higher-resolution displays to support video analysis.

Because a mixed-reality display is not constrained to a single perspective (e.g., pilot's perspective and payload operator's perspective), there are a number of different ways for information to be presented in a mixed-reality display. This chapter now briefly describes the *chase perspective* and the *map-based perspective*. Each technique is appropriate for certain types of tasks and modes of interaction between the human and a UAV.



Fig. 99.5 A mixed-reality display fusing control, navigation, and sensor information

99.4.1 Reactive Navigation and the Chase Perspective

The chase perspective is illustrated in Fig. 99.5. This perspective presents sensor information in a way that supports locomotion and is a typical representation used in racing games because it allows the direct perception of the relationship between the vehicle and the afforded directions of vehicle movement. An additional example of the chase perspective is shown in Fig. 99.6 (Quigley et al. 2004). In this display, a virtual UAV is included in the display to represent the pose of the UAV relative to the ground. This virtual UAV is overlaid on the video image received from the UAV and allows the operator to directly perceive the attitude of the aircraft with respect to the ground. (In the figure, two virtual UAVs are shown; one indicates the actual pose of the UAV as received from telemetry, and the other indicates the commanded pose of the UAV.) The chase perspective shown in Fig. 99.6 is taken from an interface that runs on a 7-in. or smaller display.

Note that the chase perspective for the UAV is earth-centered rather than pilot-centered; in the human factors literature, these differences of perspective are known as outside in and inside out, respectively. When the operator is on the ground, banking right is not accompanied by a pilot-perceived change in the earth's horizon nor is it accompanied by other vestibular cues. Since the operator is on the ground, the chase perspective adopts a ground-based perspective wherein a bank command is depicted by having the virtual UAV dip its wing in the commanded direction, thus matching the model of the aircraft to the mental model of the remote operator.

It is possible to take this ground-centered perspective a step further. Although the technology discussed in the next few paragraphs is a bit outdated, it is still relevant since it can be directly applied to controlling UAVs using tablet-based displays when those tablets are equipped with inertial sensing, as in “flying by iPad,” especially given the increasing use of smart phone and tablet devices in control applications for UAVs (Jackson et al. 2011; Cooper and Goodrich 2005). Since a fixed-mount

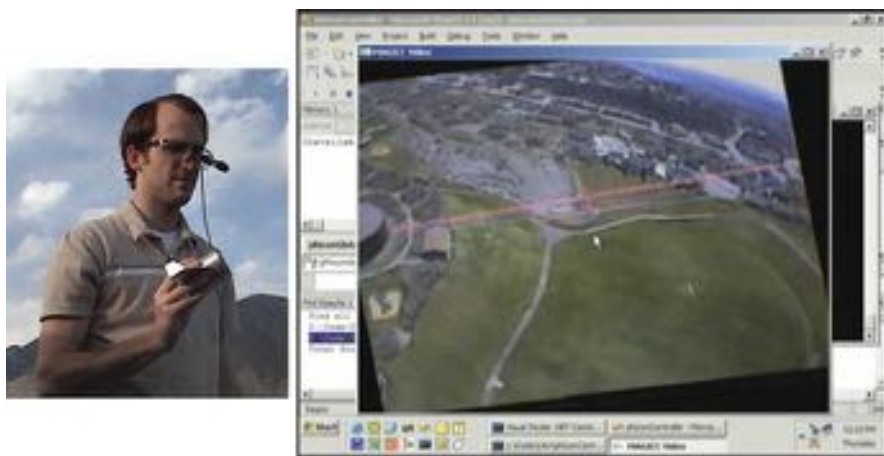


Fig. 99.6 Integrating a chase perspective with flight control

camera, typical of cameras mounted on micro aerial vehicles, rotates when the UAV banks, the operator must switch from a ground perspective to the UAV perspective in order to interpret video. This switch might be a cause of cognitive workload because the ground-based operator must interpret rotations in the video caused by a banking UAV.

An interface that makes both video and bank angle have a ground-centered reference frame is shown in Fig. 99.6. This interface is built for a control device called the PhyCon (for Physical Icon) (Quigley et al. 2004). Rather than using a handheld computer to issue commands to the UAV, a physical model of the UAV is used. When the operator banks the model, the actual UAV also banks. The PhyCon is almost certainly not viable for practical systems, but it does reinforce the idea that the inertial sensors from modern smart phones and tablets can be used for human control of UAVs. Although it is somewhat difficult to see in the figure, the pose of the aircraft is projected onto the video from a ground-centered reference frame using the chase perspective. This is a type of mixed-reality interface (Milgram and Kishino 1994). (In practice, there are actually two virtual depictions of the UAV. The first virtual UAV is depicted using the actual telemetry from the UAV. The second virtual UAV is depicted using the commanded pose from the PhyCon. Having both of these projected into the augmented reality display allows the operator to see that the actual UAV is responding appropriately to the commanded pose.)

The video feed is digitized and displayed on a computer (in this case, a laptop which presents the video through an eyeglass-mounted display). Prior to presenting the imagery, the telemetry from the UAV is used to rotate the image so that the horizon stays approximately level. This is depicted in Fig. 99.7. Rotating the image so that the horizon stays level means that both the video and the UAV attitude are depicted in a ground-relative reference frame.

An important lesson from Cooper (2007) is that chase perspectives are best suited to so-called reactive searchers. In a reactive search, an operator is not systematically



Fig. 99.7 North-up perspective

navigating through a search pattern or a series of waypoints. Instead, the operator is using real-time video information to directly control the UAV so as to track a moving target, follow a trail or terrain feature, or inspect a pipeline or convoy. Chase-based perspectives are useful in such domains because they create a tight coupling between flight dynamics and video information. Figure 99.6 enables this tight connection by changing the orientation of the UAV in response to video information, and Fig. 99.5 enables this tight connection by allowing the operator to control a controllable, floating waypoint (labeled the “carrot”) that the operator can reactively control.

99.4.2 Systematic Navigation and the Map-Based Perspective

The chase perspective primarily supports reactive control. When it is necessary to operate at the level of navigation or planning, it is often useful to have a map of some sort. For example, a common UAV mission is to search an area for a missing person or a target. Issuing commands for these searches and depicting the progress of these searches may be easier for the operator if a map-centered reference frame is used (Plumlee and Ware 2003), and in a direct comparison of a chase perspective and map-based perspective it was found that the map-based perspective was more useful for systematic search (Cooper 2007). The task is to present map information in a useful way and then to integrate the video into the map using this map-centered perspective.

Figure 99.7 depicts a mixed-reality display that integrates a virtual map with video from a robot (Cooper 2007). The video is depicted in this virtual world in such a way that video, map, and UAV pose information are simultaneously visible. There are a number of desirable features of such an interface, including (a) the ability to determine what has been searched and what needs to be searched, (b) the ability to perceive how the robot is oriented with respect to landmarks in the world, most importantly, a fixed north-up orientation, and (c) the ability to augment map information with icons or other semantic labels.

The ability to determine what has been searched and what needs to be searched has been shown to be very useful in wilderness search and rescue (Morse et al. 2010) as has the ability to keep a stabilized camera view in a north-up orientation (Morse et al. 2008). Map-centered displays, with either a north-up or linked orientation (Plumlee and Ware 2003), can be augmented with mission-specific symbology.

It is important to note that map-based interfaces have been used to construct augmented reality displays in aviation. These displays, which may be either heads-up or head-down and which may be retrofitted to older aircraft (Prinzel et al. 2004), are referred to as *synthetic vision* displays (Schnell et al. 2004). Several human factors studies have been conducted, many showing that there is an increase in navigation-related situation awareness with negligible loss in aviation-related situation awareness (Alexander and Wickens 2005), presumably because a greater field of view and subsequent sense of realism can be obtained with such displays.

99.5 Multi-operator, Multi-UAV Supervisory Control and Payload Management

The previous sections explored how autonomy and the user interface affected the state of the art in human supervisory control of single and multiple UASs. In this section, emerging trends are summarized that will likely impact the next generation of UASs.

The first emerging area follows immediately from the increases in complexity that occur when a human is required to manage multiple UASs. *Decision support systems* (DSSs) will be needed (a) to help humans manage attention between different vehicles, sensors, and mission parameters and (b) to help humans allocate and schedule resources so that assets are used efficiently. In single operator, multiple UAV control as depicted in Fig. 99.3, DSSs will be needed to assist primarily in the mission payload management task, as well as possibly lower-level tasking such as navigation tasks.

Given that the large majority of military UAVs, as well as likely future civilian UAV applications, are surveillance missions, an important question is whether a single operator can manage multiple video feeds that will occur in such multi-UAV surveillance missions. Such missions are only possible with high degrees of automation in the lower-level control loops in Fig. 99.3, as well as also in the mission management loop. Such automation could take the form of automated target detection, which is currently a very active area of basic research but with no major breakthroughs as of yet for airborne systems.

Another form of automated assistance in such settings could take the form of scheduling tools that prompt operators when to switch their attention across different UAVs under their control. Such DSSs rely on developing operator models that also consider the operational context of the vehicles in order to make recommendations to operators for not only when to switch but to which UAV or task that needs attention. While such attention management is key to efficient single operator–multiple UAV control, developing operator models that are robust in the inherently stochastic world of command and control is difficult (i.e., Bertuccelli et al. (2010)), so this is also an open area of current research.

The second emerging area is one of organization, particularly *organizational issues that arise when multiple humans share management responsibilities* for multiple UASs. The philosophy behind work in this area is that future operational models of human-UAS teams will have N humans manage M vehicles where N is much less than M . One key organizational issue is whether specific humans are assigned to fixed UASs or whether the humans can interact with any UAS according to mission demands (Mekdeci and Cummings 2009; Lewis et al. 2010). A second key organizational issue is how humans can provide mutual support to each other so that resulting behavior exploits the advantages of highly effective teams without invoking the negative behaviors (such as social loafing) that can arise when responsibility is distributed across an organization (Whetten et al. 2010).

The third emerging theme is the need to form a *common operational picture* when multiple sensor payloads are distributed across a wide geographical area (Sirak

2007; Feitshans et al. 2008; Velagapudi et al. 2008). This theme builds on current operational models where specific humans are responsible for controlling unmanned air and ground platforms but where other humans can benefit from information being returned from those platforms. This is particularly important in military applications where micro-UAVs might be used by a small squad of soldiers and large UAVs might be used as a strategic asset. Although different groups of humans might be responsible for controlling operational, tactical, and strategic UAVs, each group of humans might benefit in the role of so-called information consumers (Goodrich and Schultz 2007) of a common operational picture.

The fourth emerging theme is an extension of the third theme and seeks to automatically allocate UAV resources by providing information to and requesting information about a particular mission. Although this has been explored in other contexts, it is easily illustrated in wilderness search and rescue where the path of a sensor footprint may be automatically planned as the result of (a) search experts providing information about the likely influences on missing person movement, (b) incident commanders providing information about the relative difficulty of air and ground resources detecting a missing person, and (c) the UAV technical search specialist providing information about where the UAV should begin and end a search but allowing the UAV to optimally search the area in between (Lin et al. 2010).

References

- 32nd Army Air and Missile Defense Command, *Patriot Missile Defense Operations during Operation Iraqi Freedom*, U. S. Army, Ed. Washington, DC, 2003
- A.L. Alexander, C.D. Wickens, Synthetic vision systems: flightpath tracking, situation awareness, and visual scanning in an integrated hazard display, in *Proceedings of the 13th International Symposium on Aviation Psychology*, Oklahoma City, Oklahoma, 2005
- L. Bertuccelli, N. Pellegrino, M.L. Cummings, Choice modeling of relook tasks for UAV search missions, in *IEEE American Control Conference*, Baltimore, MD, June 2010
- D.E. Broadbent, *Perception and Communication* (Pergamon, Oxford, 1958)
- J.L. Burke, R.R. Murphy, Human-robot interaction in USAR technical search: two heads are better than one, in *Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication* Kurashiki, Okayama, Japan, 2004
- A.S. Clare, M.L. Cummings, Task-based interfaces for decentralized multiple unmanned vehicle control, *Proceedings of AUVSI 2011: Unmanned Systems North America*, Washington DC, Aug 2011
- J.L. Cooper, M.A. Goodrich, Portable mini-UAV control interfaces for non-pilots. Presented at the 2005 human factors of UAVs workshop, Mesa, Arizona, 2005
- J.L. Cooper, Supporting Flight Control for UAV-Assisted Wilderness Search and Rescue through Human Centered Interface Design, Computer Science Department Master's Thesis, Brigham Young University, Provo, Utah, 2007
- M.L. Cummings, Automation bias in intelligent time critical decision support systems. Presented at AIAA 3rd intelligent systems conference, Chicago, 2004
- M.L. Cummings, S. Guerlain, Developing operator capacity estimates for supervisory control of autonomous vehicles. *Human Factors* **49**, 1–15 (2007)
- M.L. Cummings, P.J. Mitchell, Predicting controller capacity in remote supervision of multiple unmanned vehicles. *IEEE Syst. Man Cybern. A* **38**(2), 451–460 (2008)

- M.L. Cummings, C.E. Nehme, J. Crandall, Predicting operator capacity for supervisory control of multiple UAVs, in *Innovations in Intelligent Machines*, ed. by J.S. Chahl, L.C. Jain, A. Mizutani, M. Sato-Ilic. Studies in Computational Intelligence, vol. 70 (Springer, Berlin/Heidelberg, 2007)
- S. Dixon, C.D. Wickens, D. Chang, Unmanned aerial vehicle flight control: false alarms versus misses. Presented at Humans Factors and Ergonomics Society 48th Annual Meeting, New Orleans, 2004
- S. Dixon, C. Wickens, D. Chang, Mission control of multiple unmanned aerial vehicles: a workload analysis. *Human Factors* **47**(3), 479–487 (2005)
- R.D. Dunlap, The evolution of a distributed command and control architecture for semi-autonomous air vehicle operations. Presented at Moving Autonomy Forward Conference, Grantham, UK, 2006
- M.R. Endsley, Toward a theory of situation awareness in dynamic systems. *Human Factors* **37**, 32–64 (1995)
- M.R. Endsley, D.B. Kaber, Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics* **42**, 462–492 (1999)
- G.L. Feitshans, A.J. Rowe, J.E. Davis, M. Holland, L. Berger, Vigilant spirit control station (VSCS): the face of counter, in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Honolulu, Hawaii, 2008
- M.A. Goodrich, B.S. Morse, D. Gerhardt, J.L. Cooper, M. Quigley, J.A. Adams, C. Humphrey, Supporting wilderness search and rescue using a camera-equipped mini UAV. *J. Field Robot.* **25**, 89–110 (2008)
- M.A. Goodrich, A.C. Schultz, Human-robot interaction: a survey. *Found. Trends Human Comput. Interact.* **1**, 203–275 (2007)
- B. Hilburn, P.G. Jorna, E.A. Byrne, R. Parasuraman, The effect of adaptive air traffic control (ATC) decision aiding on controller mental workload, in *Human-Automation Interaction: Research and Practice* (Lawrence Erlbaum, Mahwah, 1997), pp. 84–91
- K.F. Jackson, P.W. Quimby, M.L. Cummings, MAV-VUE Outdoor Usability Analysis, (HAL2011-01), MIT Humans and Automation Laboratory, Cambridge, MA, 2011
- M. Lewis, J. Polvichai, K. Sycara, P. Scerri, Scaling-up human control for large UAV teams, in *Human Factors of Remotely Operated Vehicles*, ed. by N. Cooke, H. Pringle, H. Pedersen, O. Connor (Elsevier, New York, 2006), pp. 237–250
- M. Lewis, H. Wang, S. Chien, P. Scerri, P. Velagapudi, K. Sycara, B. Kane, Teams organization and performance in multi-human/multi-robot teams, in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Istanbul, Turkey, 2010
- L. Lin, M. Roscheck, M.A. Goodrich, B.S. Morse, Supporting wilderness search and rescue with integrated intelligence: autonomy and information at the right time and the right place, in *Proceedings of the 24th AAAI Conference on Artificial Intelligence: Special Track on Integrated Intelligence*, Atlanta, Georgia, 2010
- B. Mekdeci, M.L. Cummings, Modeling multiple human operators in the supervisory control of heterogeneous unmanned vehicles, in *Proceedings of the 9th Workshop on Performance Metrics for Intelligent Systems*, (ACM, New York, 2009)
- P. Milgram, F. Kishino, A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst.* **E77-d**, 1321–1329 (1994)
- B.S. Morse, C.H. Engh, M.A. Goodrich, UAV video coverage quality maps and prioritized indexing for wilderness search and rescue, in *Proceedings of the 5th ACM/IEEE RAS International Conference on Human-Robot Interaction*, Osaka, Japan, 2010
- B.S. Morse, D. Gerhardt, C. Engh, M.A. Goodrich, N. Rasmussen, D. Thornton, D. Eggett, Application and evaluation of spatiotemporal enhancement of live aerial video using temporally local mosaics, in *Proceedings of CVPR*, Anchorage, Alaska, 2008

- K.L. Mosier, L.J. Skitka, Human decision makers and automated decision aids: made for each other? in *Automation and Human Performance: Theory and Applications, Human Factors in Transportation*, ed. by R. Parasuraman, M. Mouloua (Lawrence Erlbaum, Mahwah, 1996), pp. 201–220
- R.R. Murphy, E. Steimle, C. Griffin, C. Cullins, M. Hall, K. Pratt, Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane Wilma. *J. Field Robot.* **25**, 164–180 (2008)
- R. Parasuraman, T.B. Sheridan, C.D. Wickens, A model for types and levels of human interaction with automation. *IEEE Trans. Syst. Man Cybern. A* **30**, 286–297 (2000)
- M. Plumlee, C. Ware, An evaluation method for linking 3D views, in *Proceedings of the 2003 Symposium on Interactive 3D Graphics*, Monterey, California, 2003
- L.J. Prinzel, J.R. Comstock Jr., L.J. Glaab, L.J. Kramer, J.J. Arthur, The efficacy of head-down and head-up synthetic vision display concepts for retro- and forward-fit of commercial aircraft. *Int. J. Aviat. Psychol.* **14**, 53–77 (2004)
- M. Quigley, M.A. Goodrich, R. Beard, Semi-autonomous human-UAV interfaces for fixed-wing mini-UAVs, *Proceedings of IROS*, Sendai, Japan, 2004
- H.A. Ruff, S. Narayanan, M.H. Draper, Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. *Presence* **11**, 335–351 (2002)
- H.A. Ruff, G.L. Calhoun, M.H. Draper, J.V. Fontejon, B.J. Guilfoos, Exploring automation issues in supervisory control of multiple UAVs. Presented at 2nd human performance, situation awareness, and automation conference (HPSAA II), Daytona Beach, FL, 2004
- J. Rasmussen, Skills, rules, and knowledge; signals, signs, and symbols, and other distractions in human performance models. *IEEE Trans. Syst. Man Cybern.* **SMC-13**, 257–266 (1983)
- T. Schnell, Y. Kwon, S. Merchant, T. Etherington, Improved flight technical performance in flight decks equipped with synthetic vision information system displays. *Int. J. Aviat. Psychol.* **14**, 79–102 (2004)
- T.B. Sheridan, W.L. Verplank, Human and computer control of undersea teleoperators, MIT, Cambridge, Man-Machine Systems Laboratory Report, 1978
- T.B. Sheridan, *Telerobotics, Automation and Human Supervisory Control* (MIT, Cambridge, 1992)
- M. Sirak, DARPA, Northrop Grumman move into next phase of UAV control architecture. *Defense Daily*, Nov 29, 2007, Retrieved 2009-03-16
- P. Velagapudi, P. Scerri, K. Sycara, H. Wang, M. Lewis, J. Wang, Scaling effects in multi-robot control, *Proceedings of IROS*, Nice, France, 2008
- K.J. Vicente, K. Christoffersen, A. Pereklita, Supporting operator problem solving through ecological interface design. *IEEE Trans. Syst. Man Cybern.* **25**, 529–545 (1995)
- A.T. Welford, The psychological refractory period and the timing of high-speed performance – a review and a theory. *Br. J. Psychol.* **43**, 2–19 (1952)
- J.M. Whetten, M.A. Goodrich, Y. Guo, Beyond robot fan-out: towards multi-operator supervisory control, in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Istanbul, Turkey, 2010
- C.D. Wickens, S.E. Gordon, Y. Liu, *An Introduction to Human Factors Engineering* (Longman, New York, 1998)
- D.D. Woods, J. Tittle, M. Feil, A. Roesler, Envisioning human-robot coordination in future operations. *IEEE Trans. Syst. Man Cybern. A* **34**, 749–756 (2004)

Julie A. Adams

Contents

100.1 Introduction 2426

100.2 Analysis Techniques 2426

 100.2.1 Task Analysis 2426

 100.2.2 Work Analysis 2429

 100.2.3 Information Analysis 2432

 100.2.4 Combining Analysis Techniques 2434

100.3 Conclusion 2437

References 2438

Abstract

Unmanned aerial systems are complex tools requiring thoughtful designs that account for the platform, payloads, and human users as a holistic system. Often, the focus is first placed on designing the platform or payload, and the human element is considered later. Unfortunately, such system design approaches do not properly consider the human cognitive limitations and do not design the system to support the human’s tasks and decision-making. This chapter provides an overview of cognitive analysis methods that can be integrated into the early system design process to account for the human element. Specifically, this chapter provides an overview of task analysis, work analysis, and information flow analysis methods along with an overview of how individual methods from the different analysis classes have been combined. Specific examples of applying these analyses methods to unmanned aerial systems, ranging from potential futuristic systems to research systems to deployed systems, are also reviewed.

J.A. Adams
Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville,
TN, USA
e-mail: julie.a.adams@Vanderbilt.Edu

100.1 Introduction

The proliferation of unmanned aerial systems (UAS) with varying onboard capabilities has led to widely varying human–UAS interaction systems. For example, the Predator UAS, operated by the United States Air Force, essentially re-creates a traditional cockpit inside a ground-based control station, complete with stick-and-rudder controls. There exist research-based UAS architectures that measure atmospheric composition by flying preprogrammed flight paths to obtain precise scientific data (Goetzendorf-Grabowski et al. 2006). Finally, UAS systems exist that require the human to perform line-of-sight remote control with little or no onboard autonomy (Murphy et al. 2008). Thus, UAS platforms span the spectrum of autonomy levels (Endsley and Kaber 1999; Parasuraman et al. 2000) that directly impact the system capabilities and the human’s task of interacting with the system (Sheridan 1992, Sheridan 2002).

Given the largely varying UAS system capabilities, it is clear that the overall system design places high demands on the human operators. Human–UAS interaction design is fundamentally interrelated with UAS autonomy design and is essentially a multidimensional trade-off between application requirements, precision, response time, neglect tolerance, portability, and team size (Crandall et al. 2005; Quigley et al. 2005). The system designers must balance these trade-offs appropriately.

The capabilities of a particular combination of airframe, autopilot, and control algorithm delineate the set of affordances that frame the human–UAS system and deployment constraints (Gibson 1979; Norman 1988). These affordances implicitly define the possible available interaction control modalities and the associated cognitive demands (Adams et al. 2007). Each UAS application has a unique set of constraints that define the feasible interaction space, and analyses can be conducted regarding the characteristics of various system capabilities. Application constraints, especially human factors considerations, need to be incorporated into the system design in order to eliminate infeasible interaction modes and to promote appropriate system design (Adams et al. 2007).

Many approaches exist for deriving human centric system requirements and specifications. This chapter provides an overview of analysis methods, including cognitive task analysis, cognitive work analysis, and information flow analysis along with the benefits of combining the analysis methods and examples supported by the literature.

100.2 Analysis Techniques

100.2.1 Task Analysis

Task analysis is the process of identifying and examining the tasks to be performed by users when interacting with a system (Kirwan and Ainsworth 1992). Task analysis encompasses a range of techniques that capture actions and cognitive processes necessary for a human attempting to achieve a goal while using a system.

The objective is to effectively and efficiently integrate the human into the system in a systematic way. Cognitive task analysis extends task analysis techniques in order to better understand the humans' knowledge, thought processes, and goal structures that directly impact the task performance (Chipman et al. 2000; Hollnagel 2003). Overall, cognitive task analysis seeks to understand the cognition required by the human user to complete tasks and how to turn that understanding into tools that assist the human (Crandall et al. 2006).

A broad range of cognitive and noncognitive task analysis methods exist. Each method has a set of advantages, and disadvantages and many practitioners offer differing opinions of which methods are the best. Many sources review the various (60+) analysis methods; thus, this chapter does not recount all the methods (Annett and Stanton 2000; Crandall et al. 2006; Diaper and Stanton 2004; Hollnagel 2003; Kirwan and Ainsworth 1992; Schraagen et al. 2000).

The ability for humans to supervise, command, and interact with unmanned aerial systems is cognitively demanding. Often, UASs are deployed in highly dynamic and uncertain domains. If the UAS is remotely controlled using line of sight, then the human pilot must focus on the vehicle itself to successfully complete missions. If the human pilot does not have direct line of sight, but rather must use a human-machine interface or control station, then the human is constrained by the information system displays. The ability to supervise, command, and interact with the UAS becomes increasingly difficult when the system incorporates additional personnel, such as a sensor operator, distributed personnel, or a mixed airspace that incorporates both manned and unmanned systems.

When choosing a task analysis method for UAS system design, the choice of method will likely depend upon a number of criteria. If the focus of the analysis effort is on physical vehicle design, then the choice of analysis will likely differ from the case in which the analysis focuses on the human aspects as well.

The ability to provide situational awareness to remotely located human operators of UASs and their associated but distributed stakeholders is critical to mission success. UAS deployments tend to fall into two broad categories, those completed within the line of the pilots' sight and those that are distributed or remotely located. Either deployment scenario requires a focus on the pilot's situation awareness (Endsley 1988); however, situation awareness for distributed team situations is inherently more complex (Salmon et al. 2008). The goal directed task analysis (GDTA) is a type of cognitive task analysis that places a direct focus on situation awareness (Endsley et al. 2003). Specifically, the GDTA focuses on the dynamic situation awareness requirements while identifying the basic goals and decisions that the personnel must fulfill independent of the information source. The goals and decision questions can be used to identify the cognitive demands that impact situation awareness and the required capabilities that UASs must have in order to appropriately integrate into an existing system, such as a manned air space.

Humphrey and Adams (2011) augmented the GDTA goal-decision-SA structure when applying the technique to the analysis of the integration of unmanned systems, both ground and aerial into first response to chemical, biological, radiological,

nuclear, and explosive (CBRNE) incidents. The application of the GDTA to the CBRNE domain analyzed the human responders as system components in order to understand how unmanned systems can be designed for effective integration into the overall domain or system. Specifically, Humphrey and Adams (2011) extended the GDTA with a loose representation of the goal timing/dependences, the ability to represent tools and thought processes associated with a subgoal, and the individuals and groups associated with subgoals. They also developed a method for tracking unique situation awareness information requirements across multiple goals and tasks for very large, complex domains.

Adams et al. applied the GDTA to an existing research system that employs mini-UASs for wilderness search and rescue, or WiSAR (Adams et al. 2009; Goodrich et al. 2008, 2009). The purpose of the analysis was to identify effective operational paradigms for coordination among the UAS operator, the incident commander, and the ground searchers. Unlike Humphrey and Adams' CBRNE GDTA analysis, the WiSAR domain analysis was a textbook application of the GDTA (Endsley et al. 2003). The analysis resulted in a set of primary goals that were decomposed into a number of subgoals and associated decision questions (Adams et al. 2009). Each subgoal and task was numbered and placed in an order that loosely matches the timing of the search events, similar to Humphrey and Adams' CBRNE research (2011). A difference was that two complex information requirements that applied to multiple goals and subgoals were called out as separate detailed lists, which represented the environmental characteristics and the missing person information (Adams et al. 2009). The GDTA resulted in an understanding of the situation awareness information requirements indicating what information the UAS can gather to inform and augment the information provided by the human searchers. The GDTA also identified a number of cognitively demanding tasks that the system personnel must complete. As well, the analysis identified search techniques and potential communication breakdowns between the searchers and the UAS operator and incident commander.

Beyond the applications of Adams and colleagues (Humphrey and Adams 2009, 2011; Adams et al. 2009), other examples of applying GDTA to unmanned systems exist. Riley et al. (2006) conducted a GDTA of an existing urban search and rescue unmanned ground vehicle in order to understand the situation awareness requirements for tasking and controlling a single vehicle. Riley and Endsley (2005) applied the GDTA to a proposed military system containing both unmanned ground and aerial systems in order to understand the situation awareness for control and tasking for a collaborative task involving multiple unmanned systems and operators. Conners et al. (2008) used GDTA for developing collaborative control interfaces for teams of operators tasked with managing multiple unmanned systems, while Cosenzo et al. (2009) applied the GDTA to the development of a common interface design for soldier-machine interaction with manned and unmanned ground systems. Hanson et al. (2001) applied cognitive task analysis, including the GDTA, to the supervision of composite striker packages composed of manned and unmanned aerial systems, with a primary focus of increasing the mission controller's situation awareness.

The hybrid cognitive task analysis method proposed by Nehme et al. (2006), and extended in Almirao et al. (2007), involves generating a scenario task overview, an event flow diagram, situation awareness requirements, decision ladders of critical decisions, and a requirements summary table. The hybrid cognitive task analysis' scenario task overview and event flow diagram provide system goals, while the list of situation awareness requirements and decision ladders represents information. The resulting requirements summary table represents the type of requirement, a requirement description, and the source of the requirement. Nehme et al. (2006) applied the hybrid cognitive task analysis method in order to generate information and display requirements for futuristic unmanned systems (underwater and aerial) applications.

Cooke and Shope (2004) designed a synthetic task environment, a simulation for Predator UAS operation based on a cognitive hierarchical goal-based task analysis conducted with U.S. Air Force Predator personnel (Gugerty et al. 1999), actual operation deployments, and results from an existing synthetic task environment (Martin et al. 1998). Hou et al. (2007) also applied hierarchical goal analysis to the development of an intelligent adaptive interface for the development of a multiple UAS tactical workstation.

As can be seen, the existing research, as applied to UASs, has used different methods. The analyses had different foci and purposes, thus resulting in different outcomes. This outcome is expected, since task analysis is intended to be applicable to multiple problems and domains, and the analysis can be applied for different purposes.

100.2.2 Work Analysis

Another important aspect when designing new technology is an appropriate understanding of the work domain. Cognitive work analysis permits an analysis of the human work based on domain constraints that can be device-independent and independent of worker competencies (Burns and Hajdukiewicz 2004; Vicente 1999). Cognitive work analysis is composed of five phases: work domain analysis, control task analysis, strategies analysis, social organization and cooperation analysis, and worker competencies analysis.

The work domain analysis permits a representation of the system independent of specific events, tasks, activities, and actions. The work domain analysis is an event-independent analysis that provides robust results applicable to all situations, including events that subject matter experts and system developers cannot anticipate (Jamieson et al. 2007; Miller and Vicente 2001). The control task analysis extends the work domain analysis findings by identifying the activities necessary to achieve the purposes, priorities, values, and functions within the work domain (Naikar et al. 2006) and provides an understanding of the information processing steps in relation to the required knowledge for achieving the stated goals. The control task analysis focuses on decomposing the system activities into control tasks for the various response situations and processes (Jamieson et al. 2007; Vicente 1999). The focus of the strategies analysis is to develop a process description of how the work should be

completed (Vicente 1999). Specifically, the strategies analysis attempts to identify the different means by which a task can be completed. The social organization and cooperation analysis focuses on identifying how the social and technical factors in the system can work together in such a way as to enhance the system performance (Vicente 1999). The final step, the worker competencies analysis, seeks to identify the competencies those using the system must possess in order to function effectively (Vicente 1999). The result of this analysis is a consolidation of the requirements from the first four phases as well as a determination of how the requirements can be achieved when considering the humans' capabilities and limitations (Vicente 1999).

The applied cognitive work analysis technique is grounded in the standard cognitive work analysis but is designed to be a "pragmatic, effective processes" for the design of revolutionary systems (Elm et al. 2003). This method was developed in order to better integrate cognitive analysis and the design and development of revolutionary systems. The method involves developing a functional abstraction network model capturing the domain concepts and relationships, identifying cognitive work requirements and information and relationship requirements, specifying the representation design requirements, and developing presentation design concepts.

Cummings (2003) developed the modified cognitive work analysis, which augments cognitive work analysis with an analysis of the global social, organizational, and ethical factors and the development of a system simulation. In general, both the modified cognitive work analysis and cognitive work analysis processes provide an understanding of the socio-technical context in which workers perform tasks (Vicente 1999). Cummings' modified cognitive work analysis is applicable not only to causal systems but also to intentional systems. The modified cognitive work analysis has been applied to naval warfare (Cummings and Guerlain 2003), and cognitive work analysis has been applied to a number of domains including emergency management (Vicente 1999).

Nehme et al. (2006) argue that traditional cognitive work analysis and cognitive task analysis cannot support the development of futuristic systems (i.e., systems that do not exist), because these methods require access to existing users, systems, and documentation. However, Drumm et al. (2003) applied work domain analysis to the design and development of a first-of-a-kind military system, where a first-of-a-kind system has no existing equivalent. Drumm et al. (2003) also modified cognitive work analysis for the design of teams by conducting a work domain analysis, an activity analysis, and a tabletop analysis to ascertain the feasibility of the proposed team designs. Humphrey and Adams (2010, 2011) built on Drumm et al.'s concepts to analyze the CBRNE domain for the purpose of informing the design and development of a first-of-a-kind unmanned systems to support teams of responders.

Humphrey and Adams' (2009, 2011) CBRNE domain analysis provided an event-independent analysis with results applicable to all situations, including events that subject matter experts and system developers cannot anticipate (Jamieson et al. 2007; Miller and Vicente 2001). The resulting understanding of the work domain's impact on the human-based CBRNE system permitted the identification of critical response activities that must be considered when developing unmanned systems.

Humphrey and Adams (2011) extended the cognitive work analysis (Cummings 2003; Vicente 1999) on multiple fronts. First, they adapted a concept of dividing the domain into multiple abstraction–decomposition spaces (Naikar et al. 2005). Others (Drumm et al. 2003; Jenkins et al. 2008) analyzed their domains by decomposing the teams into a hierarchy and sub-teams. The CBRNE domain contains many sub-teams (or subsystems) with unique, but sometimes overlapping, skill sets, tools, and responsibilities. Thus, Humphrey and Adams (2011) adapted Drumm et al.’s concept to the much larger CBRNE analysis in order to better identify roles and interaction customizations.

The cognitive work analysis’ control task analysis captures the relationship between actions and knowledge states (Vicente 1999) as well as some of the action dependences and scheduling via decision ladders (Rasmussen 1976). An additional modification by Humphrey and Adams was the use of statecharts (Harel 1987) to represent the control task analysis’ decision concurrency and hierarchical relationships succinctly (Humphrey 2009; Humphrey and Adams 2011). The representation of such constraints is paramount to understanding the team decision-making process (Gonzalez 2004). Humphrey and Adams’ analysis included the analysis of the global social, organizational, and ethical factors (Humphrey and Adams 2011). The analysis of relevant social groups identified 56 individuals or groups who may be affected by the CBRNE system. A communication flow map represents how different social groups communicate and depicts the direct and authorized communication interactions, but it does not represent how and by whom information is produced, consumed, or transformed.

Adams et al. (2009) applied the cognitive work analysis’ work domain analysis and the control task analysis to a research-based micro-UAS system for the WiSAR domain. In addition to providing insight into potential critical events, the work domain analysis captured (a) the high-level objectives of the WiSAR search process, (b) the priorities to be maintained during WiSAR, (c) the general functionality of the human-based WiSAR response, (d) the physical functions conducted and supported by the human responders and physical response devices, and (e) the physical devices required to conduct WiSAR. A standard work domain analysis (Vicente 1999; Naikar et al. 2005) represented by an abstraction–decomposition space was conducted. The decomposition dimension represents the different levels of detail required to reason about the search domain, with the rows representing the abstract functional decomposition for the different response concepts or the procedures to be reasoned over within the non-UAS, human-only WiSAR domain. The result is a categorization of the purposes, values and priorities, processes (functions), and physical resources (including the human responders) required for WiSAR. The work domain analysis indicated specific tasks and processes, as well as certain physical components required for an efficient search that locates the missing person as quickly as possible. The work domain analysis results informed the UAS-enabled WiSAR team’s organization and tactics as well as the development of operator interfaces to support managing system states based upon the current human-based response procedures and processes. The analysis strongly impacted the system design by clearly identifying that the WiSAR response has two key subsystems: information acquisition and information analysis.

The WiSAR control task analysis focused on decomposing the system activities into control tasks for the various response situations and processes (Jamieson et al. 2007; Vicente 1999). This analysis developed a decision ladder (Rasmussen 1976) to represent the information processing activities and the associated knowledge states required for conducting WiSAR (Adams et al. 2009). The control task analysis represents the control tasks necessary for understanding the current state of the WiSAR activities, which informs the control tasks required to achieve the desired system state and the resolution of the search activities. The control task analysis required an understanding of the work that needs to be completed with regard to the work processes, specifically the search procedures, and provided a very important lesson in that the search is essentially an iterated refinement of the probability describing the likely location of the missing person (Goodrich et al. 2008, 2009).

Rasmussen (1998) applied cognitive work analysis to modeling the changes to command and control when UASs are integrated into suppression of enemy air defense missions. Gonzalez Castro et al. (2007) conducted a cognitive work analysis focused on developing procedures, functions, and a proposed ground control station for continuous target surveillance via an UAS. The scope of this analysis was a well-defined problem for a single vehicle, and their analysis was not intended to inform vehicle design. Stilson (2008) analyzed the work requirements in order to develop a representational structure of work and a display design of a multiple UAS manager for the Predator unmanned aerial system. Pratt et al. (2008) also applied cognitive work analysis to determine the crew organization and protocols for deploying small tethered UASs during structural inspections in the field. Most recently, Jenkins (2012) applied cognitive work analysis to systematically understand the constraints for military air operations and define information requirements in order to postulate the implications of introducing UASs. Additional examples of applying CWA to display design for UASs exist (Stoner et al. 2006; Yagoda and Coovert 2012).

Cognitive work analysis and its derivations are intended to understand the work domain and, through the analysis steps, shift the focus to the human components. Many analysis examples employ the work domain analysis and cognitive task analysis steps, but very few examples exist of analyses that complete all designated steps. As is demonstrated by the cited examples of applying cognitive work analysis to UASs, even an application that only includes components of the analysis technique can be beneficial to system design.

100.2.3 Information Analysis

A subset of the cognitive task analysis methods focus on information analysis. Simple information flow diagrams represent the decisions and actions performed to complete a task by capturing the information flowing through the system, but such diagrams quickly become cumbersome for complex systems, particularly those that include large teams of humans (Kirwan and Ainsworth 1992). The limitation is the inability to easily represent multiple different individuals who are responsible for a single task or how information is modified as it flows through a system or organization. The function–information integration method represents the

information flow through the system, including data dependencies (Essens et al. 2000). This method provides a clear understanding of the information flow and dependencies but does not easily represent the association of different humans responsible for the task or the human organization. Operational sequence diagrams represent the information collection required to complete a task (Kirwan and Ainsworth 1992). Temporal operational sequence diagrams add the time required to complete each action. Partitioned operational sequence diagrams permit the representation of small team-based tasks at different task abstraction levels. The operation sequence diagram techniques are too simple to represent very complex, dynamic, and highly uncertain systems that contain UASs and multiple stakeholders. Additionally, information flow and operation sequence diagrams do not provide a clear understanding of how information may be used for each task, for example, tasks for which some information is required, while in other cases information may be optional.

Boy's cognitive function analysis results in linking artifacts that are being designed by considering a user's activities, the required tasks, and the organizational environment (Boy 1998). Boy extended the concept to multiple agent systems composed of humans and machine agents (Boy and Ferro 2004; Boy 2011); however, there appears to be limited validation of the ability to use the method for analyzing large scale, ad hoc multiple agent systems. In particular, the cited examples are quite constrained when compared to the analysis of very complex, highly dynamic, and highly uncertain domains that UASs may encounter. Boy (2011) indicates that the cognitive function analysis provides a "notion of context"; however, it is not possible to predict all possible contexts that deployed UASs and their human stakeholders will encounter. Boy's orchestra metaphor, employed to extend cognitive function analysis for multiple agents, is intended to be applied to problems that require multiple entities to develop contracts and coordinate with one another in real time while having specialized abilities (Boy 2009b). Boy's method has been applied to manned aviation domains (e.g., Boy and Ferro 2004, 2009a), but not to UAS domains.

Humphrey and Adams (2010) cognitive information flow analysis integrates the results from cognitive task and work analyses in order to provide a focus on the flow, understanding, and usage of information in large, ad hoc team domains. Specifically, this method is intended to allow the designer to understand how information is produced, consumed, and transformed in the system and by whom. Information is produced when it is gathered via humans, system sensors, or is the result of creating a new information product from existing information. Information is consumed when it is used by processes or individuals to satisfy a system function. Information is transformed when it is consumed by a system process that results in creating a new information product.

The cognitive information flow analysis combines results and provides a new perspective that includes the flow of information, potential information bottlenecks, and who is responsible for receiving and using the information as well as creating new information products. A limitation of many cognitive task analysis and the cognitive work analysis methods is that they do not explicitly represent what information is used by the stakeholders, which is important in

team-based environments, such as UAS deployments with multiple stakeholders (Adams et al. 2009; Goodrich et al. 2008; Murphy et al. 2008; Pratt et al. 2008), or how information items are produced, consumed, and transformed. Specifically, the GDTA employs goals that represent information to be consumed or used, but does not explicitly represent what information is produced or how it is changed. The cognitive work analysis' control task analysis represents how knowledge states are produced, consumed, and changed; however, knowledge states and information components are not equivalent. A knowledge state is a holistic, cognitive state a decision maker is in either before the decision or after. Information components are individual pieces of knowledge that assist a decision maker in making a decision or are produced by the decision maker as a consequence of the decision. The modified cognitive work analysis communications flow map creates a hierarchy of humans representing direct and authoritative lines of communication, but does not represent how and by whom information is produced, consumed, or used. The cognitive information flow analysis addresses these limitations by focusing explicitly on the flow of information through tasks and identifies groups of individuals responsible for the tasks. The flow of information represents the production, consumption, and transformation of information, which can directly inform system and interface design requirements that support decision-making.

100.2.4 Combining Analysis Techniques

It is well known that there exist various benefits and pitfalls associated with each analysis approach, as well as the individual techniques within the analysis categories (Humphrey and Adams 2011; Kirwan and Ainsworth 1992; Vicente 1999; Xu 2007). Much debate has ensued as to the benefits of individual cognitive task analysis methods compared to one another and the benefits of cognitive task analysis as compared to cognitive work analysis. As such, comparisons of the methods have been reported. Miller and Vicente (2001) compared hierarchical task analysis with an abstraction–decomposition space, or work domain analysis, and presented the associated advantages and disadvantages. Jamieson et al. (2007) also compared hierarchical task analysis and work domain analysis. Miller and Vicente (2001) and Jamieson et al. (2007) conclude that the two techniques have differences but that the differences are complementary.

Due to the limitations associated with individual analysis methods, it has become common to combine multiple methods when analyzing systems (Crandall et al. 2006; Humphrey and Adams 2010, 2011; Potter et al. 2000). The applied cognitive work analysis (Elm et al. 2003) is an example of a method intended to capitalize on the various benefits. Laberge et al. (2007) combined hierarchical task analysis and work domain analysis results when identifying information requirements for developing an intersection decision support display for rural drivers. Kaber et al. (2006) applied GDTA and constructed an abstraction–decomposition space (e.g., work domain analysis) of a high-throughput organic compound screening operation. Kaber et al. found that combining the analyses results facilitated system design by providing a better understanding of the automation needs, the existing technologies,

the operator needs, and the interface design limitations. Diaper et al. developed pentanalysis to integrate task analysis with data flow analysis (Diaper et al. 1998), while Ntuen et al. combined various cognitive task analysis methods with work domain analysis (Ntuen et al. 2006).

Humphrey and Adams (2011) employed both cognitive task analysis, via the GDTA, and cognitive work analysis, via the modified cognitive work analysis. Humphrey and Adams also created the cognitive information flow analysis to combine the results and provide a unique understanding of the system (Humphrey and Adams 2010). Specifically, the intent was to define tasks for unmanned aerial and ground systems based on an analysis of the human-based tasks (Humphrey and Adams 2009), result in unmanned system design specifications (Humphrey and Adams 2011), define emergency response human–robot interaction roles (Humphrey 2009), and identify the information flow, and teaming and design of a system of interfaces to support a large (nine levels) command hierarchy (Humphrey 2009). Humphrey and Adams specifically employed both the GDTA and the modified cognitive work analysis in order to balance each method's cited limitations; see Humphrey and Adams (2011) for limitations.

Humphrey and Adams specifically wanted to include a representation of the humans in the incident response system. This perspective differs from traditional cognitive task and work analysis that views the human as a system user. As well, many cognitive task and work analysis methods focus on a single human user, rather than a team of individuals. A number of analyses have focused on relatively small teams of one to four team members (see Schraagen and Rasker (2003) for a review), while a few analyses have incorporated teams of 20–25 members (Annett and Cunningham 2000; Essens et al. 2000; Zachary et al. 2000). The ability to scale the analysis techniques to very large teams, such as the hundreds of humans represented in the CBRNE domain, is not simplistic. As a result, Humphrey and Adams (2011) augmented both the GDTA and the modified cognitive work analysis. An important aspect is that Humphrey and Adams (2011) conducted the analyses in a manner that viewed the human responders as system components. This perspective was deemed appropriate because the humans complete tasks that are necessary for the overall system to achieve the system goals. Adams et al. (2009) analysis of mini-unmanned aerial systems for WiSAR also considered the human responders as system components.

Humphrey and Adams' GDTA permitted a focus on the information required by personnel and the information that unmanned systems must provide in order to emulate human responder provided information (Humphrey and Adams 2011). However, the GDTA does not focus on the required information processing steps, how information is transformed by the system, and the timing of response activities and does not capture the activities that occur simultaneously or the conflicts that occur between simultaneous goals. Humphrey and Adams (2011) augmentations to the GDTA were intended to overcome these limitations. Additionally, the GDTA does not focus on unexpected events that may occur in the work domain, which was provided by modified cognitive work analysis. An understanding of the work domain's impact on the human-based system allows for the identification of critical events that must be considered when designing the unmanned system and the

associated system of interfaces. The work domain analysis captured high-level objectives, the priorities to be maintained, the general functionality of the human-based system, the physical functions conducted and supported by the humans, and the physical devices required by the system.

Humphrey and Adams (2010) cognitive information flow analysis was designed to assist the design and development of systems by explicitly representing how and by whom information items are produced, consumed, and transformed. Specifically, the cognitive information flow analysis was developed to combine the results from the GDTA and cognitive work analysis methods in a manner that provided new insights for system development while building on the strengths of the GDTA and the modified cognitive work analysis and addressing some of their cited limitations. Specifically, the cognitive information flow analysis identifies how information from the unmanned systems is used to complete different goals and tasks within the system and how this information is transformed by different system tasks and provided to personnel who require the information for task planning, ultimately resulting in new task assignments for the unmanned systems. This combination of the GDTA, cognitive work analysis, and cognitive information flow analysis provided a general response system level of context when analyzed as a whole, rather than the individual analysis representations.

Adams et al. (2009) also employed multiple techniques in their analysis of the WiSAR domain, in particular the GDTA and the cognitive work analysis' work domain and control task analyses. The combination of the analysis techniques provided a base of information that allowed the UAS developers to better understand the human-based WiSAR response and to formulate the correct questions earlier in the development process. The analysis informed the design and development of autonomous UAS algorithms, the associated operator interfaces, and the human roles and responsibilities (Goodrich et al. 2008, 2009). However, it is important to note that these analysis techniques were not the sole contributors to the design and development of the UAS-enabled WiSAR technology. The early technology development, via early flight tests and field trials, informed the analysis conducted by Adams et al. and the analysis results informed the later technology development, thus successfully integrating the analysis techniques into the systems engineering process.

The combination of analysis techniques can lead to a richer representation of the task requirements, work domains, cognitive demands, system representations, and human roles for the system being analyzed. However, combining techniques should be done as a means of obtaining information not captured by one or more of the techniques, rather than simply capturing the same or a significant subset of the information. The techniques should be chosen to compliment the weakness of each other and result in new insights. However, if a particular technique can meet all of the analysis objectives, then it may not be necessary to employ multiple analysis techniques. Information gathering, analysis development, and maintaining the analysis are very complicated for complex systems such as UASs; this difficulty is increased by conducting multiple types of analyses.

100.3 Conclusion

The representations of applications of the analysis methods in the literature abound. These examples cross domains, methods, and purposes. The application of cognitive task analysis, cognitive work analysis, and information analysis techniques is relatively new to the unmanned system domain and in particular to UASs. However, the basic lessons learned from applying such techniques across different problems are important when considering their use for UASs. The intent of employing such methods is to understand the underlying cognitive and work demands associated with system operation. UASs range from the very simple remotely controlled systems available as toys, to the very complex, long flight duration military UASs. The planned deployments for such systems include benign image collection; very dynamic and dangerous domains, such as first response; and very dense and dynamic air spaces, such as military deployments incorporating both manned and unmanned aircraft systems. The future design of such systems and their associated human-machine system interaction needs to accommodate these complexities and the associated demands placed on the humans. It is clear that cognitive task analysis, cognitive work analysis, and information flow analysis techniques are critical to the development of future systems. Unfortunately, there is no single clear method that is appropriate for all system design and development endeavors. Advocates exist for particular methods, such as cognitive task analysis, who do not believe that cognitive work analysis is appropriate, and vice versa. Thus, it is necessary to analyze the objectives to be achieved by conducting the analysis, the scope of the analysis, the constraints on the analysis, and the capabilities of each method prior to commencing the analysis.

One issue associated with task analysis and cognitive work analysis techniques is the difficulty of translating the analysis results into actual interaction design, specifically user interface design (Potter et al. 2003). This particular issue has been noted by many (e.g., Humphrey and Adams 2011, Xu 2007) and should be considered when determining the analysis scope and objectives. Humphrey and Adams (2010) cognitive information flow analysis provides insight into the type of information required for a particular type of use, but does not specify the interaction design. One approach to informing design for revolutionary systems, meaning systems that require an innovative design, is applied cognitive work analysis (Elm et al. 2003).

Conducting these types of analyses is very time-consuming and cannot be completed based upon simply reviewing the existing literature or system design documents. These types of analyses require direct interaction with the system users, preferably while they are using the existing system. However, it is not always possible to conduct the data collection on the actual system; rather, it may be necessary to seek other means of collecting the data. Adams et al. (2009) employed field trials of wilderness search and rescue training sessions; however, Humphrey and Adams (2010, 2011) were limited in their analysis of the chemical, biological, radiological, nuclear, and explosive device incident response domain.

Such incidents do not occur frequently, and researchers are not permitted into the contaminated area; however, it is possible to observe training exercises that occur across the country. Successful UAS design, development, and deployment are highly dependent on a thorough analysis of the existing or planned system, including the demands placed on the integrated humans. Thus, it is absolutely necessary to identify and analyze actual or potential users of the resulting system in order to have a valid analysis.

References

- J.A. Adams et al., Cognitive task analysis for developing UAV wilderness search support. *J. Cogn. Eng. Decis. Mak.* **3**(1), 1–26 (2007)
- F.M. Almirao, F.B. da Silva, S.D. Scott, M. Cummings, Designing decision and collaboration support technology for operators in multi-UAV operations teams, Technical Report #HAL2007-02, Massachusetts Institute of Technology Humans and Automation Laboratory, Cambridge, 2007
- J. Annett, D. Cunningham, Analysing command team skills, in *Cognitive Task Analysis*, ed. by J.M. Schraagen, S.F. Chipman, V.L. Shalin (Lawrence Erlbaum, London, 2000), pp. 401–415
- J. Annett, N. Stanton, *Task Analysis* (Taylor and Francis, London, 2000)
- G.A. Boy, *Cognitive Function Analysis* (Ablex Publishing Corporation, Stamford, 1998)
- G.A. Boy, Authority in increasingly complex human and machine collaborative systems: application to the future air traffic management construction, in *Proceedings of the International Ergonomics Association Triennial Congress*, Tainan, 2009a
- G.A. Boy, The orchestra: a conceptual model for function allocation and scenario-based engineering in multi-agent safety-critical systems, in *Proceedings of the IEEE Energy Conversion Congress and Exposition*, San Jose, 2009b, pp. 187–193
- G.A. Boy, Cognitive function analysis in the design of human and machine multi-agent systems, in *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*, ed. by G.A. Boy (Ashgate, Surrey, 2011), pp. 189–206
- G.A. Boy, D. Ferro, Using cognitive function analysis to prevent controlled flight into terrain, in *Human Factors for Civil Flight Deck Design*, ed. by D. Harris (Ashgate, Burlington, 2004), pp. 55–68
- C.M. Burns, J.R. Hajdukiewicz, *Ecological Interface Design* (CRC, London, 2004)
- S.F. Chipman, J.M. Schraagen, V.L. Shalin, Introduction to cognitive task analysis, in *Cognitive Task Analysis*, ed. by J.M. Schraagen, S.F. Chipman, V.L. Shalin (Lawrence Erlbaum Associates, London, 2000), pp. 3–23
- E.S. Connors, L.D. Strater, J.M. Riley, M.R. Endsley, Applying SA-oriented design to the integrative collaborative control for multiple unmanned vehicle systems, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Santa Monica, vol. 52, no. 4, 2008, pp. 211–215
- N.J. Cooke, S.M. Shope, Designing a synthetic task environment, in *Scaled Worlds: Development, Validation, and Application*, ed. by S.G. Schiflett, L.R. Elliott, E. Salas, M.D. Coover (Ashgate, Surrey, 2004), pp. 263–278
- K.A. Cosenzo, E. Capstick, R. Pomranky, T. Johnson, S. Dungrani, Solider machine interface for vehicle formations: interface design and an approach evaluation and experimentation, Technical Report: ARL-TR-4678, Army Research Laboratory, 2009
- J.W. Crandall, M.A. Goodrich, D.R. Olsen Jr., C.W. Nielsen, Validating human-robot interaction schemes in multitasking environments. *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans* **35**(4), 438 (2005)
- B. Crandall, G. Klien, R.R. Hoffman, *Working Minds: A Practitioner's Guide to Cognitive Task Analysis* (MIT, Cambridge, 2006)

- M.L. Cummings, Designing decision support systems for a revolutionary command and control domains, Doctoral Dissertation, University of Virginia, 2003
- M.L. Cummings, S. Guerlain, The tactical tomahawk conundrum: designing decision support systems for revolutionary domains, in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Washington, DC, 2003, pp. 1583–1588
- D. Diaper, S. McKearney, J. Hurne, Integrating task and data flow analyses using the pentanalysis technique. *Ergonomics* **41**, 1553–1582 (1998)
- D. Diaper, N. Stanton, *The Handbook of Task Analysis for Human-Computer Interaction* (Lawrence Erlbaum Associates, London, 2004)
- D. Drumm, N. Naikar, B. Pearce, P.M. Sanderson, Designing teams for first-of-a-kind, complex systems using the initial phases of cognitive work analysis: case study. *Hum. Factor* **45**(2), 202–217 (2003)
- W.C. Elm, S.S. Potter, J.W. Gualtieri, J.W. Easter, E.M. Roth, Applied cognitive work analysis: a pragmatic methodology for designing revolutionary cognitive affordances, in *The Handbook of Cognitive Task Design*, ed. by E. Hollnagel (Lawrence Erlbaum, London, 2003), pp. 357–382
- M.R. Endsley, Situation awareness global assessment technique, in *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference*, 1988, pp. 789–795
- M.R. Endsley, B. Bolté, D.G. Jones, *Designing for Situation Awareness: An Approach to User-Centered Design* (Taylor & Francis, London, 2003)
- M.R. Endsley, D.B. Kaber, Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics* **42**(3) 462–492 (1999)
- J.M.D. Essens, W.M. Post, P.C. Rasker, Modeling a command center, in *Cognitive Task Analysis*, ed. by J.M. Schraagen, S.F. Chipman, V.L. Shalin (Lawrence Erlbaum, London, 2000), pp. 385–399
- J.J. Gibson, *The Ecological Approach to Visual Perception* (Houghton Mifflin, Boston, 1979)
- T. Goetzendorf-Grabowski, A. Frydrychewicz, Z. Goraj, S. Suchodolski, MALE UAV design of an increased reliability level. *Aircr. Eng. Aerosp. Technol.* **78**(3), 226–235 (2006)
- C. Gonzalez, Learning to make decisions in dynamic environments: effects of time constraints and cognitive abilities. *Hum. Factor* **46**(3), 449–460 (2004)
- L.N. Gonzalez Castro et al., Applying coherent design to uninhabited aerial vehicle operations and control stations, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Baltimore, 2007, pp. 181–185
- M.A. Goodrich, B.S. Morse, C. Engh, J.L. Cooper, J.A. Adams, Towards using UAVs in Wilderness search and rescue: lessons from field trials. *Interact. Stud.* **10**(3), 453–478 (2009)
- M.A. Goodrich, B.S. Morse, D. Gerhardt, J.L. Cooper, M. Quigley, C. Humphrey, J.A. Adams, Supporting Wilderness Search and Rescue using a Camera-Equipped Mini UAV. *J. Field Rob.* **25**(1–2), 89–110 (2008). Special issue on Safety, Security, and Rescue Robotics
- L. Gugerty, D. DeBoom, R. Walker, J. Burns, Developing a simulated uninhabited aerial vehicle task based on cognitive task analysis: task analysis results and preliminary simulator data, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Houston, 1999, pp. 86–90
- M.L. Hanson, O. Sullivan, K.A. Harper, On-line situation awareness for unmanned aerial vehicles, in *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference (AAAI, Menlo Park, 2001)*, pp. 44–48
- D. Harel, Statecharts: a visual formalism for complex systems. *Sci. Comput. Progr.* **8**(3), 231–274 (1987)
- E. Hollnagel, *Handbook of Cognitive Task Design* (Lawrence Erlbaum Associates, London, 2003)
- M. Hou, R.D. Kobierski, M. Brown, Intelligent adaptive interfaces for control of multiple UAVs. *J. Cognit. Eng. Decis. Mak.* **1**(3), 372–362 (2007)
- C.M. Humphrey, Information abstraction visualization for human-robot interaction, Ph.D. Dissertation. Computer Science, Vanderbilt University, 2009
- C.M. Humphrey, J.A. Adams, Robotic tasks for chemical, biological, radiation, nuclear and explosive incident response. *Adv. Rob.* **23**, 1217–1232 (2009)

- C.M. Humphrey, J.A. Adams, Cognitive information flow analysis. *Cognit. Technol. Work* (2010) <http://link.springer.com/article/10.1007/s10111-011-0198-z>
- C.M. Humphrey, J.A. Adams, Analysis of complex team-based systems: augmentation to goal-directed task analysis and cognitive work analysis. *Theor. Issue Ergon. Sci.* **12**(2), 149–175 (2011)
- G. Jamieson et al., Integrating task- and work domain-based work analyses in ecological interface design: a process control case study. *IEEE Trans. Syst. Man Cybern. Part A* **37**(6), 887–905 (2007)
- D.P. Jenkins, Using cognitive work analysis to describe the role of UAVs in military operations. *Theor. Issue Ergon. Sci.* **13**(3), 335–357 (2012)
- D.P. Jenkins et al., Applying cognitive work analysis to the design of rapidly reconfigurable interfaces in complex networks. *Theor. Issue Ergon. Sci.* **9**(4), 273–295 (2008)
- D.B. Kaber et al., Using multiple cognitive task analysis methods for supervisory control interface design in high-throughput biological screening processes. *Cognit. Technol. Work* **8**(4), 237–252 (2006)
- B. Kirwan, L.K. Ainsworth, *A Guide to Task Analysis* (Taylor & Francis, London, 1992)
- J. Laberge, N. Ward, M. Rakauskas, J. Creaser, A comparison of work domain and task analyses for identifying information requirements: a case study of rural intersection decision support systems, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Baltimore, 2007, pp. 298–302
- E. Martin, D.R. Lyon, B.T. Schreiber, Designing synthetic tasks for human factors research: an application to uninhabited air vehicles, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 1998, pp. 123–127
- C. Miller, K.J. Vicente, Comparison of display requirements generated via hierarchical task and abstraction-decomposition space analysis techniques. *Int. J. Cognit. Ergon.* **5**, 335–355 (2001)
- R.R. Murphy, K.S. Pratt, J.L. Burke, Crew roles and operational protocols for rotary-wing micro-UAVs in close urban environments, in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, Amsterdam, 2008, pp. 73–80
- N. Naikar, R. Hopcroft, A. Moylan, Work domain analysis: theoretical concepts and methodology, Technical Report, Australian Government Air Operations Division, DSTO Defence Science and Technology Organisation, Victoria, 2005
- N. Naikar, A. Moylan, B. Pearce, Analysing activity in complex systems with cognitive work analysis: concepts, guidelines and case study for control task analysis. *Theor. Issue Ergon. Sci.* **7**(4), 371–394 (2006)
- C.E. Nehme et al., Generating requirements for futuristic heterogeneous unmanned systems, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, San Francisco, 2006, pp. 235–239
- D.A. Norman. *The Design of Everyday Things* (Currency Doubleday, New York, 1988). Previously published as *The Psychology of Everyday Things*
- C.A. Ntuen, O. Balogun, E. Boule, A. Turner, Supporting command and control training functions in the emergency management domain using cognitive system engineering. *Ergonomics* **49** (12–13), 1415–1436 (2006)
- R. Parasuraman, T.B. Sheridan, C.D. Wickens, A model for types and levels of human interaction with automation. *IEEE Trans. Syst. Man Cyber. Part A* **30**(3), 286–297 (2000)
- S.S. Potter, J.W. Gualtiere, W.C. Elm, Case studies: applied cognitive work analysis in the design of innovative decision support, in *The Handbook of Cognitive Task Design*, ed. by E. Hollnagel (Lawrence Erlbaum, London, 2003), pp. 654–677
- S.S. Potter, E.M. Roth, D.D. Woods, W.C. Elm, Bootstrapping multiple converging cognitive task analysis techniques from system design, in *Cognitive Task Analysis*, ed. by J.M. Schraagen, S.F. Chpman, V.L. Shalin (Lawrence Erlbaum Associates, London, 2000), pp. 317–340
- K.S. Pratt, R.R. Murphy, J.L. Burke, J. Craighead, C. Griffin, S. Stover, Use of tethered small unmanned aerial system at Berkman Plaza II collapse, in *Proceedings of the 2008 IEEE International Workshop of Safety, Security and Rescue Robotics*, 2008, pp. 134–139

- M. Quigley, M.A. Goodrich, S. Griffiths, A. Eldredge, R.W. Beard, Target acquisition, localization, and surveillance using a fixed-wing, mini-UAV and gimbaled camera, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 3028–3033
- J. Rasmussen, Outlines of a hybrid model of the process plant operator, in *Monitoring Behavior and Supervisory Control*, ed. by T. Sheridan, G. Johanssen (Plenum, New York, 1976)
- J. Rasmussen, Ecological interface design for complex systems: an example: SEAD UAV systems. Technical Report, United States Air Force Research Laboratory, Wright-Patterson AFB, 1998
- J. Riley, M.R. Endsley, Situation Awareness in HRI with collaborating remotely piloted vehicles, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Orlando, 2005, pp. 407–411
- J.M. Riley, R.R. Murphy, M.R. Endsley, Situation awareness in the control of unmanned ground vehicles, in *Human Factors of Remotely Operated Vehicles (Advances in Human Performance and Cognitive Engineering Research)*, vol. 7, ed. by N.J. Cooke, M.H. Pringle, H. Pedersen, O. Connor (Emerald Group/JAI, Bingley, 2006), pp. 359–371
- P.M. Salmon et al., What is really going on? Review of situation awareness models for individuals and teams. *Theor. Issue Ergon. Sci.* **9**(4), 297–323 (2008)
- J.M. Schraagen, S.F. Chipman, V.L. Shalin, *Cognitive Task Analysis* (Lawrence Erlbaum Associates, London, 2000)
- J.M. Schraagen, P. Rasker, Team Design, in *Handbook of Cognitive Task Design*, ed. by E. Hollnagel (Lawrence Erlbaum, London, 2003), pp. 753–786
- T.B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control* (MIT, Cambridge, 1992)
- T.B. Sheridan, *Humans and Automation: System Design and Research Issues* (Wiley, New York, 2002)
- M.T. Stilson, Multi-UAV control: an envisioned world design problem, Master's Thesis, Wright State University, 2008
- H.A. Stoner, A. Alexander, M. Lingang, M. Patterson, Using cognitive work analysis for developing user-centered unmanned vehicle displays, in *The Third Annual Human Factors of UAV Workshop Manning the Unmanned*, San Francisco, 2006
- K. Vicente, *Cognitive Work Analysis: Toward Safe, Productive and Healthy Computer-Based Work* (Lawrence Erlbaum Associates, Mahwah, 1999)
- W. Xu, Identifying problems and generating recommendations for enhancing complex systems: applying the abstraction hierarchy framework as an analytical tool. *Hum. Factor* **49**(6), 975–994 (2007)
- R.E. Yagoda, M.D. Coovert, How to work and play with robots: an approach to modeling human-robot interaction. *Comput. Human Behav.* **28**(1), 60–68 (2012)
- W.W. Zachary, J.M. Ryder, J.H. Hicinbothom, Building cognitive task analyses and models of a decision-making team in a complex real-time environment, in *Cognitive Task Analysis*, ed. by J.M. Schraagen, S.F. Chipman, V.L. Shalin (Lawrence Erlbaum, London, 2000), pp. 365–383

Gloria L. Calhoun and Mark H. Draper

Contents

101.1 Introduction 2444

101.2 Multi-UAV Operator Ground Control Station Test Beds 2445

101.3 Multi-UAV Interface Concepts for Vehicle/Mission Management 2446

101.4 Display Concepts for Vehicle/Mission Management 2447

101.5 Control Concepts for Vehicle/Mission Management 2458

101.6 Multi-UAV Interface Concepts for Payload Management 2463

101.7 Conclusion 2469

References 2469

Abstract

Advances in automation technology are leading to development of unmanned aerial vehicle (UAV) concepts in which a single pilot is responsible for multiple UAVs. With these highly automated UAVs, the operator’s role will become more supervisory in nature. In response, a new interface paradigm is required for effective supervisory control of multiple UAVs by a single operator. This chapter introduces a variety of display and control concepts that may be especially applicable to multi-UAV applications. Many of these are technologies being investigated at the United States Air Force Research Laboratory and include multimodal interfaces, temporal displays, and integrated visualization concepts. For any candidate technology, however, rigorous user-centered design and evaluation needs to be conducted that focuses on the tasks and requirements of the specific multi-UAV application being targeted. It is also important to ensure that the operator has adequate means to first observe and then direct the automation’s functioning in order to be responsive to changes in the mission, vehicle status, or operational environment.

G.L. Calhoun (✉) • M.H. Draper
Wright-Patterson Air Force Base, 711 HPW/RHCI, Air Force Research Laboratory, Dayton,
OH, USA

101.1 Introduction

Unmanned aerial vehicles (UAVs) are becoming an increasingly critical aspect of military operations and are starting to be more extensively used in civilian applications as well (Department of Defense (DoD) 2007). With these successes comes a vision for even more capable systems that can accomplish missions with greater complexity, including the ability of a *single operator* to control *multiple platforms simultaneously*. Currently, most UAV systems require two or more operators to fly the vehicle while managing the onboard sensor(s). Single-operator control of multiple UAVs is envisioned to be a particularly time-critical, cognitively demanding work environment (Scott et al. 2006). Additionally, with newer, highly automated UAVs, the operator's role will become more supervisory in nature: monitoring the environment and situation along with the automated activation of programmed events, managing changes to the mission plans and automated systems, and collaborating intermittently with others in a distributed control network (Chen et al. 2010). In response, a new interface paradigm is required for effective supervisory control of multiple UAVs by a single operator.

A number of approaches for multi-UAV employment by a single operator are being considered. Eggers and Draper (2006) describe several for controlling multiple aircraft (interleaving, sequential, and simultaneous) as well as list operationally inspired questions that a designer should consider when developing a multi-UAV control system. These authors stress that the particular mission, its required effectiveness, and the associated UAV capability requirements should be the key drivers for a multi-UAV control system design. More specifically, the missions to be accomplished and the level of platform/sensor sophistication influence the range and levels of platform automation, as well as degree to which the system should flexibly accommodate operator intervention. This, in turn, determines the operator's display and control requirements.

In addition to the many mission-specific design drivers, there are also several generalized human factors challenges to consider when designing a multi-UAV control station for single-operator control. Any operator interface design should serve to mitigate known automation-induced problems such as complacency, bias, vigilance decrement, mode confusion, automation brittleness, and attention/cognitive tunneling (Endsley and Kaber 1999). Moreover, it is important to keep the operator continually informed of automation functioning and planned activity for accurate situation assessments and command decisions. This will involve ensuring the operator's information displays provide the *right* information at the *right* time for the operator to maintain adequate situation awareness for effective decision-making across different situations and/or vehicles. Thus, intelligent aids are needed that locate, select, and/or filter information based on specific context to help streamline information gathering and assessment tasks, especially when the operator is multitasking (St. John and King 2010). To meet all these requirements, innovative display approaches will need to be considered that provide the required information intuitively and in a compact manner. This is because presenting information using

conventional status displays for each vehicle would require considerable display space, increasing the areas that need to be visually scanned and thus complicating information retrieval.

In addition to information display issues, there are also control factors to consider. One involves determining and supporting the appropriate levels and types of interactions with the automated systems. Multi-UAV systems will necessarily involve more supervisory control interactions with little to no manual control via stick or throttle. A key challenge will be instantiating these new methods for intuitive vehicle and platform control such that many vehicles can be simultaneously managed. However, for some complex, rapidly evolving, and time-critical missions, current limitations in automation technology may preclude rapid precise control via highly supervisory means. In these special cases, the capability may still be needed for an operator to assume direct manual control for a limited period of time. Therefore, the capability to assume increased manual control might need to be incorporated into the design of certain UAV applications. Alternatively, a distributed control architecture could be developed that enables the operator to easily handover control of a UAV to an operator/station that is dedicated to handling these special missions/situations (Eggers and Draper 2006). It is also important to ensure control actions are intuitive, and it needs to be obvious which vehicle(s) will be impacted by a specific control input. For multi-UAV control, additional functionality will be required that can be rapidly employed when the operator is dealing with an especially difficult or unanticipated event – for example, a single command that results in multiple vehicles continuing in a safe manner (e.g., all loiter present location), enabling the operator to focus on the vehicle needing dedicated attention.

A number of research and development efforts are addressing the human factors issues identified above. The present chapter will introduce several display and control concepts that may be applicable to multi-UAV control, as well as identify directions for further research. Due to space limitations, the primary focus will be on technologies developed and evaluated at the United States Air Force Research Laboratory (AFRL) at Wright-Patterson Air Force Base.

101.2 Multi-UAV Operator Ground Control Station Test Beds

There have been a number of multi-UAV ground control simulations developed recently to support human factors research pertaining to future multi-UAV supervisory control issues, to name several: *MIIRO* (Multi-Modal Immersive Intelligent Interface for Remote Operation; Calhoun et al. 2007a), *MUSE* (Multiple UAV Simulation Environment; Kim et al. 2007), *ALOA* (Adaptive Levels of Autonomy; Johnson et al. 2007), *RESCHU* (Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles; Boussemart et al. 2009), *MUSIM* (Multiple-UAS Simulator; Fern and Shively 2009), *AMASE* (AVTAS (Air Vehicles Test and Simulation) Multi-Agent Simulation Environment; Duquette 2011), and *VSCS* (Vigilant Spirit Control Station; Feitshans et al. 2008). These test beds vary in terms

of functionality, flexibility, and emphasis areas, and they provide an indication of the wide variety of potential designs associated with multi-UAV control.

The Vigilant Spirit Control Station was developed by the Human Effectiveness Directorate (AFRL, 711 Human Performance Wing) to fulfill several purposes. First, the system primarily supports laboratory research evaluations of candidate operator interface technologies due to its capability to present prototype interface designs and collect usage data during preplanned trials. Second, it aids in stimulating a synthetic environment for the modeling of various vehicle payloads, sensors, and human factors testing tools because the simulation is high fidelity and designed to realistically represent the tasking and workload envisioned for future multi-UAV control. Finally, the Vigilant Spirit Control Station routinely supports high-fidelity simulation demonstrations as well as live flight testing of UAVs. To accomplish this, the simulation has the ability to communicate in a standard way with various commercial UAV platforms and operate in a manner that meets potential safety of flight issues. Besides meeting interoperability requirements, it also has dynamic mission planning interfaces. To meet all these objectives, its design utilizes an open architecture and a highly flexible modular approach in order to provide a variety of tools. This enables this multi-UAV Vigilant Spirit Control Station to be configured to accommodate diverse missions and vehicle payloads across multiple vehicle platforms. Moreover, tools outside its current core capability can be accommodated due to the simulation's robust and flexible software architecture (Feitshans et al. 2008).

Generally, all multi-UAV ground control simulations include two types of management interfaces: (1) vehicle/mission and (2) payload (e.g., sensors). In the next two sections, both types of interfaces will be addressed, in turn, by describing concepts that may be useful for implementing multi-UAV supervisory control. The majority of specific examples are drawn from the developments associated with AFRL's Vigilant Spirit Control Station (see Rowe et al. 2009), and many of these are depicted in figures. Note that illustrations of these concepts in this chapter are intended only to convey examples of candidate operator interface attributes; they are not meant to convey optimal design solutions for any function.

101.3 Multi-UAV Interface Concepts for Vehicle/Mission Management

This section will focus on candidate interface concepts for supporting an operator's tasking in managing multiple UAVs. First, *display concepts* will be addressed, starting with a description of displays designed for the Vigilant Spirit Control Station. This will be followed by an introduction to several novel display technologies, including view management, glyphs, timeline formats, and perspective displays. Finally, nonvisual display approaches, tactile and auditory, will be discussed. This section will conclude by providing several *control concepts* for vehicle/mission management, starting with techniques employed in the Vigilant Spirit Control Station. Novel control concepts under development, including speech-based control and gesture input, are also described.

101.4 Display Concepts for Vehicle/Mission Management

Primary Displays. Figure 101.1 depicts a representative mission management display configuration from the Vigilant Spirit Control Station (Rowe et al. 2009). Each panel or window can be considered a *tool* by which the operator is provided information to perform mission management functions. These include, for instance, a vehicle alert and summary tool and a tactical situational display (TSD) to provide advanced mapping capability. The *vehicle alert and summary tool* (along the left in Fig. 101.1) is intended to provide a quick look assessment of pertinent UAV information tailored to the current mission phase. To facilitate a quick crosscheck, one row per UAV is provided at the top listing basic navigation parameters (Fig. 101.2). Below that, dedicated panes are presented for each UAV. An analogy is that each UAV has a baseball card depicted with several information elements such as the vehicle's identification number (1, 2, etc.), speed, heading, altitude, estimated time of arrival to the next waypoint, operating mode, and payload information (sensor cameras, weapons, or radar systems). Various alerting mechanisms are also provided in this tool to indicate system faults such as loss of communication, global positioning sensing, low fuel/battery life, and other mission-specific alerts. Static symbology is used to represent the vehicle; however the concept of glyphs (explained below) may provide a more rapidly interpretable graphical representation of vehicle status that minimizes the use of alphanumeric. Advanced vehicle caution and warning alert systems might also be designed to immediately convey the operational ramifications of the contingency (e.g., sensor inoperable above 30,000 feet) to ease the prognostic requirements on the operator.

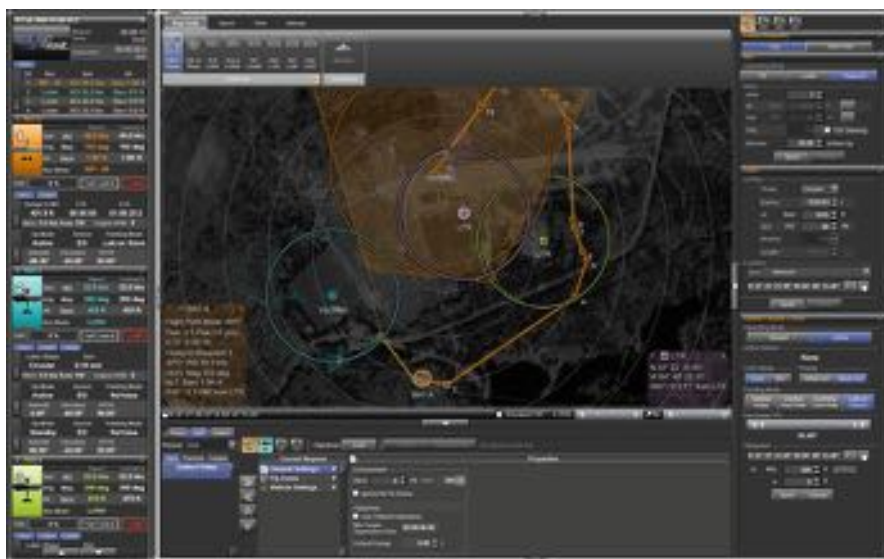


Fig. 101.1 Illustration from Vigilant Spirit Control Station

Fig. 101.2 Sample vehicle alert and summary tool from Vigilant Spirit Control Station



In future multi-UAV supervisory control scenarios, a map display will be considered a primary operator interface for both control and display of information. In the center larger window of the Vigilant Spirit Control Station, a *tactical situation display* (TSD) provides a common mission operating context consisting of spatially relevant information and geo-referenced imagery (Fig. 101.3; Rowe et al. 2009). Specific mission-related information is presented including each UAV’s position, air and ground tracks, airspace management aids, mission plans, and, if available, sensor viewing locations (footprints) and target locations. The mapping functionality of the TSD is analogous to using standard commercial mapping tools available on the Internet. Figure 101.3 shows a representative TSD format.

One challenge associated with mission management displays is how to best link information associated with each vehicle across multiple windows (i.e., utilizing the concept of visual momentum; Woods 1984). Color is used for this purpose throughout the Vigilant Spirit Control Station to uniquely identify each UAV and

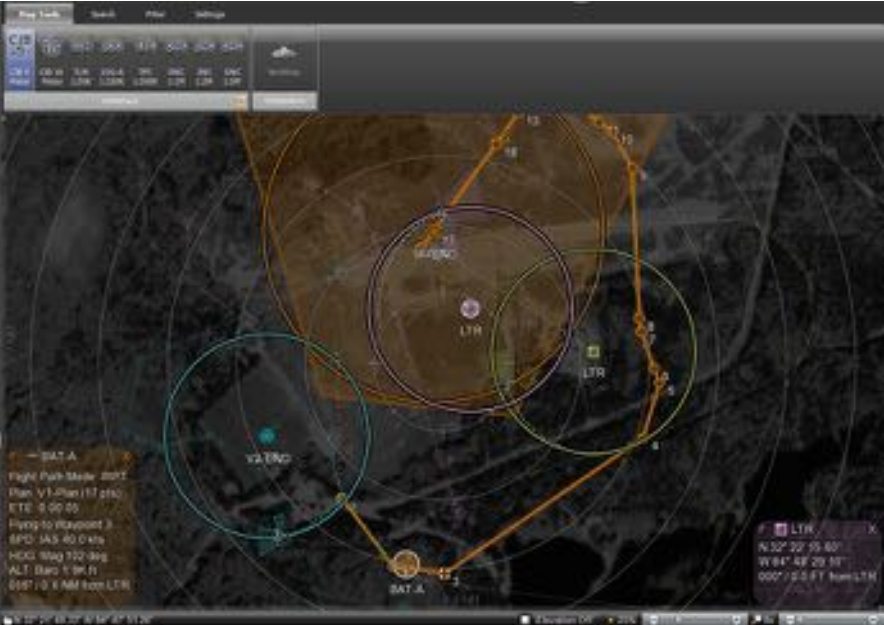


Fig. 101.3 Sample tactical situation display format from Vigilant Spirit Control Station

link its associated data such as flight plans, loiter locations, and sensor information. For example, the route symbology for a particular vehicle on the TSD is colored to match the color employed for that vehicle’s alert and summary tool.

View Management. Operators of multiple UAVs will be presented with an increased amount of information reflecting the multiple vehicles being managed. Moreover, more information will be available to be displayed, such as from the envisioned networked operations and existence of a Global Information Grid (GIG). Thus, a tremendous amount of information will be available, both in what feeds the data streams populating displays such as the TSD but also the additional information that can be presented via computer-generated symbology to augment or highlight displayed information. As one example, there are several candidate synthetic vision symbology concepts proposed to aid an operator to land UAVs as well as achieve sense and avoid functionality, including integrated, multidimensional conflict avoidance (Tadema 2011). As a result, it is likely that information elements will overlap, contributing to the visual clutter and impacting the information’s visibility.

There are numerous techniques for *view management* which maintain visual constraints on the projections of objects on a display (Bell et al. 2001). By making an adjustment in the manner information is presented, problems with information elements occluding each other can be minimized. Likewise, a more sophisticated intelligent system can ensure that information does not become illegible, and labels are automatically reoriented and repositioned based on the operator’s viewpoint.

Besides managing the view to optimize the visibility of the information, an intelligent system can highlight information that is critical for operator attention. Identifying useful coding methods to indicate qualifiers or meta-information (Bisantz et al. 2009) such as the criticality, urgency, source, timeliness, and reliability of information is one topic needing further research.

Another research area is determining how best to manage what information is displayed at any one time (St. John et al. 2004). The Vigilant Spirit Control Station has some limited filtering options, for instance, based on phase of flight and whether to hide labels. However, it would be more useful to have an intelligent multi-agent system that is capable of taking into account the real-time location and tasking of each UAV as well as current mission goals and environmental conditions. In this manner, decisions on what information to present (versus hide), as well as what information to highlight (versus de-emphasize), can be determined. For a multi-UAV application, it may be that the information detail needs to correspond to which UAV requires more dedicated attention at that moment. For example, if one vehicle's health degrades, additional information pertaining to just that one vehicle would be presented.

While having information presentation driven by context will result in less cluttered displays, it also increases the likelihood that information that the operator desires is not currently being displayed. Thus, any application of an intelligent de-cluttering system needs to include efficient means by which the operator can access *all* information elements. Likewise, the degree to which operators can control features (e.g., opacity) of information elements needs to be determined so that mission effectiveness increases without imposing additional workload (or attentional tunneling).

Glyph Symbolology. Presenting information in a more compact and/or integrated manner can also reduce display clutter. One approach involves mapping individual dimensions (variables) for a given entity to attributes of a particular multi-attribute graphical shape or symbol, such that variations and anomalies among these graphical entities are readily perceived by the operator (Chuah and Eick 1998; Whitaker and Thomas-Meyers 2006). This will conserve display space in that contextually relevant systems/mission status information is integrated into the same symbol and, as a result, it streamlines monitoring tasks by reducing the operator's need to visually scan and integrate information (Chuah and Eick 1998). The design of MIL-STD-2525B symbolology (DoD 2005) follows this convention, and modifications to this symbolology set have the potential to increase the saliency of state changes even more. Alternately, design of *glyphs* or *ymbicons* (hybrid symbolology combining best elements of symbols and icons; Smallman et al. 2005) that are tailored for multi-UAV control might be useful in maximizing information presentation.

Design and evaluation of glyph prototypes are underway at AFRL. As an example, one glyph prototype is shown in Fig. 101.4. For this modified MIL-STD-2525B symbol, many parameters of a single vehicle are depicted including type and characteristics of the vehicles. The vehicle's heading, pitch, and altitude (in thousands of feet) are also presented. The left and right bars depict, respectively,

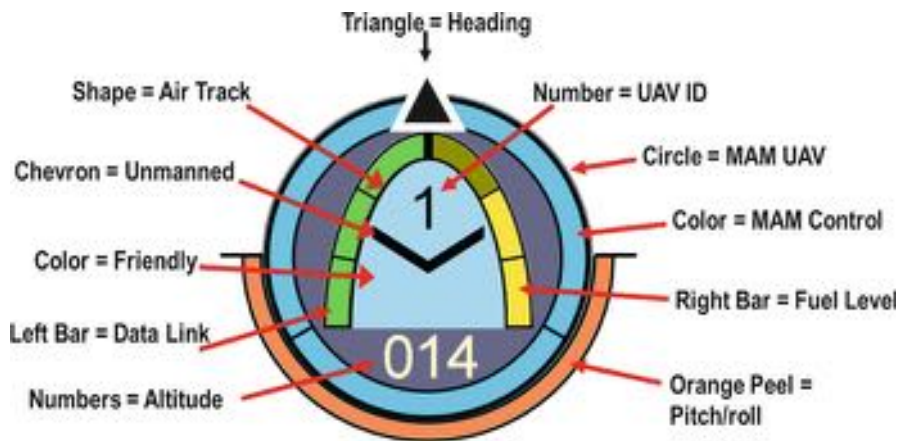


Fig. 101.4 Notional glyph showing UAV information



Fig. 101.5 Notional glyph showing time until UAV handoff

the data link and fuel status. The shade and color of these bars change, in real time, to reflect the vehicle’s state. Although this is presented simply as an example, it is clear that well-designed glyph symbology has the potential to reduce (1) the workload associated with operator monitoring of multiple vehicles and (2) the delay in reacting to certain system anomalies.

Multi-UAV applications in which an operator hands off control of a specific vehicle to another control station (e.g., to address a system malfunction or a more demanding dynamic mission) might also benefit from a glyph display approach. Figure 101.5 shows one example of how the status of a pending handoff might be depicted.

When selected, a glyph can also be appended with symbology and readouts that provide additional information. In Fig. 101.6, the first glyph shows readouts along the left that pertain to a UAV’s call sign, heading, altitude, speed, fuel, and frequency. To the right are some selectable icons for tools to manage the route and handoff. The second glyph example shows one method of how caution and warning alerts might be presented to operators.

It is important to note that the glyphs depicted above are all notional examples, presented simply to convey the potential benefits associated with this



Fig. 101.6 Notional glyphs with additional information displayed when selected

design approach. Systematic evaluations remain to be accomplished, especially for candidate multi-UAV control applications. Besides depicting the status of vehicles, a glyph approach for information presentation might also be useful to keep track of payload and/or progress on task(s). Research in this area needs to evaluate the extent to which the symbology rapidly and accurately conveys required information, especially when the symbol is copresented with other required information on an integrated control station display.

Timeline Interface. Another display concept that may facilitate vehicle/mission management for multi-UAV applications is one that focuses on the temporal aspects of a mission, such as a *timeline interface*. Critical information regarding all relevant UAVs and their respective tasking is presented on a single temporally referenced display. Such a presentation may improve simultaneous awareness of the current and projected status of all vehicles and thus improve the operator's ability to meet temporal constraints imposed by changes in the mission or cooperative missions. A timeline display is hypothesized to especially improve level 3 situation awareness (Endsley 2000): the projection of the situation into the future and the corresponding proactive management of resources against future mission needs. Thus, a temporal display may be very useful for providing time-based information needed for multi-UAV supervisory control, in addition to a map-based display and other status indicators.

Presentation of information on a timeline has been considered for numerous military applications (Scott et al. 2006; Rousseau et al. 2007). Additionally, temporal displays specific for UAV supervisory control applications have been designed and evaluated. Hanson et al. (2004) describe a task-based panel that supports synchronization and analysis of mission plan changes for UAV teams. As part of a UAV supervisory control simulation interface, Cummings and Mitchell (2005) designed a decision support display that included timelines for multiple UAV missions. Information was plotted as color-coded blocks across time to depict scheduled tasks, mission events, waypoints, windows of opportunity, and estimated time of arrival at targets. A later study evaluated the utility of augmenting each UAV's timeline with a configural decision aid referred to as StarVis (Cummings and Brzezinski 2010). Variables related to a target acquisition task (scheduling problem type as well as number and priority of targets impacted) were mapped to features of

a single geometrical form. The aid included a what if functionality to show effects of any control input under consideration. The results showed the value of the aid in controlling an individual UAV. However, when the aid was used to represent changes across all UAVs, performance dropped. These results demonstrate the importance of not saturating operators with information and that displays should not only help operators detect and identify problems, but also solve problems (Cummings and Brzezinski 2010).

More recently, a timeline prototype was developed and integrated into AFRL's Vigilant Spirit Control Station (Fig. 101.8). In the mission view (Fig. 101.7, upper lines), each horizontal bar represents the start and end times for a mission, and the bar color denotes which vehicle is performing that mission. In the vehicle view (Fig. 101.7, lower portion), a horizontal line is devoted to each vehicle with symbology overlays pertaining to the tasking and status of the vehicle's systems (e.g., ready to deploy, performing a mission, and fuel shortage). An experiment was conducted in which participants completed trials with and without the timeline interface. Documentation is still in preparation, but the results showed that performance on tasks involving temporal awareness (e.g., judging which UAV to assign to a pop-up mission) as well as multiple secondary tasks (e.g., communications, change detection, and information retrieval) was improved when the timeline interface was included in the control station design. Effort is now being devoted to expanding the timeline to better support the supervisory control of time-coupled tasks required in multi-UAV collaborative missions. This new design will also employ glyphs to represent vehicle and task status as well as a task-centric timeline versus the vehicle-centric approach illustrated in Fig. 101.7.

To support the development of a timeline interface prototype for multi-UAV control, evaluations were also conducted on specific design issues. For instance, one evaluation addressed how the display should be oriented (Spriggs et al. 2010). The majority of timeline designs are oriented horizontally, following the conventional

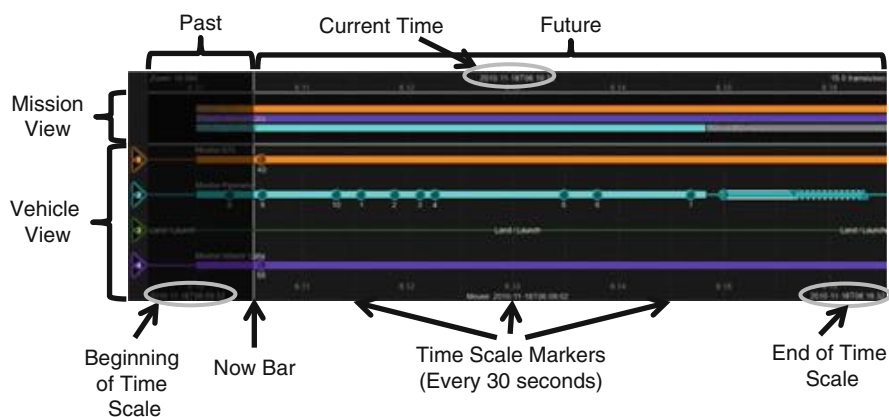


Fig. 101.7 Prototype timeline interface integrated into Vigilant Spirit Control Station

Fig. 101.8 Example of using color shading to code terrain elevation



approach used for timelines and program management schedules. For a multi-UAV application, this suggests that each horizontal line represent an individual vehicle. Since Western culture involves reading from left to right, the horizontal timeline may encourage attention to be focused on a single vehicle and interfere with the ability to complete a vertical scan for detecting temporal conflicts across vehicles. If the timeline is oriented vertically, left to right scans might provide more efficient retrieval of the status of all vehicles for a specific time slice. Moreover, a vertical orientation may benefit multi-UAV coordination tasks across time. However, the results from a part-task experiment showed little performance differences between horizontal and vertical views of a multi-UAV timeline.

Displays for Representing Third Dimension. The display technologies described above focus on how to present vehicle/mission information in two-dimensional (2-D) graphical formats. There are also other techniques for depicting relevant information in three dimensions that need to be evaluated to determine their utility for multi-UAV supervisory control applications. Terrain shading on the map may be beneficial for detecting hazardous terrain that can compromise vehicle safety (Rowe et al. 2009; Fig. 101.8). When relative altitudes are of primary importance, a vertical situation display (VSD) may be beneficial. These displays represent the vertical dimension in a veridical manner, and examples are available (see Cummings et al. 2007).

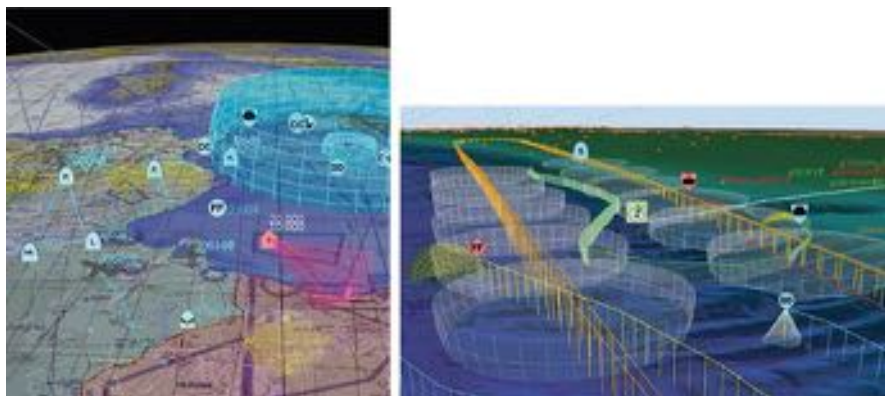


Fig. 101.9 Sample perspective prototype views showing tactical common operating picture (Analytical Graphics, Inc. 2006). http://www.agi.com/downloads/products/by-product-type/applications/stk/stk-for-c4isr/0305_T3DCOP_case_study.pdf

Both TSDs and VSDs provide planar views of the operating environment, with each display type sacrificing information along a hidden dimension. *Three-dimensional (3-D) perspective displays* offer the potential to provide all three dimensions to the operator in one integrated display (see Fig. 101.9, for example; Analytical Graphics, Inc. 2006). With perspective displays, the image is actually in two dimensions, but by employing an oblique viewing angle, all three dimensions can be represented to a greater or lesser extent. In considering a perspective display for UAV applications, it is important to first consider the possibility of naïve realism in assuming that the operator's ability to extract information is correlated with the realism of the display (Smallman and St. John 2005). In fact, some research to date has shown that perspective views typically do not enhance overall situation awareness and only benefit certain specific tasks (Smallman et al. 2005). This is most likely because the perspective distorts some information by foreshortening (depending on the angle of the perspective) and introduces ambiguities by confounding all three dimensions. For localizing objects and determining the relative distances between objects (important tasks for multi-UAV situation awareness), research has shown that a 2-D coplanar image presentation is better than a static perspective view (St. John et al. 2001). Even the displayed symbology can be more difficult to interpret in a perspective view due to distortions introduced by perspective. However, the use of contour lines and grid lines for terrain may improve the utility of a perspective view, as could the use of drop lines and drop shadows for depicting the relative altitudes of vehicles (St. John et al. 2001). Additionally, the benefits of perspective displays will likely increase if the operator is able to control the view angle dynamically (Dao et al. 2006). The benefit of a perspective view for certain tasks, though, may be sufficient rationale for its inclusion in control stations. For instance, Veltman and Oving (2003) determined that a 3-D map improves UAV camera control and related task performance. The authors recommend that the operator be able to choose

between 2-D and 3-D views. In this manner, the 3-D view would be available on demand either as a mini-view to augment orientation or as a result of morphing from a 2-D view to shallower viewing angles (Smallman et al. 2005). It may also be useful to present multiple views simultaneously with picture-in-picture techniques in which an alternate view is presented in a smaller window that overlays a portion of the TSD. This enables the operator to benefit from both views: the 2-D view for *veridical global information* (the position of objects relative to other objects in the earth reference frame) and the 3-D view for *relative local information* (position of objects relative to the viewing position of the camera). Advances in *true* 3-D display technologies that create a true sense of depth by imitating one or more of the visual depth cues may also accelerate employment of 3-D displays in UAV control stations (McAllister 2002).

The above concepts involve presenting information visually. Since multi-UAV supervisory control will require significant visual demands, any display and/or control approach that minimizes the visual load for both acquiring information and commanding actions may improve task performance, as predicted by the Multiple Resource Theory (Wickens 2002). For instance, *multisensory interfaces* may enhance time-sharing (pertinent information from different sources comes through different sensory channels simultaneously) and improve attention capture (two or more senses transmit identical or supporting information simultaneously). Additionally, such interfaces may result in a more intuitive presentation of information.

Tactile Displays. *Tactile presentation of information* is a promising display technology (van Veen and van Erp 2003). Information is transmitted through the skin by varying one or more dimensions of vibratory stimulation (locus, amplitude, frequency, duration, and spatial/temporal separation). Tactile messages are silent and unobtrusive, and their displays are omnidirectional in that the operator does not have to be looking at a particular location to be alerted. They can also be designed to convey spatial information (e.g., as implemented in a belt or vest; see Draper et al. 2007). Additionally, tactile information can be received simultaneously to visual and aural information.

The use of tactile displays as a means of informing UAV operators of serious system malfunctions was evaluated (Calhoun and Draper 2006). In these evaluations, small electromechanical vibrating factors were mounted in elastic bands over one or both inner wrists (Fig. 101.10; Calhoun et al. 2005; Calhoun and Draper 2006). In the test scenarios, the tactile cues were reserved for critical warnings to maximize their saliency (Wogalter and Leonard 1999). In a series of simulation evaluations, the results demonstrated that tactile cuing was an effective redundant alert (paired with a visual alert) and a suitable substitute for an aural redundant cue. Furthermore, operators reported the tactile cues were beneficial and that the stimulation was not uncomfortable or a distraction in attending to other tasks. However, analysis of the performance data did not show an advantage for tactile alerts over aural alerts. Further research is required to determine if tactile cues would prove more effective than aural cues in noisy environments when each is employed a sole alert, rather than redundantly with a visual alert. It is also possible that tactile



Fig. 101.10 Tactile wrist pads as high priority alert cue

alerts may be especially useful in noisy environments when long vigilance durations are required, a condition not tested to date.

Aural Displays. The *aural alerts* used in the evaluations described above were monaural and failed to leverage the natural spatial auditory processing capabilities of humans. It is possible that multi-UAV supervisory control can benefit from the ability of operators to determine the spatial location of sound sources and monitor events at multiple locations simultaneously (Simpson et al. 2004). Advances in spatial auditory display technologies have made it possible to recreate and present to an operator the spatial information that would naturally be available in a real-world listening environment. For instance, spatial audio could be used to assist operators in finding items of interest in the spatial environment by having the cues emanate from directions that the operator should attend to (Bolia and Nelson 2003). Spatial audio cueing might also improve speech intelligibility and reduce workload for operators listening to multiple radios or intercom channels (Bolia and Nelson 2003). This could be useful to operators who often engage in verbal communications with a variety of distributed team members, not only because of its potential to improve communications effectiveness but also because spatial awareness of talker location would provide the operators with an additional cue to the identity of the talkers based on a predetermined mapping of communications channels.

The use of *spatial audio* has been combined with other features in a Multi-Modal Communication (MMC) network-centric communication management suite (Finomore et al. 2011). With the goal of improving the communication performance of command and control operators, this tool provides an enhanced ability to manage communications from voice and text-based systems in a single intuitive dynamic display. MMC captures, records, and displays radio and chat communication to the operator, providing instant access and full control over all current and past information exchanges. Moreover, there are additional functions that may be useful for multi-UAV control. For example, the ability to perform searches across communication and chat channels for information and set up alerts for keywords that are instrumental to the mission could ease the operator's task of monitoring multiple streams of communication as well as other displays.

101.5 Control Concepts for Vehicle/Mission Management

The previous section focused on information *display* concepts for vehicle/mission management. This section will address associated vehicle/mission *control* technologies, including direct manipulation concepts, speech-based control, delegation control, and gesture commands that may be useful for multi-UAV applications. Once again, it is important to note that the operator control concepts described in this chapter should be viewed only as examples of candidate interface attributes; they are not intended to convey optimal design solutions for any application.

As with the display concepts described earlier in this chapter, the controls currently employed in the AFRL's multi-UAV Vigilant Spirit Control Station will be described first. This station was developed with the assumption that all vehicles are highly automated, enabling the operator to focus on tasks involving the payload (e.g., monitoring each UAV's video; Rowe et al. 2009). Thus, for this supervisory control, there is no hardware for Hands on Throttle and Stick (HOTAS) control. Rather, control is accomplished via higher level commands using the command and control tool presented on the right side of the ground control station (see Figs. 101.1 and 101.11). With this tool, controls are conveniently located for a variety of tasks: vehicle navigation, loiter management, sensor payload management, and dynamic mission planning. Besides exercising control functions with defined keyboard and mouse inputs, the station also supports alternative control techniques such as speech-enabled control and touch inputs. All control inputs are linked to associated information on other displays such as the TSD.

Direct manipulation control concepts are also supported such as interacting directly with TSD symbology to quickly issue various commands. This type of human-machine interaction is considered *direct manipulation* because the operator is directly manipulating a visual representation that metaphorically represents the actual system or task (Shneiderman 1987). Typically, inputs using direct manipulation are viewed as more natural or intuitive. Control may also be more rapid because the interaction taps the human's analogical reasoning power when determining what actions to take when executing the task. Also, the results of the interaction are typically immediately visible. For example, by selecting a symbol representing a UAV on the TSD, a context-sensitive suite of commands can be called up. Using this tool, the operator can manipulate the constraints and parameters of a particular command directly on the geo-registered map, which greatly simplifies the input. As an example, see Fig. 101.12. Here manipulations of the Steering Command Widget symbology on the TSD enable the operator to quickly vector a UAV in response to air traffic control commands or away from a potential threat, such as restricted airspace or nearby air traffic. To designate a new heading for the UAV, the operator selects and drags the vector, and the new heading is indicated by the position of the vector on the compass rose, as well as a readout in the heading box. If the vector is within the compass rose, the heading value changes in 5° increments. When the vector is longer, outside the compass, the heading changes by a single degree. Once the desired heading value is indicated, a button is selected to send the heading to the vehicle control system.

Fig. 101.11 Sample command and control tool in Vigilant Spirit Control Station



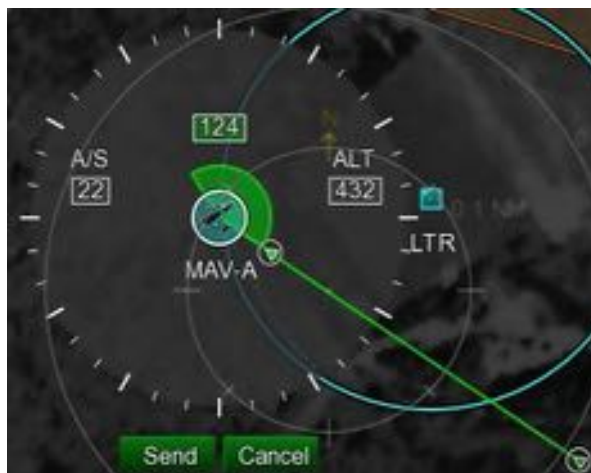


Fig. 101.12 Sample UAV Command Widget selectable on tactical situation display in Vigilant Spirit Control Station

Two of the AFRL-developed multi-UAV display concepts described earlier support direct manipulation. First, the prototype timeline display (Fig. 101.7) was designed to also have control functionality. Control inputs to manage the vehicle's actions can be accomplished by clicking directly on symbology overlaid on the UAV's respective timeline (e.g., to command the vehicle to return to base). Also, pop-up missions can be assigned to an individual UAV with a drag-and-drop action, moving the task from the mission view to the respective UAV's line in the vehicle view.

A second example is the glyph display concept. A glyph display element could be implemented such that hovering over it with the mouse calls up a polar menu of tools, configured dynamically based upon current context, for the operator to select (Fig. 101.13). This arrangement minimizes the distance between the glyph and respective tool selection and benefits from the advantages of direct manipulation.

Many of the above-described control actions involve selection of symbology elements with the mouse. With a touch screen system, inputs can be accomplished by directly touching the display surface with one or more fingers (depending on the type of hardware). This type of input may also be applicable for a multi-UAV ground control station. One concern is to ensure that the location/angle of the touch screen does not impose fatigue issues. More importantly, the nature of the inputs needs to be carefully evaluated. Operations that involve the entering of values and/or text with touch are less ideal. In general, touch screens are best suited for information retrieval (involving the pointing of items or an area on the display). Limited manipulations of displayed information (e.g., redefining a path) can also be accomplished. For example, the drawing tools and display functions of the FalconView mission planning system were applied in an evaluation to enable sketch-based inputs of

Fig. 101.13 Notional glyph with associated menu of tools



ingress paths, search regions, and other tactical areas of interest for UAV-based munitions (Lewis et al. 2006). Preliminary results indicated the control was intuitive, but defining inputs for each individual UAV was limiting; functionality to enable assignment of routes to more than one UAV at a time is desirable. Durlach et al. (2006) evaluated a prototype whereby a UAV was controlled by touching a location on a sensor image. This research demonstrated several issues to address including the need to have adequate feedback that the touch input has registered, as well as the difficulties in translating a single-touch input into three-dimensional commands to a UAV. Additional design guidelines for touch panel keypads and keyboard are available (Plaisant and Sears 1991; Scott and Konzola 1997).

For many data entry task types, *speech-based control* may provide a much more efficient input mechanism for multi-UAV control. While machine performance is still inferior to human performance, (impacted by numerous factors such as the size of the vocabulary and noise environment), increasingly sophisticated automated voice recognition systems have made UAV systems responding to voice commands progressively more feasible. The operator's commands would need to be simple though, as the ability for computers to derive meaning by analyzing linguistic phenomena (natural language processing) remains an immature technology. Nevertheless, categorical commands can substitute for many manual inputs in UAV operations. In contrast, commands to transmit continuous analog spatial information, particularly in dynamic situations, are less efficient. And there are other guidelines to be considered in implementing speech-based control, such

as to employ a speaker independent, continuous speech recognition system that requires no prior system training (Rudnicky 1996; Gardner-Bonneau 1999; Draper et al. 2007). Implementation should also consider providing visual feedback of each spoken command and the use of a conveniently located push-to-talk button to activate the speech recognition system.

AFRL conducted an experiment examining the utility of conventional manual input versus speech input for a variety of tasks performed in single-UAV control (Draper et al. 2003), and it was demonstrated that speech input was significantly better than manual input in terms of task completion time and accuracy, maintenance of flight/navigation path, and operator subjective ratings. Across tasks, data entry time was reduced by approximately 40 % with speech input. This performance advantage, coupled with the other benefits of speech input (e.g., head-up and hands-free control), warrants additional research to confirm that speech input is also beneficial in operational control station auditory environments and does not conflict with intercom operations and intra-crew communications.

An even more promising application of speech-based control for multi-UAV supervisory control is its use in a delegation approach to interact with automated systems. With *task delegation*, such as Playbook[®], the operator communicates with UAVs by using a compiled set of plans or plays with short, easily commanded levels that can also be modified as needed. This approach is designed to minimize the workload associated with delegating a plan, much like a quarterback uses establish plays to communicate to a football team (Miller et al. 2004; Miller and Parasuraman 2007). For multi-UAV applications, this delegation approach may provide a control architecture whereby the operator can flexibly change the tasking and role of automation during the course of a mission, by specifying various constraints, stipulations, contingencies, and alternatives to the automation, all within the context of operationally relevant plays. AFRL is in the process of evaluating this approach for UAV control and providing the ability to seamlessly transition between different levels of control, ranging from a level where the operator's manual inputs command a vehicle's path to higher level plays (e.g., "monitor target here" [and *here* is designated with a touch input on the map]) as well as intermediate levels (e.g., quick temporary maneuvers with a verbal command "UAV 1, hook left"). To date, operator feedback has indicated that this delegation approach has definite value for future multi-UAV applications (Calhoun et al. 2012).

The control technique of issuing a voice command in conjunction with indicating a location or item via touch or selection with a mouse increases overall functionality by using multiple input devices (Oviatt 1996). A *spatial dialog* interaction and communication technology has been proposed to apply *multimodal interactions* to improve remote virtual presence and collaboration across all elements of a distributed command and control team, including the autonomous vehicles (Chun et al. 2006). With spatial dialog, TSD inputs that involve a spatial component can capitalize on the integration of speech recognition, gesture interaction, and touch panels. The operator communicates plans and queries to the autonomous unmanned systems with speech commands that include temporal and spatial (e.g., *here* and *there*) parameters while making gesture inputs on the display (Chun et al. 2006; Green et al. 2007). A particular information element or location can also be tagged

by the operator with a voice name such that when the operator states that name again later with a verbal action command, the system understands which element to apply the action to (Batkiewicz et al. 2006). More lengthy voice memos can also be associated with specific information elements. These voice notes and annotations are an intuitive way to rapidly capture operator observations regarding an event or location which may be beneficial to enhance information communication with all entities in a networked operation space or to increase the situation awareness of a new operator when UAV control is handed over.

Besides using a simple gesture input to indicate a specific location or information element, more complex gesture commands can be used to provide additional control options. With this implementation, the gesture indicates not only the subject/focus to which the command applies, but the action to be performed. For instance, making an input in the shape of a Z (for zoom) with the stylus movement can command a zoom action as well as indicate the direction of the zoom. Once the pen or finger is released, the zoom would stop (Chatty and Leocoanet 1996). Two-handed graphical interaction has also been proposed, with the dominant hand assigned to drawing, while the nondominant hand moves the mouse to select tools, lock objects, zoom, etc. (Leganchuk et al. 1998). This would enable hold-and-pull interactions and simultaneous designation of two information elements. With the recent increase in products that enable inputs with a finger tap, pinch, swipe, etc., it is very timely to evaluate gestural input for interacting with multi-UAV control station displays. For further information, see Wigdor and Wixon (2011).

101.6 Multi-UAV Interface Concepts for Payload Management

Many of the interface concepts described above for *vehicle/mission management* may also be useful for *managing the payloads* of UAVs. For instance, use of glyphs for *displays* might be useful to provide an at-a-glance indication of payload status. For *control* functionality, an example is the use of a verbally commanded play that triggers the features of a sensor to be automatically configured appropriately for the planned mission. In fact, with highly autonomous vehicles, it is envisioned that a verbal goal-directed command specifying the objective in terms of the payload (e.g., what target to image) could result in automatic routing of multiple vehicles, automatic configuration of respective payloads, as well as completion of other tasking required to accomplish the mission (Calhoun et al. 2012). As these two examples illustrate, it is recommended that the concepts already described in this chapter also be considered as to how they might be applied to support payload management.

The remainder of this chapter will focus on other display/control concepts relevant to payload management. Many of these involve the use of overlaid computer-generated symbology. Others provide tools tailored to manage video streams. Consideration of these novel approaches is particularly important managing the multiple payloads afforded by a multi-UAV application environment.

Each UAV will have at least one associated payload, and these payloads can also vary by type (e.g., electro-optical, infrared sensor). The Multi-UAV Supervisory Control Interface Technology (MUSCIT) demonstrations have shown that the operator's visual load and attentional demands are often driven by the task of sensor management (manipulating the sensors and detecting/identifying targets of interest; Patzek et al. 2012). This workload also increases substantially with the number of UAVs being supervised, potentially resulting in a significant decrement in task performance. Therefore, interface designs need to support this highly critical and demanding sensor management task within the overall context of flight/mission management, for instance, by colocating or even integrating critical mission and flight information with the sensor management interface to reduce visual scan demands.

One approach to ease the sensor management task itself is to overlay relevant information elements directly on the imagery, as described by Hunn (2006). Calhoun and Draper (2009) report on the impact of *overlaying synthetic database information on video imagery* for several UAV operator tasks. The results showed that the synthetic symbology can improve situation awareness, reduce workload, and improve the designation of points of interest at nearly all the update rates evaluated (Draper et al. 2006). A *picture-in-picture construct* can also be applied to integrate real and synthetic imagery. Calhoun et al. (2007b) have shown the value of condensing the real-world video imagery such that it occupies a portion of the display area and is framed with a synthetic generated imagery border (Fig. 101.14). With the synthetic imagery aligned and registered with the UAV video imagery, the result is a virtual expansion of the available sensor field of view (well beyond the physical limits of the camera). This increase of spatial awareness potentially mitigates the UAV camera's narrow field of view, the basis of what has been referred to as the soda straw analogy or keyhole effect, a critical issue with UAV operations (Woods et al. 2004).

Multimodal displays and controls can also assist operators in focusing on the imagery from the sensors. For instance, with speech recognition, operators can request information about a system and then hear the information reported via speech synthesis, without changing eye gaze direction (Patzek et al. 2012). Aural and tactile alerts can also be employed (Calhoun and Draper 2006).

In the AFRL's Vigilant Spirit Control Station (see Rowe et al. 2009), several interface concepts designed to aid operators in managing multiple payloads are under evaluation. As shown in Fig. 101.14, all the videos are located adjacent to one another in the sensor management area of the control station. Functionality is also provided to control the view of the sensor feeds. For instance, the sensor can be placed in a latitude/longitude slaved mode or can be steered with the mouse/cursor via a point and click steering method. Zooming can be controlled with the mouse's scroll wheel. The rationale for *integrating the mouse sensor steering* was to reduce control time and errors in UAV video selection/manipulation and alleviate the need to switch between two input devices (e.g., a sensor stick and mouse).

Controls to manipulate the respective sensor presentations using a *digital video recorder* (DVR) have also been implemented. The DVR enables a playback

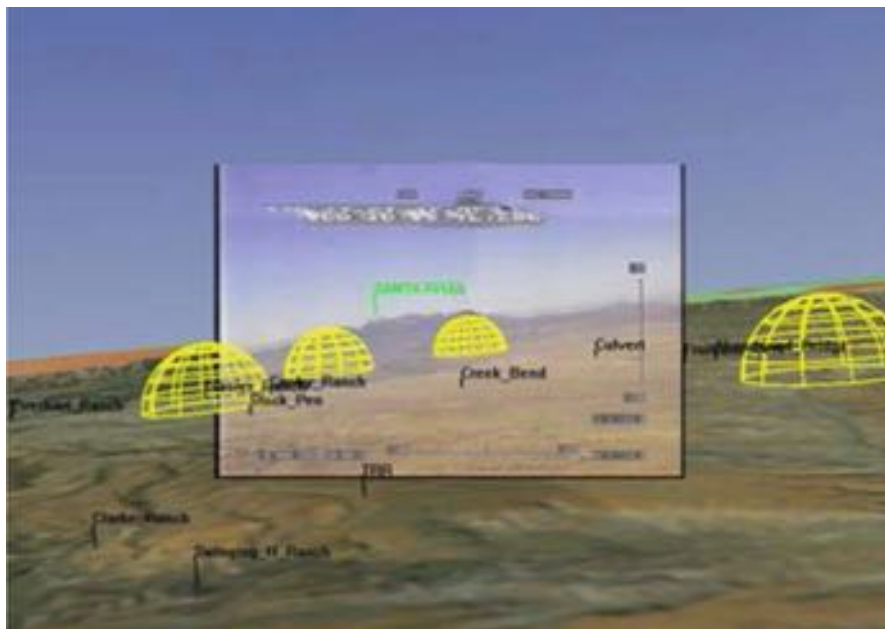


Fig. 101.14 Picture-in-picture (PIP) concept, with real video imagery surrounded by synthetic generated terrain imagery and symbology

capability, allowing an operator to reexamine any portion of a video (see Fig. 101.15). This capability to capture a segment of video to allow a second look and confirmation process may be especially useful if the vehicle's platform is such that a more stabilized view of the target is desired (Feitshans et al. 2008). Another potentially beneficial feature is the ability to “bookmark” a video segment by clicking directly in the video to designate locations of interest. When a location is designated, a computer-generated flag is overlaid on the imagery to mark the location of the tagged target. The color of this virtual flag corresponds to the assigned color of the UAV/sensor feed from which it was tagged. These virtual flags are persistent across sensors in that a flag generated in one sensor video is seen in another sensor video if that flag falls within the second sensor's field of view. In addition, a flag is placed on the map to correspond to the geo-location of the tagged object, color-coded to correspond to the UAV/camera source. These tools require further evaluation to ensure that they enhance overall mission effectiveness rather than negatively change attention dynamics (e.g., operator being distracted by focusing on recorded materials and not adequately attending to real-time sensory imagery). Research at AFRL is also addressing how additional information (e.g., weather feeds) might be integrated with the goal of *fusing information* to maximize intuitive information transfer (Carretta et al. 2009).



Fig. 101.15 Right monitor shows sensor videos from four UAVs and associated control and DVR tools in a Vigilant Spirit Control Station



Fig. 101.16 View of digital recording tool implemented in Vigilant Spirit Control Station

The Vigilant Spirit Control Station depicted in Fig. 101.14 shows the imagery from four cameras, each an equal window size. The station can also be implemented to enable the operator to enlarge one of the sensor videos, for instance, by touching it (Durlach et al. 2006). A tool whereby the video being attended to (e.g., determined by mouse clicks in the window) automatically enlarges the window and reduces the operator’s need to manipulate the zoom function is also a possibility (Porat et al. 2011). To further *reduce the need for zoom manipulations* (and potentially reduce workload), a maintain coverage area tool may be useful that automatically manipulates the zoom level to maintain the desired imagery size of a designated location, regardless of the vehicle’s position in a maneuver (Oron-Gilad et al. 2011).

Figure 101.17 shows the Vigilant Spirit Control Station configured with one large window showing the video of the selected UAV, along with small thumbprints of videos from the other UAVs. Use of a larger window can facilitate inspection of

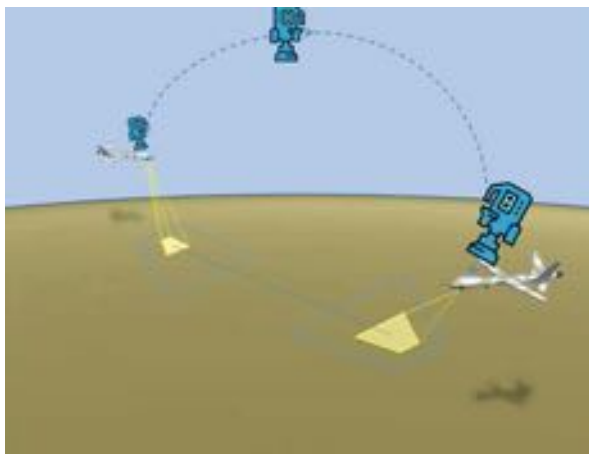


Fig. 101.17 Alternate Vigilant Spirit Control Station layout with both large and small video windows

the video. However, for this configuration (as well as any with multiple videos), switching attention between different UAVs and their associated camera views has the potential for negative transfer of context to occur, such that the specific information and tasking involved with the first video/mission might delay or degrade the operator's ability to effectively perform tasks with the new video/mission. For instance, if the operator has a mental model of friendly forces being south of the target in the first mission, will the operator inappropriately apply this mental model to the new mission? To help operators quickly acquire situation awareness associated with the new camera view, an automated aid was conceived that uses synthetic vision technology to present a gradual scene transition process. With this transition, first, the camera imagery for one UAV seamlessly fades into a synthetic imagery correlate of the real video image. The scene then follows a fly-out, fly-in metaphor over several seconds (similar to computer mapping tools), finishing with a transition back from synthetic to real video imagery at a new camera view of the new geographic area (Fig. 101.18; Calhoun and Draper 2009). During the semicontinuous transition, points of interest (landmarks, roadways, troop locations, etc.) are highlighted with overlaid, geo-registered computer-generated symbology. The results from a simulation evaluation provide support that an automated *camera view transition aid* can improve task performance and situation awareness (Calhoun et al. 2010).

Another approach to help operators manage real-time video feeds from multiple UAVs employs model-inspection techniques and extends the picture-in-picture concept of viewing a video feed in the context of surrounding terrain. With the SUAVE (simple UAV environment) experimental system, as UAVs stream videos, individual geo-referenced frames are sampled and projected to replace the texture of an existing height map of satellite imagery (Abedin et al. 2011). In that the actual video is not being viewed, the *sampled imagery fused* in this fashion is asynchronous with the ability to trade off spatial and temporal resolution. If data is sampled at high spatial resolution, then large regions can be populated and searched, although some data may be obsolete. Temporal resolution can be maintained by sacrificing spatial resolution with the UAVs covering areas at a higher frequency. Research to date indicates that use of this asynchronous 3-D terrain model where the *fused video*

Fig. 101.18 Conceptual illustration of transition display aid to acquire situation awareness in new camera view



can be viewed in context improves target identification performance compared to a synchronous (multiple video feeds) condition where information is presented to the operator as it is acquired.

The tools just described are especially supportive of operators where each UAV is imaging a different area. However, multiple UAVs can be tasked to image the same location, with the goal of maintaining persistent coverage of a target of interest. Given the movement of the UAVs, existence of obstructions, and other factors, the particular UAV that can provide the best view of the target at any given time will vary. A casting rays tool may aid such tasks by presenting *ray traces overlaid on the imagery*, one from each UAV to the location to be imaged (Porat et al. 2010). Multiple coding methods can be employed such that each ray trace presents additional information. For example, ray width may indicate sensor distance from the target (wider is closer), with line style indicating view quality (dashed is limited or obstructed line of sight), color indicating UAV source, and an adjacent angle symbol indicating the inclination of the sensor (a wide angle indicates susceptibility to occlusion). This additional information is intended to aid the operator in deciding which sensor to select for viewing the target while maintaining focus on the sensor imagery and reducing the time spent analyzing the situation on the map. Follow-on evaluations are underway at AFRL to expand this concept by implementing a *predictive vantage ray* tool which also includes coding to indicate when a sensor will likely acquire or lose a target, as well as a supporting timeline that indicates the estimated time slots each UAV can image the target of interest.

The ability to view a location from multiple UAV viewpoints is a definite advantage. However, with current sensor technology, this equates to having multiple soda straws that the operator needs to mentally integrate to acquire the big picture. While information on tactical maps that reflect the footprints of multiple sensors can help construct a mental model of the mission environment, the operator will still have to devote considerable cognitive effort to determine and understand the

relationships between multiple views. This is especially difficult considering the environment dynamically changes. With advancements in computer technology, it is anticipated that available data streams from multiple heterogeneous sensors and geospatial data can be fused with synthetic symbology to create a *virtual (acoustically, visually, etc.) world of the entire mission area of interest*. There are numerous challenges to be addressed in developing this virtual representation of the battlespace. For instance, how can live and virtual representations of the same operational environment be blended to provide an immersive rendering and yet the source of each information element can be determined? Effort is also needed to develop an intuitive interface by which the operator would interact with the telepresence virtual world. Ideally, the world will be viewable from multiple perspectives and the interface will enable the operator to move fluidly throughout the representation of the operational environment, making real-time adjustments to the assets contributing to ongoing missions.

101.7 Conclusion

Several display and control concepts are introduced in this chapter that may be applicable to multi-UAV control applications. Certainly, with advances in display and control technology, there are many more possibilities than those described here. With this plethora of candidate solutions, a designer might be inclined to focus attention on one promising technology and apply it (or a few) to all possible multi-UAV interface applications. This technology-focused approach, however, should be avoided as multi-vehicle control applications, and associated control station requirements, can vary widely and thus can be quite disparate. Instead, for each multi-UAV control application, a user-centered design approach should be utilized (e.g., Endsley 2011) with UAV operator involvement during all cycles. This approach parallels a systems engineering process with emphases on iterative user and task analysis, design prototype implementation, and testing. For task analysis, the multi-UAV operator's specific tasks, operational mission, and environmental constraints need to be taken into account. The design step involves creating candidate solutions (from rough concepts to a fully functional prototype) tailored to meet identified task needs and constraints. Regardless of the fidelity of the design concept, iterative evaluation is necessary within the context of the targeted multi-UAV application.

References

- S. Abedin, M. Lewis, N. Brooks, S. Owens, P. Scerri, K. Sycara, SUAVE: integrating UAV video using a 3D model. Proc. Hum. Factors Ergon. Soc. Ann. Meet. **1**(1), 91–94 (2011)
- Analytical Graphics, Inc., A case study: tactical 3-D COP (Battlespace Visualization for the U.S. Navy's GCCS-M System, 2006), http://www.agi.com/downloads/products/by-product-type/applications/stk/stk-for-c4isr/0305_T3DCOP_case_study.pdf

- T.J. Batkiewicz, K.C. Dohse, V. Kalivarapu, T. Dohse, B. Walter, J. Knutzon, D. Parkhurst, E. Winer, J. Oliver, Multimodal UAV ground control station, in *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA-2006-6963, 2006, 1–10
- B. Bell, S. Feiner, T. Hollerer, View management for virtual and augmented reality, in *ACM Symposium on User Interface Software and Technology*, Nov 2001 (ACM, New York, 2001), pp. 101–110
- A.M. Bisantz, R.T. Stone, J. Pfautz, A. Fouse, M. Farry, E. Roth, A.L. Nagy, G. Thomas, Visual representations of meta-information. *J. Cogn. Eng. Decis. Mak.* **3**(1), 67–91 (2009)
- R.S. Bolia, W.T. Nelson, Spatial audio displays for target acquisition and speech communications, in *Virtual and Adaptive Environments: Applications, Implications, and Human Performance Issues*, ed. by L.J. Hettinger, M.W. Haas (Lawrence Erlbaum, Mahwah, 2003), pp. 187–197
- Y. Boussemart, J. Las Fargeas, M.L. Cummings, N. Roy, Comparing learning techniques for hidden Markov models of human supervisory control behavior, in *AIAA Infotech@Aerospace*, Paper AIAA-2009-1842, 2009
- G.L. Calhoun, M.H. Draper, Multi-sensory interfaces for remotely operated vehicles, in *Human Factors of Remotely Operated Vehicles*, ed. by N. Cooke, H. Pringle, H. Pedersen, O. Connor (Elsevier, New York, 2006), pp. 149–163
- G.L. Calhoun, M.H. Draper, Unmanned aerial vehicles: enhancing video display utility, in *Human-Robot Interactions in Future Military Operations*, ed. by M. Barnes, F. Jentsch (Ashgate, Farnham, 2009), Chap. 13, pp. 229–248
- G.L. Calhoun, J.V. Fontejon, M.H. Draper, H.A. Ruff, B.J. Guilfoos, Tactile and aural alerts in high auditory load UAV control environments. *Proc. Hum. Factors Ergon. Soc. Ann. Meet.* **49**(1), 145–149 (2005)
- G.L. Calhoun, M.H. Draper, H.A. Ruff, Operator interface research testbed for supervisory control of multiple unmanned aerial vehicles (UAVs), in *Decision Making in Complex Environments*, ed. by M. Cook, J. Noyes, Y. Masakowski (Hampshire England: Ashgate, Hampshire, 2007a), Chap. 10, pp. 105–115
- G.L. Calhoun, H.A. Ruff, A. Lefebvre, M.H. Draper, A. Ayala, “Picture-in-Picture” augmentation of UAV workstation video display. *Proc. Hum. Factors Ergon. Soc. Ann. Meet.* **51**(1), 70–74 (2007b)
- G.L. Calhoun, L. Warfield, H.A. Ruff, S.E. Spriggs, N.F. Wright, Automated aid design for transitioning between camera views. *Proc. Hum. Factors Ergon. Soc. Ann. Meet.* **54**(1), 413–417 (2010)
- G. Calhoun, M. Draper, H. Ruff, T. Barry, C. Miller, J. Hamell, Future unmanned aerial systems control: feedback on a highly flexible operator-automation delegation interface concept, in *AIAA Infotech@Aerospace*, Paper AIAA-2012-2549, 2012
- T.R. Carretta, M.J. Patzek, L. Warfield, S.E. Spriggs, A.J. Rowe, A. Gonzalez-Garcia, K.K. Liggett, *Target Acquisition Involving Multiple Unmanned air Vehicles: Interface for Small Unmanned Systems (ISUS) Program*, AFRL-RH-WP-TR-2009-0021, 2009
- S. Chatty, P. Leocanet, Pen computing for air traffic control, in *Proceedings of CHI’96, Conference on Human Factors in Computing Systems* (1996), pp. 87–94, <http://sigchi.org/chi96/proceedings/papers/Chatty/sc-txt.htm>
- J. Chen, M.J. Barnes, M. Harper-Sciari, Supervisory control of multiple robots: human-performance issues and user-interface design. *IEEE Trans. Syst. Man Cybern. C* **41**(4), 435–454 (2010)
- M.C. Chuah, S.G. Eick, Information rich glyphs for software management data. *IEEE Comput. Graph. Appl.* **18**(4, July–Aug), 24–29 (1998)
- W.H. Chun, T. Spura, F.C. Alvidrez, R.J. Stiles, Spatial dialog and unmanned aerial vehicles, in *Human Factors of Remotely Operated Vehicles*, ed. by N. Cooke, H. Pringle, H. Pedersen, O. Connor (Elsevier, New York, 2006), pp. 193–208
- M.L. Cummings, A.S. Brzezinski, Global vs. local decision support for multiple independent UAV schedule management. *Int. J. Appl. Decis. Sci.* **3**(3), 188–205 (2010)
- M.L. Cummings, P.J. Mitchell, Managing multiple UAVs through a timeline display, in *AIAA Infotech@Aerospace*, AIAA-2005-7060, 2005

- M.L. Cummings, J.J. Marquez, M. Visser, *Shadow TUAV Single Operator Consolidation: Display Assessment*. Human Automation Laboratory Technical Report HAL-2007-01, MIT, Cambridge, Jan 2007
- A-Q. Dao, V. Battiste, S. Granada-Vigil, Evaluation of the usefulness and usability of cockpit situation display perspectives for ROV operations in approach civil air space. *Proc. Hum. Factors Ergon. Soc.* **50**(1), 161–165 (2006)
- Department of Defense, *MIL-STD-2525B: Department of Defense Interface Standard: Common Warfighting Symbolology*, Department of defence, Washington DC, Revised Change, 1st edn. (2005)
- Department of Defense, *Unmanned Systems Roadmap 2007-2032*, Office of the Secretary of Defense, Washington, DC, 2007
- M.H. Draper, G.L. Calhoun, H.A. Ruff, D. Williamson, T. Barry, Manual versus speech input for unmanned aerial vehicle control station operations. *Proc. Hum. Factors Ergon. Soc. Ann. Meet.* **47**(1), 109–113 (2003)
- M.H. Draper, G. Calhoun, J. Nelson, A. Lefebvre, H. Ruff, Synthetic vision overlay concepts for uninhabited aerial vehicle operations: evaluation of update rate on four operator tasks, in *Proceedings of the NATO RTO Human Factors and Medicine Panel Symposium, HFM-135, Symposium on Human Factors of Uninhabited Military Vehicles as Force Multiplier*, Biarritz, France (RTO/NATO, Neuilly sur Seine, 2006), pp. 18–1–18–30
- M.H. Draper, et al., Advanced UMV operator interface, in *Uninhabited Military Vehicles (UMVs): Human Factors Issues in Augmenting the Force* (NATO Report RTO-TR-HFM-078) (NATO RTO, Neuilly-sur-Siene, 2007), Chap. 6
- M. Duquette, The common mission automation services interface, in *AIAA Infotech@Aerospace 2011*, AIAA-2011-1542, 2011
- P.J. Durlach, J.L. Neumann, L.D. Bowens, Evaluation of a touch screen-based operator control interface for training and remote operation of a simulated micro-uninhabited aerial vehicle, in *Human Factors of Remotely Operated Vehicles*, ed. by N. Cooke, H. Pringle, H. Pedersen, O. Connor (Elsevier, New York, 2006), pp. 165–178
- J.W. Eggers, M.H. Draper, Multi-UAV control for tactical reconnaissance and close air support missions: operator perspectives and design challenges, in *Proceedings of the NATO RTO Human Factors and Medicine Panel Symposium HFM-135* held in Biarritz, France, 9–11 Oct 2006 (NATO TRO, Neuilly-sur-Siene, 2006)
- M. Endsley, Theoretical underpinnings of situation awareness: a critical review, in *Situation Awareness Analysis and Measurements*, ed. by M.R. Endsley, D.J., Garland (Lawrence Erlbaum, Mahwah, 2000)
- M. Endsley, *Designing for Situation Awareness: An Approach to User-Centered Design* (CRC, Boca Raton, 2011)
- M. Endsley, D. Kaber, Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics* **42**(3), 462–492 (1999)
- G.L. Feitshans, A.J. Rowe, J.E. Davis, M. Holland, L. Berger, Vigilant spirit control station (VSCS): the face of COUNTER, in *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA-2008-6309, 2008
- L. Fern, R.J. Shively, A comparison of varying levels of automation on the supervisory control of multiple UASs, in *Proceedings of the Association for Unmanned Vehicle Systems International (AUVSI) North America 2009*, Washington DC, 2009
- V. Finomore, D. Popik, R. Dallman, J. Stewart, K. Satterfield, C. Castle, Demonstration of a network-centric communication management suite: multi-modal communication. *Proc. Hum. Factors Ergon. Soc.* **55**(1), 1832–1835 (2011)
- D.J. Gardner-Bonneau, *Human Factors and Voice Interactive Systems* (Kluwer Academic, Boston, 1999)
- S. Green, S. Richardson, V. Slavin, R. Stiles, Spatial dialog for space system autonomy, in *Proceedings of the 2nd ACM/IEEE International Conference on Human-Robot Interaction* (ACM, New York, 2007), pp. 341–348

- M.L. Hanson, E. Roth, C.M. Hopkins, V. Mancuso, Designing human system interfaces for supervising multiple UAV teams, in *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2004-1305, 2004
- B.P. Hunn, Video Imagery's role in network centric, multiple unmanned aerial vehicle (UAV) operations, in *Human Factors of Remotely Operated Vehicles*, ed. by N. Cooke, H. Pringle, H. Pedersen, O. Connor (Elsevier, New York, 2006), pp. 179–192
- R. Johnson, M. Leen, D. Goldberg, *Testing Adaptive Levels of Automation (ALOA) for UAV Supervisory Control*, AFRL-HE-WP-TR-2007-0068, Air Force Research Laboratory, 2007
- D.-M. Kim, D. Kim, J. Kim, N. Kim, J. Suk, Development of near-real-time simulation environment for multiple UAVs, in *International Conference on Control, Automation and Systems*, Seoul, (2007), pp. 817–820
- A. Leganchuk, S. Zhai, W. Buxton, Manual and cognitive benefits of two-handed input: an experimental study. *ACM Trans. Comput. Hum. Interact.* **5**(4), 326–359 (1998)
- M. Lewis, J. Polvichai, K. Sycara, P. Scerri, Scaling-up human control for large UAV teams, in *Human Factors of Remotely Operated Vehicles*, ed. by N. Cooke, H. Pringle, H. Pedersen, O. Connor (Elsevier, New York, 2006), pp. 237–250
- D.F. McAllister, Stereo & 3D display technologies, display technology, in *Encyclopedia of Imaging Science and Technology*, 2 Volume Set, ed. by J.P. Hornak (Wiley, New York, 2002), pp. 1327–1344. ISBN: 978-0-471-33276-3, <http://research.csc.ncsu.edu/stereographics/wiley.pdf>
- C.A. Miller, R. Parasuraman, Designing for flexible interaction between humans and automation: delegation interfaces for supervisory control. *Hum. Factors* **40**(1), 57–75 (2007)
- C. Miller, R. Goldman, H. Funk, P. Wu, B. Pate, A playbook approach to variable autonomy control: application for control of multiple, heterogeneous unmanned air vehicles, in *Proceedings of FORUM 60, the Annual Meeting of the American Helicopter Society* (United States American Helicopter Society, Alexandria, 2004), pp. 2146–2157
- T. Oron-Gilad, T. Porat, J. Silbiger, M. Rottem-Hovev, Decision support tools and layouts for MOMU (multiple operator multiple UAV) environments, in *Proceedings of the International Symposium on Aviation 974 Psychology*, Wright State University, Dayton, Ohio (2011)
- S. Oviatt, Multimodal Interfaces for dynamic interactive maps, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground* (1996), pp. 95–102, <http://www.cse.ori.edu/CHCC/Publications/Papers/sharonPaper/text.html>
- M. Patzek, D. Zimmer, G. Feitshans, Multi-UAV supervisory control interface technology (MUS-CIT) demonstration, in *Supervisory Control of Multiple Uninhabited Systems – Methodologies and Enabling Human-Robot Interface Technologies*, (NATO Report RTO-TR-HFM- 170), Neuilly-sur-Seine (2012)
- C. Plaisant, A. Sears, Touchscreen interfaces for alphanumeric data entry. *Proc. Hum. Factors Soc.* **36**(4), 293–297 (1991)
- T. Porat, T. Oron-Gilad, J. Silbiger, M. Rottem-Hovev, 'Castling Rays' a decision support tool for UAV-switching tasks, in *Computer Human Interaction – CHI* (ACM, New York, 2010), pp. 3589–3594
- T. Porat, T. Oron-Gilad, J. Silbiger, M. Rottem-Hovev, Switch and deliver: display layouts for MOMV (multiple operators multiple video feeds) environments, in *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness & Decision Support – CogSIMA* (IEEE, Piscataway, 2011), pp. 264–267
- R. Rousseau, S. Tremblay, D. Lafond, F. Vachon, R. Breton, Assessing temporal support for dynamic decision making in C2. *Proc. Hum. Factors Ergon. Soc.* **51**(18), 1259–1262 (2007)
- A.J. Rowe, K.K. Liggett, J.E. Davis, Vigilant spirit control station: a research testbed for multi-UAS supervisory control interfaces, in *Proceedings of the International Symposium on Aviation Psychology*, Wright State University, Dayton, Ohio (2009)
- A. Rudnicky, *Speech Interface Guidelines*. School of Computer Science, Carnegie Mellon University, 1996, <http://www.speech.cs.cmu.edu/air/interfaces.html>
- B.M. Scott, V. Conzola, Designing touch screen numeric keypads: effects of finger size, key size, and key spacing. *Proc. Hum. Factors Ergon. Soc.* **41**(1), 360–364 (1997)

- S.D. Scott, S. Mercier, M.L. Cummings, E. Wang, Assisting interruption recovery in supervisory control of multiple UAVs. *Proc. Hum. Factors Ergon. Ann. Meet.* **50**(1), 699–703 (2006)
- B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer-Interaction* (Addison Wesley, Boston, 1987)
- B.D. Simpson, R.S. Bolia, M.H. Draper, Spatial audio display concepts supporting situation awareness for operators of unmanned aerial vehicles, in *Proceedings of the Human Performance, Situation Awareness and Automation Conference* (Lawrence Erlbaum Associates, Mahwah, 2004), pp. 61–65
- H.S., Smallman, M. St. John, Naïve realism: misplaced faith in realistic displays. *Ergon. Des.* **13**(3), 6–13 (2005)
- H.S. Smallman, M. St. John, M.B. Cowen, Limits of display realism: human factors issues in visualizing the common operational picture, in *Visualisation and the Common Operational Picture* (RTO, Neuilly sur-Seine, 2005), pp. 16–1–16–18. Meeting Proceedings RTO-MP-IST-043, Paper 16. Available from: <http://www.rto.nato.int/abstracts.asp>, http://www.vistg.net/IST043_Official/RTO-MP-IST-043/MP-IST-043-16.pdf
- S. Spriggs, L. Warfield, G. Calhoun, H. Ruff, Orientation of a temporal display for multi-unmanned aerial vehicle supervisory control, in *Proceedings of the HCI-Aero 2010 Crew-Ground Integration Symposium*, Cape Canaveral, FL, (2010)
- M. St. John, M.A. King, The four-second supervisor: multi-tasking supervision and its support. *Proc. Hum. Factors Ergon. Soc.* **54**(4), 468–472 (2010)
- M. St. John, M.B. Cowen, H.S. Smallman, H.M. Oonk, The use of 2D and 3D displays for shape-understanding versus relative-position tasks. *Hum. Factors* **43**(1), 79–98 (2001)
- M. St. John, D.I. Manes, H.S. Smallman, B.A. Feher, J.G. Morrison, Heuristic automation for decluttering tactical displays. *Proc. Hum. Factors Ergon. Soc. Ann. Meet.* **48**(3), 416–420 (2004)
- J. Tadema, *Unmanned Aircraft Systems HMI & Automation: Tackling Control, Integrity and Integration Issues* (Shaker, Maastricht, 2011). ISBN 978-90-423-0405-5
- H. van Veen, H., J. van Erp, Tactile information presentation in the cockpit, in *Proceedings of the Haptic Human-Computer Interaction Workshop*, University of Glasgow, 2003, pp. 50–53
- H. Veltman, A. Oving, Augmenting camera images for operators of unmanned aerial vehicles, in *The Role of Humans in Intelligent and Automated Systems* (NATO Report RTO-MP-HFM-088) (NATO RTO, Neuilly-sur-Siene, 2003), Chap. 21, pp. 1–10
- R. Whitaker, G. Thomas-Meyers, *Knowledge Glyphs: Visualization Theory Development to Support C2 Practice*, AFRL-HE-WP-TP-2006-0050, Air Force Research Laboratory, 2006
- C.D. Wickens, Multiple resources and performance prediction. *Theor. Issues Ergon. Sci.* **2**, 159–177 (2002)
- D. Wigdor, D. Wixon, *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture* (Morgan Kaufmann, Burlington, 2011)
- M.S. Wogalter, S.D. Leonard, Attention capture and maintenance, in *Warnings and Risk Communications*, ed. by M.S. Wogalter, D.M. DeJoy, K.R. Laughery (Taylor and Francis, London, 1999), pp. 123–148
- D.D. Woods, Visual momentum: a concept to improve the cognitive coupling of person and computer. *Int. J. Man Mach. Stud.* **21**, 229–244 (1984)
- D.D. Woods, J. Tittle, M. Feil, A. Roesler, Envisioning human-robot coordination in future operations. *IEEE Trans. Syst. Man Cybern. C* **34**(2), 210–218 (2004)

Kurtulus Izzetoglu, Hasan Ayaz, James T. Hing, Patricia A. Shewokis, Scott C. Bunce, Paul Oh, and Banu Onaral

Contents

| | | |
|---------|---|------|
| 102.1 | Introduction | 2476 |
| 102.1.1 | Mental Workload | 2477 |
| 102.1.2 | Expertise Development | 2478 |
| 102.2 | Measures of Cognitive Functioning of UAV Operators | 2479 |
| 102.2.1 | Functional Near-Infrared Spectroscopy | 2479 |
| 102.3 | Case Studies | 2483 |
| 102.3.1 | Cognitive Function Assessment During Simulated UAV Mission Task | 2483 |
| 102.3.2 | Cognitive Workload Assessment in UAV Interface Development | 2486 |
| 102.3.3 | Development of Expertise and Monitoring During a Simulated UAV Mission Task | 2492 |
| 102.4 | Discussion | 2495 |
| | References | 2498 |

K. Izzetoglu (✉) • H. Ayaz • B. Onaral

School of Biomedical Engineering, Science and Health Systems, Drexel University, Philadelphia, PA, USA

e-mail: ki25@drexel.edu; hasan.ayaz@drexel.edu; banu.onaral@drexel.edu

J.T. Hing

Department of Mechanical Engineering and Mechanics, Drexel Autonomous Systems Lab, Drexel University, Philadelphia, PA, USA

e-mail: jtoshiroh@gmail.com

P.A. Shewokis

College of Nursing and Health Professions and School of Biomedical Engineering, Science and Health Systems, Drexel University, Philadelphia, PA, USA

e-mail: shewokis@drexel.edu

S.C. Bunce

Department of Psychiatry, M.S. Hershey Medical Center, Penn State University College of Medicine, Hershey, PA, USA

e-mail: sbunce@hmc.psu.edu

P. Oh

Formerly with Drexel University

Currently: University of Nevada, Las Vegas, NV, USA

e-mail: paul@coe.drexel.edu

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*, DOI 10.1007/978-90-481-9707-1_22,

© Springer Science+Business Media Dordrecht 2015

2475

Abstract

The use of unmanned aerial vehicles (UAVs) is expected to increase exponentially over the next few years. UAV ground operators are required to acquire skills quickly and completely, with a level of expertise that builds the operator's confidence in his/her ability to control the UAV under adverse conditions. As UAVs are held to increasingly higher standards of efficiency and safety, operators are routinely required to perform more informationally dense and cognitively demanding tasks, resulting in increased cognitive workloads during operation. Functional brain monitoring offers the potential to help UAV operators meet these challenges. Recent research has demonstrated the utility of near-infrared-based functional brain imaging systems (fNIRs) for the purpose of monitoring frontal cortical areas that support executive functions (attention, working memory, response monitoring). This technology provides portable, safe, affordable, noninvasive, and minimally intrusive monitoring systems with rapid application times for continuous measures of cortical activity. fNIR technology allows continuous monitoring of operators during training as they develop expertise, as well as the capacity to monitor their cognitive workload under operational conditions while controlling UAVs in critical missions. This chapter discusses the utilization of fNIR in the monitoring of a cognitive workload during UAV operation, and as an objective measure of expertise development, that is, the transition from novice to expert during operator training.

102.1 Introduction

According to the Department of Defense (DoD) Unmanned Aircraft Systems Roadmap 2005–2030 (2005), unmanned aerial vehicles (UAVs) have gained prominence because of their ability to perform dirty, dangerous, and dull missions while personnel remain in a remote location. The recent increase in UAV operations, however, is not without emerging challenges. Critical issues that have been identified include (i) an increase in operation failures due to human error, largely attributed to either inadequate training, or the increased mental workload being placed on ground flight crews and (ii) limited flight certifications and regulations to govern unmanned aircraft, specifically in the NAS (National Airspace System). Based on 194,000 h of operation, the DoD's 2005 roadmap reported that 17 % of system failures for U.S. military UA fleet were due to human/ground control error, whereas 19 % were attributed to flight control error. In addition to this report, Rash et al. (2006) showed that material-related failures were reduced due to advances in technology; however, human-related operation failures had increased (60–80 %). Given the military's experience coupled with recent studies suggesting that the primary factor found to be related to the increase of accident rates for UAVs is human operator failure, integration of the neurophysiologically informed human operator management system with the training procedures and

ground control systems is essential to address these challenges. An objective measure of expertise level to validate adequate training and real-time monitor of mental workload can enable safe operation of UAVs in the theater as well as in the NAS.

102.1.1 Mental Workload

There is no singular definition of mental workload (Cain 2007). There are at least two major theoretical approaches to the construct: (1) mental workload may be defined such that a given task's requirements are viewed as an independent, external variable with which the working participants have to cope more or less efficiently, or (2) mental workload may be defined in terms of an interaction between task requirements and human capabilities or resources (Hancock and Chignell 1986). Both approaches can offer essential and well-founded contributions to different problems. In either of these paradigms, the definition of workload involves the "objective" effects of task difficulty on the participant, and the participant's effort involved in maintaining performance. Workload is an intervening variable between task and environmental demands and the operator's performance, defined by the relationship between task demand and the participants' resource supply, that is, the portion of operator's limited mental capacities actually required to perform a particular task. That is, workload can be defined in terms of some "objective" criteria for task difficulty or in terms of the participant's capacities to perform the identified task. As such, workload can be dissociated from performance. Two people performing the same task can have identical performance, yet one operator may have significant cognitive resources free to allocate to concurrent tasks, whereas the second operator may be just on the brink of performance failure. The difference between the required capacity and the available capacity of an individual can be referred to as the mental or cognitive reserve.

Historically, there have been four basic methodologies for the measurement of mental workload: (1) primary task performance, (2) subjective ratings (e.g., NASA Task Load Index (TLX)), (3) central and peripheral measures of physiological response, and (4) secondary task techniques (Meshkati et al. 1995). Each of these measures plays an important role in the experimental study of workload. However, physiological measures of central nervous system (CNS) functioning hold the most promise for operational monitoring of UAV pilots, as they provide continuous and unobtrusive monitoring of the operator and do not interfere with an operator's work as do secondary task performance or subjective measures of workload (Gevins and Smith 2003; Gevins et al. 1997; Hancock and Desmond 2001; Scerbo et al. 2003; Wilson and Russell 2003). Further, neurophysiological measures of autonomic response (e.g., skin conductance, electrooculogram, electrocardiogram, and respiration) can be unobtrusive and provide an objective measure of workload. However, measures of autonomic responses, that is, the fight or flight system,

assess emotional stress, whereas some of the difference in workload may more appropriately be measured using a measure of cognitive workload, rather than stress. Measures of CNS response can provide information on both.

102.1.2 Expertise Development

A given individual will develop expertise more quickly with increased time on task and/or by utilizing the most appropriate and effective mode of practice for the task to be learned. Native ability may contribute to the speed of an individual's progression and may assist in determining individual differences in enhanced development of expertise in a given course of training. The literature dealing with the effect of practice on the functional anatomy of task performance is extensive and complex. Practice and the development of expertise have been studied across a range of motor, visuomotor, perceptual and cognitive tasks, and from disparate research perspectives. To briefly summarize this literature, four main patterns of practice-related activation change can be distinguished (Kelly and Garavan 2005). Practice can lead to (1) an increase in activation in the brain areas involved in task performance; (2) a decrease in those areas or (3) a functional redistribution of brain activity, in which some initial areas of activation decrease, whereas other initial areas of activation increase; and (4) a functional reorganization of brain activity, that is, the pattern of activation increases and decreases occur in distinct brain areas as well as the initial areas.

The majority of studies examining task practice have found decreases in the extent or intensity of activations, particularly in the attentional and control areas (Kelly and Garavan 2005). This finding is true whether the task is primarily motor (e.g., a golf swing (Milton et al. 2004)) or primarily cognitive in nature (e.g., the Tower of London problem (Beauchamp et al. 2003)). Decreases in activation represent a contraction of the neural representation of the stimulus (Poldrack et al. 1999) or a more precise functional circuit (Garavan et al. 2000). This finding provides an important overlap with the literature on expertise. There is considerable evidence that expertise tends to be associated with overall lower brain activity relative to novices, particularly in the prefrontal areas (e.g., Milton et al. 2004). Both practice and the development of expertise (the latter of which includes individual differences in ability) typically involve decreased activation across attentional and control areas, freeing these neural resources to attend to other incoming stimuli or task demands. As such, measuring activation in these attentional and control areas relative to task performance can provide an index of level of expertise. Functional near-infrared (fNIR) spectroscopy technique has been widely used to monitor frontal cortical areas that support executive functions. Importantly, there is evidence from fNIR technology-based studies showing that differences in workload can be assessed in the dorsolateral and ventrolateral prefrontal cortices during performance of complex tasks and that the level of expertise development can be objectively monitored (Ayaz et al. 2012; Izzetoglu et al. 2004).

102.2 Measures of Cognitive Functioning of UAV Operators

Commonly employed techniques such as electroencephalography (EEG), event-related brain potentials (ERPs), magnetoencephalography (MEG), positron emission tomography (PET), single positron emission computed tomography (SPECT), and functional magnetic resonance imaging (fMRI) have dramatically increased our understanding of a broad range of cognitive and emotional states. Methods that directly measure the summation of neural function, such as MEG, EEG, and ERPs, allow researchers to monitor the direct consequences of brain electromagnetic activity with temporal resolution on the order of milliseconds. However, these technologies also have limited spatial resolution (Strangman et al. 2002) and are susceptible to electromagnetic artifacts in the field. In contrast, indirect methods, such as PET, SPECT, and fMRI, monitor the hemodynamic and metabolic changes associated with neural activity with impressive spatial resolution but are limited in temporal resolution and are associated with neuronal activity through a poorly understood neurovascular coupling function (Strangman et al. 2002). In addition, PET and SPECT do not allow for continuous or repeated measurements because they require the use of radioactive isotopes, which also limits their use in children.

Functional magnetic resonance imaging (fMRI) is currently considered the “gold standard” for measuring functional brain activation. However, fMRI is expensive to operate and requires massive installations and a large infrastructure to operate. In addition, fMRI confines participants to restricted positions, is highly sensitive to motion artifacts, and exposes participants to loud noises. As such, with the exception of EEG, these technologies cannot yet easily be deployed in an operational environment.

102.2.1 Functional Near-Infrared Spectroscopy

Near-infrared spectroscopy (NIRS) has been increasingly applied for the noninvasive measurement of changes in the relative ratios of deoxygenated hemoglobin (deoxy-Hb) and oxygenated hemoglobin (oxy-Hb) during brain activation. In the late 1980s, Delpy designed and tested an NIRS instrument on newborn heads in neonatal intensive care (Cope et al. 1988). In the late 1980s and early 1990s, Dr. Britton Chance and his colleagues, using picosecond long laser pulses, spearheaded the development of time-resolved spectroscopy techniques in an effort to obtain quantitative information about the optical characteristics of the tissue (Patterson et al. 1989). These efforts by Chance, Delpy (Cope et al. 1988), and others (Fishkin and Gratton 1993) expedited the translation of NIRS-based techniques into a neuroimaging modality for various cognitive studies (Hoshi and Tamura 1997; Villringer and Chance 1997).

The combined efforts of these researchers led to the development of three distinct NIRS implementations, namely, time-resolved spectroscopy (TRS), frequency

domain, and continuous wave (CW) spectroscopy (Strangman et al. 2002). In TRS systems, extremely short incident pulses of light are applied to tissue, and the temporal distributions of photons, which carry information about tissue scattering and absorption, are measured. In frequency domain systems, the light source is amplitude modulated to the frequencies in the order of tens to hundreds of megahertz. The amplitude decay and phase shift of the detected signal with respect to the incident are measured to characterize the optical properties of tissue (Boas et al. 2002). In CW systems, light is continuously applied to tissue at a constant amplitude. The CW systems are limited to measuring the amplitude attenuation of the incident light.

CW systems have a number of advantageous properties that have resulted in wide use by researchers interested in brain imaging relative to other near-infrared systems; they are minimally intrusive and portable, affordable, and easy to engineer relative to frequency and time domain systems (Boas et al. 2002; Chance et al. 1998). These CW systems hold enormous potential for research studies and clinical applications that require the quantitative measurements of hemodynamic changes during brain activation under ambulant conditions in natural environments.

102.2.1.1 Underlying Principles of fNIR in Brain Activity Assessment

Understanding the brain energy metabolism and associated neural activity is important for realizing principles of fNIR in assessing brain activity. The brain has small energy reserves and a great majority of the energy used by brain cells are for processes that sustain physiological functioning (Ames 2000). Ames III reviewed the studies on brain energy metabolism as related to function and reported that the oxygen (O_2) consumption of the rabbit vagus nerve increased 3.4-fold when it was stimulated at 10 Hz and O_2 consumption in rabbit sympathetic ganglia increased 40 % with stimulation at 15 Hz. Furthermore, glucose utilization by various brain regions increased several fold in response to physiological stimulation or in response to pharmacological agents that affect physiological activity (Ames 2000). These studies provide clear evidence that large changes occur in brain energy metabolism in response to changes in activity. The levels of compounds involved in energy metabolism and energy metabolites can be outlined as:

- Brain cells consume energy when activated. Oxygen is required to metabolize the glucose. The concentration of oxygen in brain is about $0.1 \mu\text{mol g}^{-1}$ (Hansen and Nedergaard 1988) of which 90 % is in oxy-Hb in brain capillaries. This concentration can support the normal oxygen consumption (about $3.5 \mu\text{mol g}^{-1} \text{min}^{-1}$) for 2 s (Ames 2000). For that reason, an increase in neural activity in the brain is followed by the rise in local cerebral blood flow (CBF) (Buxton et al. 2004).
- Oxygen is transported to neural tissue via oxygenated hemoglobin (oxy-Hb) in the blood.
- The oxygen exchange occurs in the capillary beds.
- As oxy-hemoglobin gives up oxygen, it is transformed into deoxygenated hemoglobin (deoxy-Hb).

- Local cerebral blood flow (CBF) increases much more than the cerebral metabolic rate of oxygen (CMRO₂); therefore, local blood is more oxygenated and less deoxy-Hb present (Buxton et al. 2004).

Based on this brain energy metabolism, methods and imaging modalities, such as fNIR and fMRI (Kwong et al. 1992; Ogawa et al. 1990) for measurements of deoxy-Hb and/or oxy-Hb, are implemented to provide correlates of brain activity through oxygen consumption by neurons.

Because oxy-Hb and deoxy-Hb have characteristic optical properties in the visible and near-infrared light range, the change in concentration of these molecules during increase in brain activation can be measured using optical methods. Most biological tissues are relatively transparent to light in the near-infrared range between 700 and 900 nm, largely because water, a major component of most tissues, absorbs very little energy at these wavelengths (Fig. 102.1). Within this window the spectra of oxy- and deoxy-hemoglobin are distinct enough to allow spectroscopy and measures of separate concentrations of both oxy-Hb and deoxy-Hb molecules (Cope 1991). This spectral band is often referred to as the “optical window” for the noninvasive assessment of brain activation (Jobsis 1977).

Photons introduced at the scalp pass through layers of tissue and are absorbed and scattered mainly by oxy-Hb and deoxy-Hb molecules. Because a predictable quantity of photons follows a “banana-shaped” path and leaves the tissue, these photons can be measured using photodetectors (Gratton et al. 2005). If wavelengths are chosen to maximize the amount of absorption by oxy-Hb and deoxy-Hb, changes in these chromophore concentrations cause alterations in the number of absorbed photons as well as in the number of scattered photons that leave the scalp. These changes in light intensity measured at the surface of the scalp are quantified using a modified Beer–Lambert law (MBLL), which is an empirical description of optical attenuation in a highly scattering medium (Cope 1991; Cope and Delpy 1988).

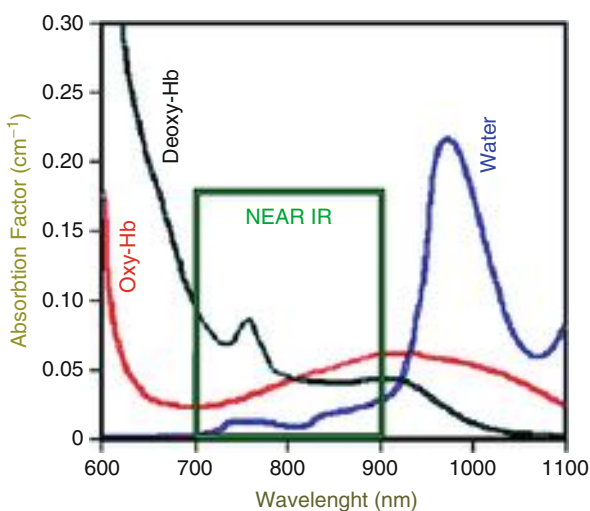


Fig. 102.1 Absorption spectrum in NIR window: spectra of oxy-Hb and deoxy-Hb in the range of 700–900 nm allows spectroscopy methods to assess oxy-Hb and deoxy-Hb concentrations, whereas water absorption becomes substantial above 900 nm, and thus majority of photons are mainly absorbed by water

By measuring absorbance/scattering changes at two (or more) wavelengths, one of which is more sensitive to oxy-Hb, the other to deoxy-Hb, changes in the relative concentration of these chromophores can be calculated. Using these principles, researchers have demonstrated that it is possible to assess hemodynamic changes in response to brain activity through the intact skull in adult human participants (Ayaz et al. 2012; Bunce et al. 2006; Hoshi and Tamura 1997; Izzetoglu et al. 2011; Strangman et al. 2002; Villringer and Chance 1997).

102.2.1.2 CW fNIR System

The CW fNIR system was originally described by Chance et al. (1993). The current generation, flexible headband sensor developed in the Drexel's Optical Brain Imaging laboratory, consists of four LED light sources and ten detectors.

The fNIR system is composed of a computer that runs COBI Studio software (Ayaz et al. 2011) to acquire data from a flexible sensor pad through a hardware box. The sensor pad is a modular design consisting of two parts: a reusable, flexible circuit board that carries the necessary infrared sources and detectors and a disposable, single-use cushioning material that serves to attach the sensor to the participant's forehead (see Fig. 102.2). The flexible circuit provides a reliable integrated wiring solution, as well as consistent and reproducible component spacing and alignment. Because the circuit board and cushioning material are flexible, the components move and adapt to the various contours of the participant's head, thus allowing the sensor elements to maintain an orthogonal orientation to the skin surface, dramatically improving light coupling efficiency and signal strength.



Fig. 102.2 Overview of the CW fNIR system. Flexible sensor housing four LED light sources, ten photodetectors and wires. Computer running COBI Studio collects data through hardware control box

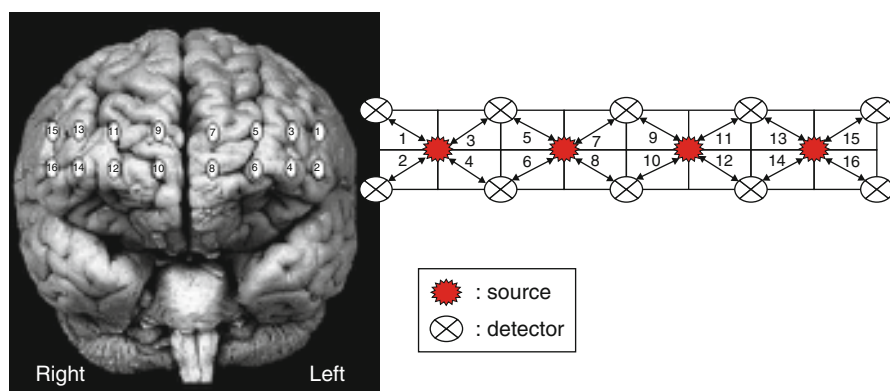


Fig. 102.3 Spatial map of the 16-channel fNIR sensor on the curved brain surface, frontal lobe

The flexible sensor covers the forehead using 16 channels (Fig. 102.3), with a source-detector separation of 2.5 cm. The light sources (manufactured by Epitex Inc.; type no: L4X730/ 4X805/4X850-40Q96-I) contain 3 built-in LEDs having peak wavelengths at 730, 805, and 850 nm with an overall outer diameter 9.2 ± 0.2 mm. The photodetectors (manufactured by Bur Brown; type no: OPT101) are monolithic photodiodes with a single supply trans-impedance amplifier having the size of 0.90×0.90 in. (Bunce et al. 2006). The fNIR sensor, illustrated in Fig. 102.2, reveals information in localizing brain activity, particularly in dorsolateral prefrontal cortex (Izzetoglu et al. 2004, 2005). Figure 102.3 shows a spatial map of the 16-channel fNIR sensor on the curved brain surface, frontal lobe (Ayaz et al. 2006).

The following case studies demonstrate implementation of the CW fNIR system in workload and expertise assessments (Ayaz et al. 2006).

102.3 Case Studies

102.3.1 Cognitive Function Assessment During Simulated UAV Mission Task

The purpose of this case study was to monitor an operator's brain activity during the performance of flight simulator missions. The UAV Predator add-on (First Class Simulations) for Flight Simulator X (Microsoft) provides multiple methods for controlling a realistically functioning and realistically represented UAV model based on the popular Predator UAV. Using the UAV Predator add-on with Flight Simulator, ten missions were designed which incorporated a variety of cognitive workload factors (e.g., crosswinds, cloud cover, visual search tasks, navigation, changing objectives) and that vary in their level of difficulty from easy to extremely challenging. Each mission lasted no longer than 2 h and was designed to require at least 1 h of performance. As an individual successfully completed each mission,



Fig. 102.4 Participant operating the Predator UAV simulator with fNIR sensor on

they were introduced to progressively more difficult missions. Information on events within the simulation, the aircraft's latitude, altitude, and speed, as well as data on the motion of the controllers, was recorded for subsequent assessment along with the fNIR data.

Microsoft Flight Simulator X is a readily available flight simulation program with current support from the company as well as from a large worldwide community of developers, both for the program itself and its software development kit (SDK). It takes full advantage of modern computer hardware to deliver a realistic simulation of every aspect of flight with a variety of aircraft and under a wide range of conditions. Flight Simulator X does not come with a UAV model, so the Predator add-on by First Class Simulations has been used. This add-on provides a realistic aircraft model of the MQ-1 Predator UAV, as well as a realistic interface that can be adapted in many ways. The multiple displays from the typical vertically stacked configuration are adapted to a horizontal arrangement to better suit the multi-monitor configuration (Fig. 102.4).

102.3.1.1 Method

Procedure

Several highly challenging flight scenarios were designed, representing a variety of tasks required of UAV operators (coordinate-based navigation, landing, visual

search/target categorization, etc.) and incorporating a variety of workload factors (e.g., crosswinds, cloud cover, fuel constraints). After a short training session for the purpose of familiarization with the protocol and simulation, participants were asked to fly these scenarios during eight subsequent flight sessions. Each participant performed one flight session per day, each lasting approximately 2 h, for a total of 18 h over 9 days. All participants signed informed consent statements approved by the Human Subjects Institutional Review Board at Drexel University and by the U.S. Army Medical Research and Materiel Command (USAMRMC), Office of Research Protections (ORP), and Human Research Protection Office (HRPO).

The following procedures and tests were designed and implemented during the human experiments:

Session 1: After signing an informed consent, each participant completed the Edinburgh Handedness (Oldfield 1971) questionnaire and a brief questionnaire regarding previous flight and video game experience. Following the fNIR sensor placement on the participant's forehead, each participant performed a 1.5-h introductory flight, during which they are introduced to the UAV simulation, given an overview of the interface, and then they were required to complete a guided tutorial session in order to develop familiarity with the simulation and the very basic proficiencies required (Fig. 102.4).

Sessions 2–9: Each subject attempted one or more of the flight scenarios (Fig. 102.5), with the fNIR device attached with behavioral and neural data obtained during the flight. At the end of each session, a confidence survey and the NASA-TLX are administered to allow subjects to self-rate overall performance.

Data Analysis and Results

Data analysis was performed on the fNIR data while participants performed a visual search/target categorization task (Menda et al. 2010). In this protocol, each participant performed one scenario per session in which they were asked to fly along a coastline and search for a submarine, knowing only that the submarine will be just offshore. When they noticed the submarine, participants were asked to press a button that caused a time-stamped marker to be sent to the fNIR data acquisition software. Five healthy participants with no neurological or psychiatric history

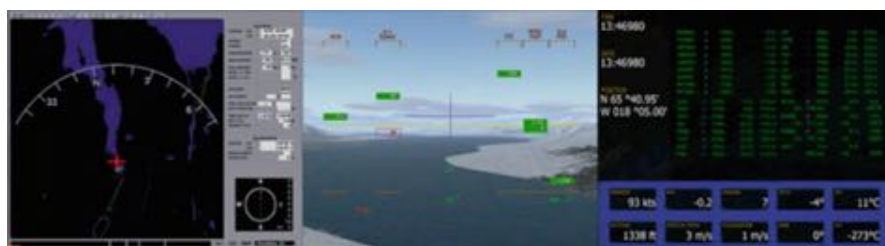


Fig. 102.5 Visual demonstration of UAV-simulated missions

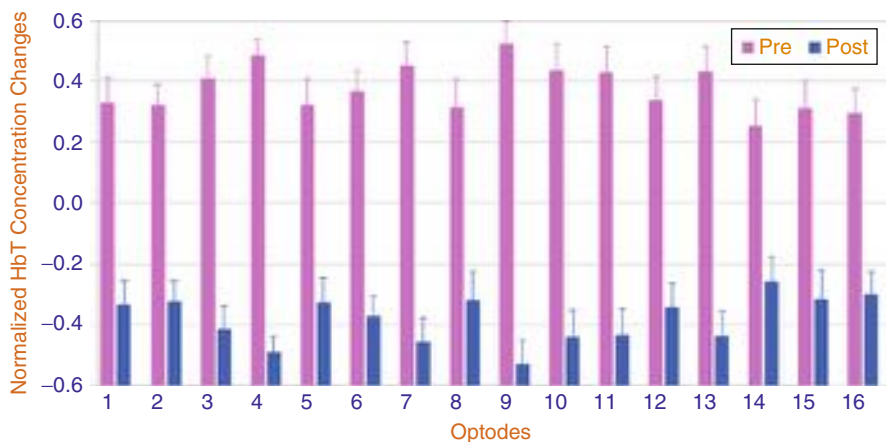


Fig. 102.6 Normalized average HbT in all 16 channels from 5 participants in the target search task. *Pink* represents average HbT in the 100 s before reported target sighting; *blue* represents the same measure within 100 s after reported target sighting

(ages 18–35) voluntarily participated in the study. These data compare average oxygenation changes before and after detection of the submarine.

Time synchronization markers that indicate key events during flight were time-correlated with fNIR data. A linear phase, finite impulse (FIR) low-pass filter with a cutoff frequency of 0.2 Hz was applied to the 16-channel raw fNIR data to eliminate high-frequency noise. For oxygenation calculations, a modified Beer–Lambert law was applied to the data to calculate oxy-hemoglobin and deoxy-hemoglobin concentration changes. fNIR signal data were then averaged over 100 s before and after each participant indicated locating the submarine in each trial. The averages of total hemoglobin (HbT) concentration changes were calculated for pre- and post-blocks and normalized using z-score calculation for each pair independently. Figure 102.6 displays average total hemoglobin concentration changes over all recorded trials from five participants. Separate 2 X 5 (Block, Pre; Post X Trials) repeated measures ANOVAs on both factors were calculated for channels 2 and 4. Channels 2 and 4 were significant; for the main effect of block for channel 2, $F(1,16) = 23.67$, $p < 0.01$, and for channel 4, $F(1,16) = 53.25$, $p < 0.01$ (Menda et al. 2010).

102.3.2 Cognitive Workload Assessment in UAV Interface Development

The aim of this case study is to show that application of fNIR during operator trials with new UAV interfaces may be able to produce an objective assessment of operator cognitive workload, thus providing a vital source of input for evaluating and improving the UAV interface design. In UAV interface development, an increased amount of data displayed on the interface has been shown to improve situational

awareness, but it tends to come at the cost of increased cognitive workload to the operator (Kaber et al. 2000). A balance between performance increase and cognitive workload increase needs to be continually addressed in the development of new UAV interfaces to minimize the risks of operating near the limit of the operator's cognitive capacity as described in Sect. 102.1.1.

An experimental mixed-reality chase view interface was developed for use in direct piloting of UAVs in near-Earth, cluttered environment scenarios. Direct piloting in this case means operation of the UAV through remote control by directly commanding thrust and control surface deflections (aileron, elevator, etc.) through the use of a standard radio 4 channel controller. The chase view interface (shown in Fig. 102.7) integrates both real world "onboard the aircraft" views (onboard camera) with virtual representations of the vehicle and surrounding operating environment. The result is a viewpoint from behind the aircraft such that the extremities of the aircraft are easily seen by the operator and the viewing field is virtually increased. Also, the onboard camera image is stabilized such that the horizon remains level. Aircraft roll is seen directly through visualization of the aircraft motion rather than rolling of the image as experienced from standard onboard camera viewpoints.

A similar approach was originally employed by Drury et al. (2006), supported by their continuing work to define and evaluate situational awareness in unmanned vehicle operations (see Drury and Scott 2008). Their findings indicated that an augmented display improved comprehension of spatial relationships between a UAV and elements of the environment in observational tasks, similar to those performed by payload operators for high-altitude UAVs. Cooper and Goodrich (2008) found

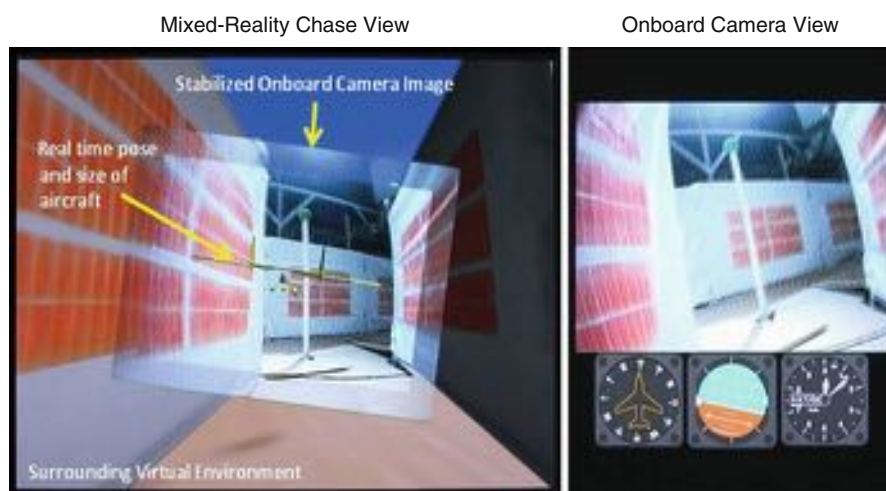


Fig. 102.7 (Left) Screen shot of the mixed reality chase view interface during UAV operation in a cluttered environment. (Right) Onboard camera viewpoint with virtual instruments positioned below the image to relay information about the vehicle state

similar results for search tasks using a mixed-reality system. In their studies, they also found that giving subjects navigation control (i.e., setting waypoints) while conducting the search task helped in flight path recollection, which in turn, improves target localization. These works support the use of a mixed-reality interface in UAV operations for higher altitude missions and waypoint control of the aircraft. Related to this, future applications will also require small UAVs to fly low and in urban/cluttered environments. The present case study focuses on direct piloting of the UAVs in near-Earth environments, where such comprehension of spatial relationships is crucial. We make use of an indoor robotic gantry system, which was developed as a safe means to evaluate factors relevant to UAV operations in near-Earth environments.

This case study addresses the hypotheses: For operating a UAV in a cluttered environment, (1) the chase view will improve an operator's comprehension of the three-dimensional position and orientation of the aircraft with respect to the surrounding environment resulting in more efficient flights (i.e., tighter turns around obstacles), (2) fNIR will detect a higher change in blood oxygenation (i.e., higher cognitive workload) for participants operating via an onboard camera perspective than participants operating via a mixed-reality chase view interface. The narrower viewable field, and rolling of the environment from the onboard camera perspective, results in more cognitive processing for the operator in order to construct an accurate working mental model of the environment and location of the aircraft's position within it.

102.3.2.1 Method

Procedure

A 1:43.5 scale indoor cluttered environment consisting of corridors and obstacles was built inside of a large 3-degree-of-freedom (dof) robotic gantry with a workspace of 18 foot long by 14 foot wide by 6 foot high (shown in Fig. 102.8). The gantry arm was driven through the workspace by the scaled output from the flight simulation software package X-Plane from Laminar Research. A small fixed wing UAV was used as the model for the flight simulator. Henceforth, the gantry arm controlled by the flight simulator is referred to as "the aircraft." Attached to the end of the gantry arm was a 3-dof servo unit that housed a wireless camera and was capable of replicating the yaw, pitch, and roll motions of the model UAV fed to it by the flight simulation software. The wireless camera image was relayed to the operating station and used as the onboard camera perspective and also integrated into the chase view perspective.

Eleven laboratory personnel volunteered as participants during the development of the test methodology. Prior to the start of the study, each subject was given an overview of the experiment and signed a consent form. Six participants flew the aircraft through the environment using only the chase view interface (chase view), and five participants flew the aircraft using only the onboard camera interface (onboard view) (example shown in Fig. 102.8 right). The experiment protocol involved a total of seven sessions per participant, one session per day, and four missions per session. Six sessions involved mission tasks with the fNIR sensor

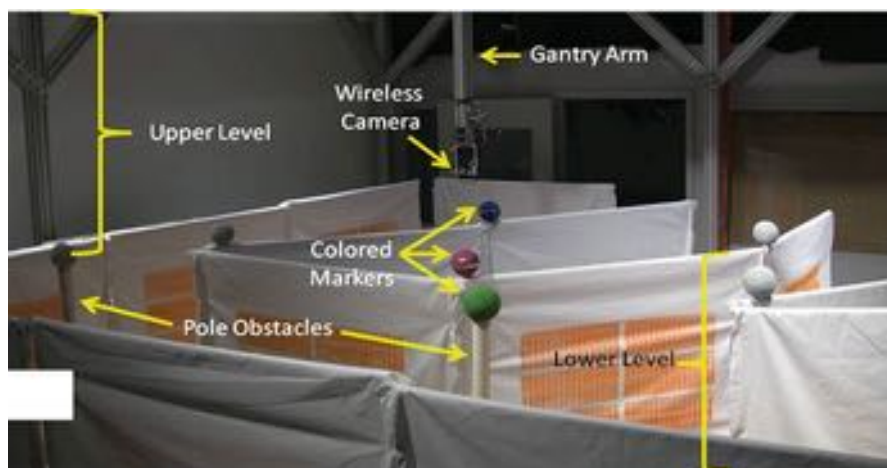


Fig. 102.8 Flight environment inside the gantry built at 1:43.5 scale. In between the corridor walls is considered the “lower level” of the setup and above the corridor walls where the colored markers sit (and above) is considered the “upper level” of the setup

placed on each participant’s forehead, while session one was used to allow subjects to become acquainted with the aircraft controls and flight environment.

During each mission, participants first operated in the lower level of the environment and were tasked with flying through the corridors following a predefined route. The second part of the mission operated in the upper level of the environment and tasked the participants to search for colored markers and fly over the centerline of the markers in a specific order. Participants provided subjective mental effort and performance evaluation using the NASA Task Load Index (TLX) questionnaire (Hart and Staveland 1988). Each session lasts 45 min to an hour.

Data Analysis and Results

Behavioral performance comes from data such as the participant’s accuracy in positioning the aircraft. From the results shown in Fig. 102.9 left, it was found that chase view participants flew tighter paths around the obstacles (closer distance to the obstacles), while onboard view participants took much longer paths ($p < 0.0001$ for both pole obstacles) (Hing et al. 2011). Figure 102.9 right (error means distance from the centerline of the marker) shows that for the second part of the mission (upper level), the chase view participants were better able to position themselves over the centerline of the markers ($p < 0.02$ ($p < 0.05$ set for significance)) (Hing et al. 2011; Hing and Oh 2009). The data supported the hypothesis that a chase view enhances the awareness of the vehicle’s position and extremities, especially when deciding if the aircraft wing tips have safely cleared an obstacle during a turning maneuver. As stated earlier, this also supports the previous findings by Drury and Scott (2008) and Cooper and Goodrich (2008).

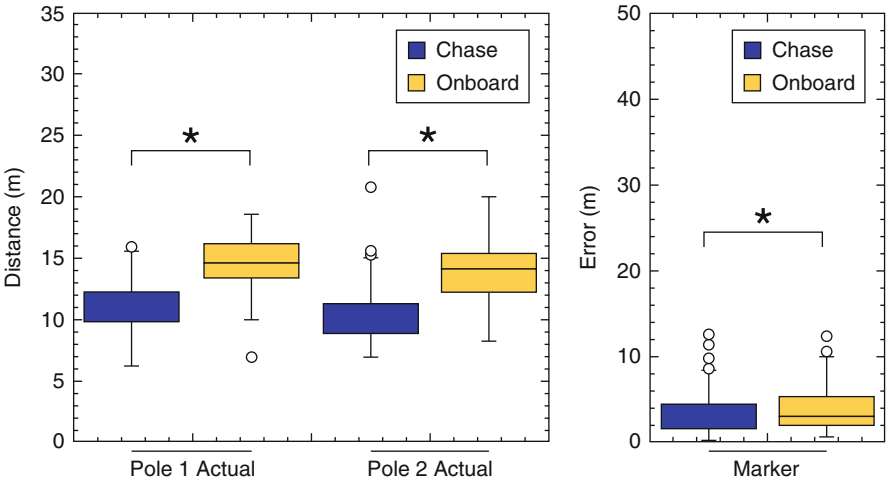


Fig. 102.9 (Left) Horizontal distance (m) of aircraft to the two pole obstacles during turning maneuvers. (Right) Horizontal distance (m) (aka. Error) of the aircraft position from the center line of the marker during upper level flight tasks. Significant distances are highlighted by an asterisk

Table 102.1 summarizes the parameters and results for the analysis of cognitive measures. The NASA-TLX gave a subjective workload assessment for each participant and each session. Chase and onboard views were compared for adjusted weight rating and mental demand using a Mann–Whitney U nonparametric test ($p < 0.05$ for significance) to assess differences between onboard view and chase view groups’ subjective workloads. Monte Carlo permutation tests (10,000 replications) and bootstrapped 95 % confidence intervals of the Monte Carlo tests were calculated. The hemodynamic response features from the fNIR measures (i.e., mean and peak oxy-Hb, deoxy-Hb, oxygenation) were also analyzed. The fNIR measurements were first cleaned of motion artifacts (Ayaz et al. 2011). A linear phase, finite impulse response (FIR) low-pass filter with a cutoff frequency of 0.2Hz was applied to the 16-voxel raw fNIR data for each participant to eliminate high-frequency noise. For oxygenation calculations, a modified Beer–Lambert law was applied to the data to calculate oxy-hemoglobin and deoxy-hemoglobin concentration changes. Analysis was run on all participants. Mann–Whitney U nonparametric tests with view as the grouping variable (with onboard and chase as the levels) was performed across the flights to determine if there were median differences in mean and peak oxygenation for the channels ($\alpha = 0.05$).

The NASA-TLX results are shown in Fig. 102.10. When comparing overall task load score and the mental demand score between chase view and onboard view, no statistically significant differences were found (p values were 0.395 for the overall score and 0.103 for the mental demand score) (Menda et al. 2010). In Table 102.1, the task-weighted workload shows a significant difference for the permutation tests

Table 102.1 Parameters and results of Mann–Whitney U and Monte Carlo permutation tests for key cognitive measures

| Dependent measure | Monte Carlo p value | Mean difference | Exact probabilities $z = (p \text{ value})$ | Bootstrap mean | Bootstrap lower limit 95 % | Bootstrap upper limit 95 % |
|-------------------|---------------------|-----------------|---|----------------|----------------------------|----------------------------|
| Mental demand | 0.004 | 72.359 | 0.854 (0.393) | 72.524 | 17.387 | 122.903 |
| Weighted rating | 0.440 | −2.801 | −1.634 (0.102) | −2.857 | −10.069 | 4.296 |
| Mean oxygenation | 0.020 | −0.386 | −2.001 (0.045) | −0.387 | −0.387 | −0.059 |
| Max oxygenation | 0.029 | −0.386 | −2.367 (0.018) | −0.385 | −0.725 | −0.046 |

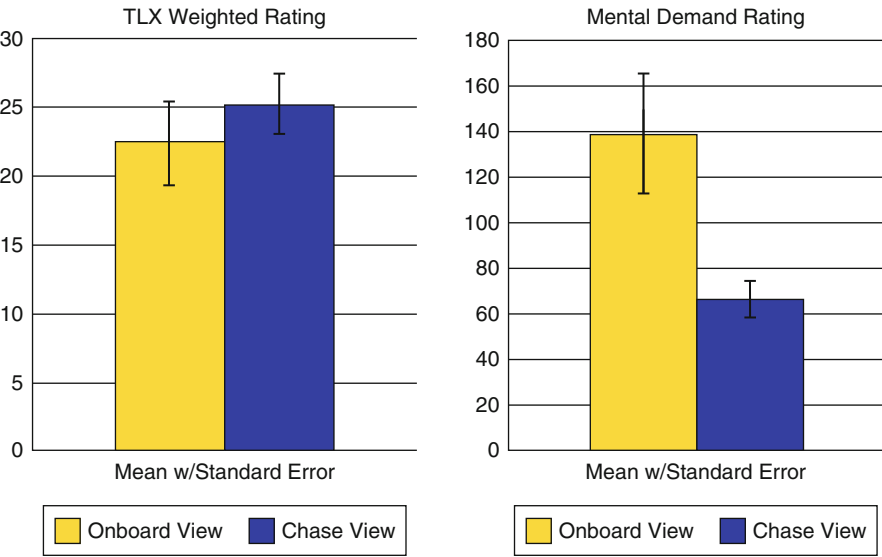


Fig. 102.10 (Left) Task load index weighted rating across sessions. (Right) Mental demand rating across sessions

($p = 0.004$) along with bootstrapped confidence interval width not containing zero. Additional testing with a sufficient sample size, as well as tasks that require more mental stimulation, may permit the NASA-TLX to sufficiently distinguish between the view groups.

While the subjective tests showed no significance, the fNIR analysis showed otherwise. The difference of average oxygenation changes for all chase and onboard view groups were found to be significant ($z = -2.001$, $p < 0.045$) (Menda et al. 2010). Onboard View was found to be significantly higher than chase view. These results are shown in Fig. 102.11 left. The difference of maximum oxygenation changes for chase view and onboard view groups were found to be significant

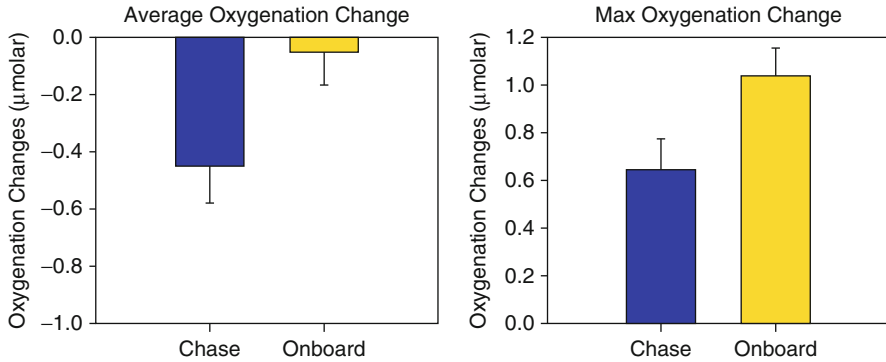


Fig. 102.11 (Left) Average oxygenation changes for chase and onboard view participants. Plot shows Onboard View group's levels are higher. (Right) Maximum oxygenation changes for chase and onboard view groups. Plot shows onboard view group's levels are higher

($z = -2.367, p < 0.018$) (Menda et al. 2010). Figure 102.11 right shows that onboard view had higher maximum oxygenation change when compared with the chase view group. This result suggests that the fNIR approach has potential to be a more sensitive measure of workload than NASA-TLX, since the differences in performance results correlate with the fNIR differences.

These comparisons were performed on channel 4. Activation in the brain area corresponding to channel 4 has been found to be sensitive during completion of standardized cognitive tasks dealing with concentration, attention, and working memory (Ayaz et al. 2012; Izzetoglu et al. 2004). Higher oxygenation in this area is well correlated with higher cognitive workload.

While performance analysis showed that the chase view interface has potential to increase performance, the fNIR data showed its' potential to decrease cognitive workload of an operator directly piloting a UAV in a cluttered environment. The benefit of a lower cognitive workload, as mentioned earlier, is that a pilot would have more cognitive resources available to handle any warnings, system faults, or other unexpected events that might occur during the flight.

102.3.3 Development of Expertise and Monitoring During a Simulated UAV Mission Task

The aim of this case study is to demonstrate that neuroimaging technologies, in this case fNIR, can be implemented to objectively measure expertise development, that is, transition from novice to expert during operator training. This case study can address and test two hypotheses: activation in dorsolateral prefrontal cortical (DLPFC) areas supporting attention, decision making, and working memory

decreases with practice time on task for each individual, and given the same level of task difficulty and the same level of performance, higher levels of activation in DLPFC is associated with less “cognitive reserve,” that is, less capacity to perform at the next level. Similar to aforementioned case study in Sect. 102.3.1, UAV-simulated missions are used for experimental protocols, namely, approach and landing scenarios.

102.3.3.1 Method

Procedure

Participants practiced approach and landing scenarios while piloting a virtual UAV in a flight simulator. The scenarios were designed to expose novice participants to realistic and critical tasks for a UAV ground operator directly piloting an aircraft. The first scenario was a turn-to-approach task, in which the pilot flies through several waypoints on an approach to land at an airfield. The second scenario was a landing task, in which the pilot performs the actual touchdown (Ayaz et al. 2012). In both scenarios, participants were told to fly as smoothly as possible, learn the optimal paths, cope with crosswinds, and operate within certain speed and bank angle constraints (Fig. 102.12).

The experiment protocol involved a total of nine sessions per participant, one session per day. The first session on day 1 was to allow participants to become acquainted with the flight simulator; by the end of this session, they needed to demonstrate basic understanding of flight simulator controls. Study data were collected during the following eight practice sessions. Practice sessions consisted of ten repetitions each of two scenarios (one approach, one landing), a total of 20 flights per session, for a total of 160 trials per participant over the 8 days. Participants provided subjective mental effort and performance evaluation using the NASA Task Load Index (TLX) questionnaire (Hart and Staveland 1988). Each session lasted 2–3 h, with no more than one session per day.

During recruitment and consenting, all participants signed informed consent statements approved by the Human Subjects Institutional Review Board at Drexel University and by the U.S. Army Medical Research and Materiel Command (USAMRMC), Office of Research Protections (ORP), and Human Research Protection Office (HRPO).



Fig. 102.12 UAV simulators on triple panel display running turn-to-approach task screenshots

Data Analysis and Results

UAV fNIR data was preprocessed using the low-pass filters and saturated channel attenuation criteria. fNIR data epochs for the rest and task periods were extracted from the continuous data using time synchronization markers. Blood oxygenation and volume changes for each of the 16 channels were calculated using the modified Beer–Lambert law (MBLL) for task periods with respect to rest periods at beginning of each task (Ayaz et al. 2012, 2011). The main effect for practice level was tested using a one-way repeated measures ANOVA, with participant and practice level designated as fixed effects factors. Geisser–Greenhouse (G-G) correction was used when violations of sphericity occurred in the omnibus tests. Tukey's post hoc tests were used to determine the locus of the main effects with a 0.05 significance criterion. Three practice levels were defined for each participant: the beginner phase included days 2 through 4, intermediate phase included days 5 through 7, and the advanced phase included days 8 and 9 (Ayaz et al. 2012). The Number Cruncher Statistical Software (NCSS) 2007 (www.ncss.com) was used for the statistical tests. The false discovery rate (FDR) approach was used for multiple comparison correction (Benjamini and Hochberg 1995; Singh and Dan 2006).

Ayaz et al. (2012) reported that the NASA-TLX results indicated a significant main effect of practice level (beginner/intermediate/advanced conditions) for mental demand ($F_{2,8} = 17.87$, $p < 0.01$, $\eta_p^2 = 0.817$), effort ($F_{2,8} = 16.32$, $p < 0.01$, $\eta_p^2 = 0.803$), and frustration ($F_{2,8} = 8.60$, $p < 0.01$, $\eta_p^2 = 0.682$). Both mental demand and perceived effort followed a monotonic decrease from beginner to advanced phase (Fig. 102.13).

In the same study reported by Ayaz et al. (2012), behavioral performance for the practice effect was also analyzed. For the approach task, the authors demonstrate that there was a significant monotonic decrease in deviation from the optimal path when comparing practice level (beginner through advanced) for latitude ($F_{2,8} = 18.30$, $p < 0.05$, $\eta_p^2 = 0.793$), longitude ($F_{2,8} = 17.58$, $p < 0.05$, G-G, $\eta_p^2 = 0.815$), bank ($F_{2,8} = 15.80$, $p < 0.05$, G-G, $\eta_p^2 = 0.798$), and heading ($F_{2,8} = 16.41$, $p < 0.05$, $\eta_p^2 = 0.804$) (Fig. 102.14). The analysis for the landing task revealed results similar to the aforementioned approach task, that is, errors for the latitude, longitude, and banking parameters were significant (Ayaz et al. 2012).

For each type of UAV task (approach and landing), the average total hemoglobin (HbT) concentration changes throughout the practice levels (beginner/intermediate/advanced) were analyzed using one-way repeated measures ANOVA and reported by Ayaz et al. (2012). The results indicated that the only significant measurement location was channel #2. Channel #2 is close to AF7 in the International 10–20 System which is located within the left prefrontal cortex (i.e., inferior frontal gyrus). The response was found to be significant for both the approach task ($F_{2,8} = 6.08$, $p < 0.05$, $\eta_p^2 = 0.603$) and the landing task ($F_{2,8} = 7.70$, $p < 0.05$, $\eta_p^2 = 0.658$), see Fig. 102.15. Post hoc analyses confirmed that activation for the beginner phase was significantly higher than the other two phases ($q_{0.05/2,8} = 4.04$, $p < 0.05$) (Ayaz et al. 2012).

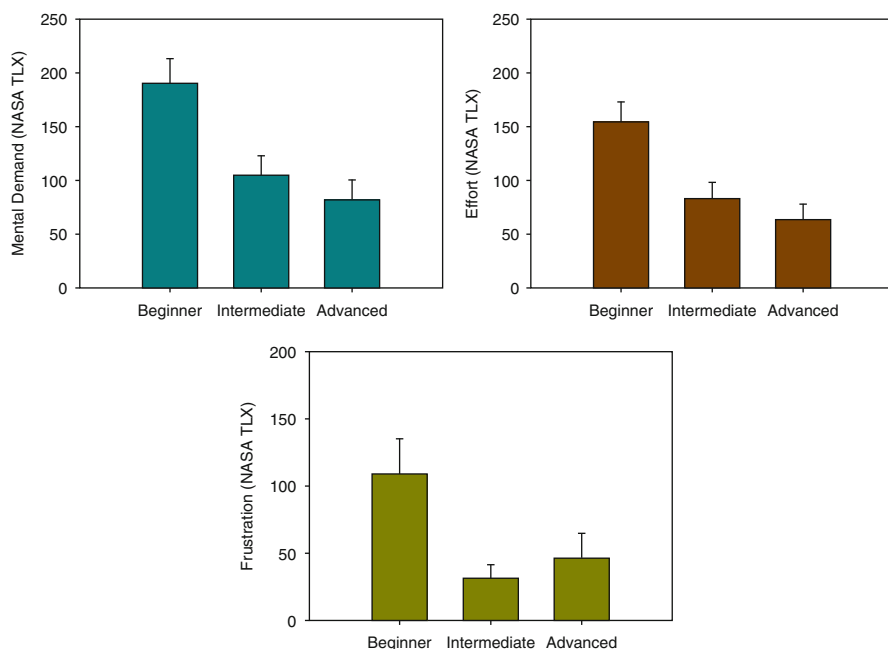


Fig. 102.13 NASA-TLX indices for UAV tasks (5 participants, 8 days per participant): mental demand (*left, top*), effort (*right, top*), and frustration (*bottom*). Error bars are SEM (©Neuroimage)

102.4 Discussion

An accurate measure of mental workload would help improve operational safety and efficacy in many environments that involve multitasking or sustained vigilance. Deployment of portable neuroimaging technologies to operating settings could help assess cognitive states of personnel assigned to perform critical tasks and thus help improve the efficiency and safety of human–machine systems. Functional near infrared spectroscopy (fNIR) is an emerging noninvasive brain imaging technology that relies on optical techniques to detect brain hemodynamics within the prefrontal cortex in response to sensory, motor, or cognitive activation. In this chapter, three case studies have been reviewed where fNIR technology has been used together with simulated UAV tasks to demonstrate efficacious utility of the technique. The results corroborate with previous studies where we have used fNIR to detect cognitive workload of operators in the DARPA Augmented Cognition Warship command program (Izzetoglu et al. 2004) and in FAA Air Traffic Controllers (Ayaz et al. 2012).

In the first case study, cognitive function assessment during simulated UAV mission task, each participant performed one flight session per day for a total of 18 h over 9 days. The result indicates that fNIR is sensitive to high and low cognitive workload states of a UAV operator in a visual search/vigilance task. The second case

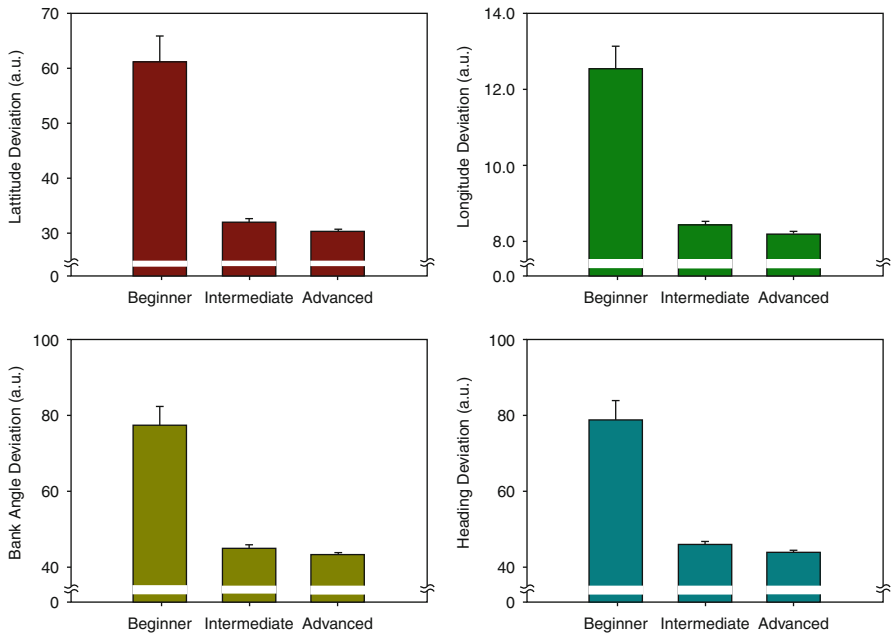


Fig. 102.14 Behavioral performance results for UAV approach task. Deviation from optimal for latitude (*top, left*), longitude (*top, right*), bank angle (*bottom, left*), and heading (*bottom, right*). (5 participants, 10 trials per day, 8 days). Error bars are SEM (©Neuroimage)

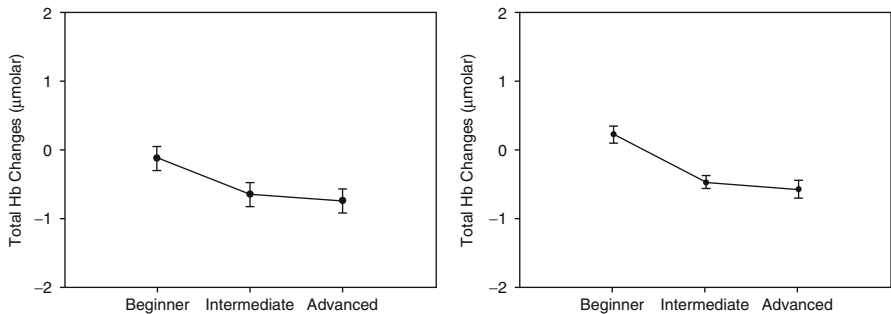


Fig. 102.15 Total hemoglobin concentration changes for the UAV approach (*left*) and landing (*right*) tasks (5 participants, 10 trials per day, 8 days) Error bars are SEM

study, the UAV interface development and workload assessment, deployed the fNIR sensor into the evaluation and analysis of pilot performance. Analysis of the fNIR data found that chase view subjects' average oxygenation levels was significantly lower than onboard view subjects, revealing that subjects using the onboard camera view were utilizing more mental resources to conduct the flights. This result is most likely attributable to the narrower viewable angle and rolling of the environment

in the onboard view. Importantly, this study demonstrates that the fNIR signal is responsive to factors that are relevant to situational awareness and affect UAV operator performance and safety. If these results are validated and elaborated by future studies, fNIR could become a powerful tool in monitoring and improving human factors in UAV operations. In the third case study, expertise development and monitoring, the activation changes in the prefrontal cortex were monitored each day in a 9-day series of simulated UAV tasks as participants practiced “landing” and “approach to turn” tasks. Decreases in the fNIR measures, shown in Fig. 102.15, are significant and a valid hypothesis can be derived from the evidence that expertise tends to be associated with overall lower brain activity relative to novices, particularly in the prefrontal areas (Milton et al. 2004). Both practice and the development of expertise typically involve decreased activation across attentional and control areas, freeing these neural resources to attend to other incoming stimuli or task demands. As such, measuring activation in these attentional and control areas relative to task performance can provide an index for level of expertise, and it illustrates how task-specific practice influences the learning of tasks. The differences in activation of the attentional and control regions of the prefrontal cortex may also indicate neural plasticity as a function of task-specific practice (Kelly and Garavan 2005).

It is particularly important to assess and measure operator training and mental workload in situations where performance failures could result in catastrophic losses (e.g., military command and control, air traffic control). Accurate assessment of mental workload could help in preventing operator error and allow for pertinent intervention by predicting performance decline that can arise from either work overload or understimulation (Meshkati et al. 1995; Parasuraman and Rizzo 2007; Parasuraman and Wilson 2008).

In summary, a field deployable optical brain imaging (fNIR) holds enormous potential for research studies and clinical applications that require the quantitative measurements of hemodynamic changes during brain activation under ambulant conditions in natural environments. As such, fNIR has been already deployed in many field settings for objective measurements of cognitive workload, and further studies have been underway to validate objective assessment of the expertise development which will, among other advantages, allow for dynamic interventions in the training process and helping to assure robust performance under adverse circumstances. fNIR application areas include, but are not limited to, the following: brain computer interfaces for cognitive enhancement, neurological and gaming applications, pediatric solutions, education and training, and cognitive aging.

The case studies described here provide important albeit preliminary information about fNIR measures of DLPFC hemodynamic response and its relationship to mental workload, expertise, and performance, in a complex multitasking environment. Level of expertise does appear to influence the hemodynamic response in dorsolateral/ventrolateral prefrontal cortices, at least for some complex tasks. Since fNIR technology allows the development of mobile, nonintrusive, and miniaturized devices, it has the potential to be deployed in future learning and training

environments to personalize the training regimen and/or to assess the effort of human operators in critical multitasking environments.

Acknowledgments The U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick, MD 21702-5014 is the awarding and administering acquisition office. This investigation was funded under a U.S. Army Medical Research Acquisition Activity, Cooperative Agreement W81XWH-08-2-0573 and in part by W81XWH-09-2-0104. The content of the information herein does not necessarily reflect the position or the policy of the U.S. Government or the U.S. Army, and no official endorsement should be inferred.

The authors would also like to thank Justin Menda and Adrian Curtin for conducting experiments and part of data analyses.

References

- A. Ames III, CNS energy metabolism as related to function. *Brain Res. Rev.* **34**(1–2), 42–68 (2000). doi: 10.1016/s0165-0173(00)00038-2
- H. Ayaz, M. Izzetoglu, S.M. Platek, S. Bunce, K. Izzetoglu, K. Pourrezaei, B. Onaral, Registering fNIR data to brain surface image using MRI templates. *Conference Proceedings – IEEE Engineering in Medicine and Biology Society* 2671–2674 (2006). doi:10.1109/IEMBS.2006.260835
- H. Ayaz, P.A. Shewokis, A. Curtin, M. Izzetoglu, K. Izzetoglu, B. Onaral, Using MazeSuite and functional near infrared spectroscopy to study learning in spatial navigation. *J. Vis. Exp.* (56), e3443 (2011). doi:10.3791/3443
- H. Ayaz, P.A. Shewokis, S. Bunce, K. Izzetoglu, B. Willems, B. Onaral, Optical brain monitoring for operator training and mental workload assessment. *Neuroimage* **59**(1), 36–47 (2012). doi:10.1016/j.neuroimage.2011.06.023
- M. Beauchamp, A. Dagher, J. Aston, J. Doyon, Dynamic functional changes associated with cognitive skill learning of an adapted version of the tower of London task. *Neuroimage* **20**(3), 1649–1660 (2003)
- Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B (Methodological)* **57**(1), 289–300 (1995)
- D. Boas, J. Culver, J. Stott, A. Dunn, Three dimensional Monte Carlo code for photon migration through complex heterogeneous media including the adult human head. *Opt. Exp.* **10**(3), 159–170 (2002)
- S.C. Bunce, M. Izzetoglu, K. Izzetoglu, B. Onaral, K. Pourrezaei, Functional near-infrared spectroscopy: an emerging neuroimaging modality. *IEEE Eng. Med. Biol. Mag.* **25**(4), 54–62 (2006)
- R.B. Buxton, K. Uludag, D.J. Dubowitz, T.T. Liu, Modeling the hemodynamic response to brain activation. *Neuroimage* **23**, S220–S233 (2004)
- B. Cain, A review of the mental workload literature (vol. RTO-TR-HFM-121-Part-II): DTIC document (2007)
- B. Chance, Z. Zhuang, C. UnAh, C. Alter, L. Lipton, Cognition-activated low-frequency modulation of light absorption in human brain. *Proc. Natl. Acad. Sci. USA* **90**(8), 3770–3774 (1993)
- B. Chance, E. Anday, S. Nioka, S. Zhou, L. Hong, K. Worden, R. Thomas, A novel method for fast imaging of brain function, non-invasively, with light. *Opt. Exp.* **2**(10), 411–423 (1998)
- J. Cooper, M.A. Goodrich, Towards combining uav and sensor operator roles in uav-enabled visual search, in *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction*, Amsterdam, The Netherlands, 2008
- M. Cope, The development of a near infrared spectroscopy system and its application for non invasive monitoring of cerebral blood and tissue oxygenation in the newborn infant. Ph.D. thesis, University College London, London, 1991

- M. Cope, D. Delpy, System for long-term measurement of cerebral blood and tissue oxygenation on newborn infants by near infra-red transillumination. *Med. Biol. Eng. Comput.* **26**(3), 289–294 (1988)
- M. Cope, D.T. Delpy, E.O.R. Reynolds, S. Wray, J. Wyatt, P. Van der Zee, Methods of quantitating cerebral near infrared spectroscopy data. *Adv. Exp. Med. Biol.* **222**, 183–189 (1988)
- J.L. Drury, S.D. Scott, Awareness in unmanned aerial vehicle operations. *Int. C2 J.* **1**, 1–10 (2008)
- J.L. Drury, J. Richer, N. Rackliffe, M.A. Goodrich, Comparing situation awareness for two unmanned aerial vehicle human interface approaches, in *Proceedings of the Conference on Safety, Security, and Rescue Robotics (SSRR)*, Gaithersburg, MD, 2006
- J.B. Fishkin, E. Gratton, Propagation of photon-density waves in strongly scattering media containing an absorbing semi-infinite plane bounded by a straight edge. *JOSA A* **10**(1), 127–140 (1993)
- H. Garavan, D. Kelley, A. Rosen, S.M. Rao, E.A. Stein, Practice-related functional activation changes in a working memory task. *Microsc. Res. Tech.* **51**(1), 54–63 (2000)
- A. Gevins, M. Smith, Neurophysiological measures of cognitive workload during human-computer interaction. *Theor. Issue Ergon. Sci.* **4**(1), 113–131 (2003)
- A. Gevins, M. Smith, L. McEvoy, D. Yu, High-resolution EEG mapping of cortical activation related to working memory: effects of task difficulty, type of processing, and practice. *Cereb. Cortex* **7**(4), 374 (1997)
- E. Gratton, V. Toronov, U. Wolf, M. Wolf, A. Webb, Measurement of brain activity by near-infrared light. *J. Biomed. Opt.* **10**(1), 11008 (2005)
- P. Hancock, M.H. Chignell, Toward a theory of mental work load: stress and adaptability in human machine systems, in *Proceedings of the International IEEE Conference on Systems, Man and Cybernetics*, Atlanta, GA, 1986
- P.A. Hancock, P.A. Desmond, *Stress, Workload, and Fatigue* (Lawrence Erlbaum, Mahwah, 2001)
- A.J. Hansen, M. Nedergaard, Brain ion homeostasis in cerebral ischemia. *Mol. Chem. Neuropathol.* **9**(1), 195–209 (1988)
- S. Hart, L. Staveland, Development of NASA-TLX (task load index): results of empirical and theoretical research. *Hum. Ment. Workload* **1**, 139–183 (1988)
- J.T. Hing, P.Y. Oh, Development of an unmanned aerial vehicle piloting system with integrated motion cueing for training and pilot evaluation. *J. Intell. Robot. Syst.* **54**(1), 3–19 (2009)
- J.T. Hing, J. Menda, K. Izzetoglu, P.Y. Oh, An indoor study to evaluate a mixed-reality interface for unmanned aerial vehicle operations in near earth environments. *Int. J. Intell. Control Syst.* **16**(2), 132–141 (2011)
- Y. Hoshi, M. Tamura, Near-infrared optical detection of sequential brain activation in the prefrontal cortex during mental tasks. *Neuroimage* **5**(4 Pt 1), 292–297 (1997)
- K. Izzetoglu, S. Bunce, B. Onaral, K. Pourrezaei, B. Chance, Functional optical brain imaging using near-infrared during cognitive tasks. *Int. J. Hum. Comput. Interact.* **17**(2), 211–227 (2004)
- M. Izzetoglu, K. Izzetoglu, S. Bunce, H. Ayaz, A. Devaraj, B. Onaral, K. Pourrezaei, Functional near-infrared neuroimaging. *IEEE Trans. Neural. Syst. Rehabil. Eng.* **13**(2), 153–159 (2005)
- K. Izzetoglu, H. Ayaz, A.C. Merzagora, M. Izzetoglu, P.A. Shewokis, S. Bunce, B. Onaral, The evolution of field deployable fNIR spectroscopy from bench to clinical settings. *J. Innov. Opt. Health Sci.* **4**(3), 239–250 (2011)
- F.F. Jobsis, Noninvasive, infrared monitoring of cerebral and myocardial oxygen sufficiency and circulatory parameters. *Science* **198**(4323), 1264–1267 (1977)
- D.B. Kaber, E. Onal, M.R. Endsley, Design of automation for telerobots and the effect on performance, operator situation awareness, and subjective workload. *Hum. Factor Ergon. Manuf.* **10**(4), 409–430 (2000)
- A. Kelly, H. Garavan, Human functional neuroimaging of brain changes associated with practice. *Cereb. Cortex* **15**(8), 1089 (2005)
- K.K. Kwong, J.W. Belliveau, D.A. Chesler, I.E. Goldberg, R.M. Weisskoff, B.P. Poncelet, R. Turner, Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation. *Proc. Natl. Acad. Sci.* **89**(12), 5675 (1992)

- J. Menda, J.T. Hing, H. Ayaz, P.A. Shewokis, K. Izzetoglu, B. Onaral, P. Oh, Optical brain imaging to enhance UAV operator training, evaluation, and interface development. *J. Intell. Robot. Syst.* **61**(1–4), 423–443 (2010). doi:10.1007/s10846-010-9507-7
- N. Meshkati, P.A. Hancock, M. Rahimi, S.M. Dawes, Techniques of mental workload assessment, in *Evaluation of Human Work: A Practical Ergonomics Methodology*, 2nd edn., ed. by J. Wilson, E.N. Corlett (Taylor and Francis, London, 1995)
- J.G. Milton, S.S. Small, A. Solodkin, On the road to automatic: dynamic aspects in the development of expertise. *J. Clin. Neurophys.* **21**(3), 134 (2004)
- S. Ogawa, T. Lee, A. Kay, D. Tank, Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proc. Natl. Acad. Sci.* **87**(24), 9868 (1990)
- R.C. Oldfield, The assessment and analysis of handedness: the Edinburgh inventory. *Neuropsychologia* **9**(1), 97–113 (1971)
- R. Parasuraman, M. Rizzo, *Neuroergonomics: The Brain at Work* (Oxford University Press, Oxford, 2007)
- R. Parasuraman, G. Wilson, Putting the brain to work: neuroergonomics past, present, and future. *Hum. Factor* **50**(3), 468 (2008)
- M.S. Patterson, B. Chance, B.C. Wilson, Time resolved reflectance and transmittance for the non-invasive measurement of tissue optical properties. *Appl. Opt.* **28**(12), 2331–2336 (1989)
- R.A. Poldrack, A.D. Wagner, M.W. Prull, J.E. Desmond, G.H. Glover, J.D.E. Gabrieli, Functional specialization for semantic and phonological processing in the left inferior Prefrontal Cortex* 1. *Neuroimage* **10**(1), 15–35 (1999)
- C.E. Rash, P.A. LeDuc, S.D. Manning, Human factors in U.S. military unmanned aerial vehicle accidents, in *Human Factors of Remotely Operated Vehicles*, vol. 7, ed. by N.J. Cooke, H.L. Pringle, H.K. Pedersen, O. Connor. Advances in Human Performance and Cognitive Engineering Research (Emerald Group Publishing Limited, Boston, 2006), pp. 117–131
- M.W. Scerbo, F.G. Freeman, P.J. Mikulka, A brain-based system for adaptive automation. *Theor. Issue Ergon. Sci.* **4**, 1(2), 200–219 (2003)
- A.K. Singh, I. Dan, Exploring the false discovery rate in multichannel NIRS. *Neuroimage* **33**(2), 542–549 (2006)
- G. Strangman, D.A. Boas, J.P. Sutton, Non-invasive neuroimaging using near-infrared light. *Biol. Psychiatry* **52**(7), 679–693 (2002)
- A. Villringer, B. Chance, Non-invasive optical spectroscopy and imaging of human brain function. *Trends Neurosci.* **20**(10), 435–442 (1997)
- G.F. Wilson, C.A. Russell, Real-time assessment of mental workload using psychophysiological measures and artificial neural networks. *Hum. Factor* **45**(4), 635 (2003)

Section XVIII

Human Factors and Training

Fulvia Quagliotti

Kimon P. Valavanis and George J. Vachtsevanos

Human Factors and Training addresses the topic of training of OEMs, military, and other human operators to operate, monitor, and execute a complete mission for all classes of platforms currently available to the military and other sectors.

► **Using Best Practices as a Way Forward for Remotely Piloted Aircraft Operators: Integrated Combat Operations Training-Research Testbed** by Rowe, Conwell, Morris, and Schill centers on work undertaken at the U.S. Air Force Research Laboratory (AFRL) in the Integrated Combat Operations Training-Research Testbed (ICOTT), where training gaps in the RPA community are extensively researched. This is the aftermath of the massive growth and increase of medium altitude, long endurance (MALE) remotely piloted aircraft combat air patrols, which has enforced the need to train RPA operators quickly and effectively. Using a competency-based approach, researchers, engineers, subject matter experts, and warfighters are developing a training research testbed to evaluate curriculum, training methodologies, and best practices in training simulation. The chapter addresses the evolution of training requirements for the RPA community as well as targeted solutions to support RPA training, operational readiness needs, and an exploration into the existing methods and common tools used to understand and evaluate the relative trade-off in training utility associated with alternative training systems. The overarching discussion focuses on methods for providing more rapid and efficient knowledge and skill development for RPA operators.

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

► **From Research to Operations: The PITVANT UAS Training Experience** by Madrugá Matos, Vieira Caetano, Morgado, and Borges de Sousa presents the PITVANT UAS R&D program training experience, including the different crew positions and responsibilities, the several training phases, and the systems engineering methodology behind all developments carried by the joint team of the Portuguese Air Force Academy and Porto University. The training program is described including theoretical classes, written exams, and practical instruction and evaluation. The process of developing, testing, and validating concepts of operations and the correspondent operational procedures and their training with the involvement of end users is reported. The current operational capabilities of the systems are explained, showing that the PITVANT program has achieved a level of maturity that enables testing and validation in near real scenarios. This leads to the next step in the systems spiral development that should include, in the near future, the possibility of technology transfer to the industry.

Contributions also provide useful information highlighting commonality of UAV training requirements and methods required to reduce operator workload.

Using Best Practices as a Way Forward
for Remotely Piloted Aircraft Operators:
Integrated Combat Operations
Training-Research Testbed

Leah J. Rowe, Sharon L. Conwell, Sean A. Morris, and Noah P. Schill

Contents

104.1 Introduction 2506

104.2 A Need for Training 2510

104.3 Training for Coordination 2512

104.4 Training Solutions: The Integrated Combat Operations Training-Research Testbed... 2513

 104.4.1 Joint Terminal Attack Controller Training Rehearsal System 2515

 104.4.2 Air Support Operations Center Training Rehearsal System 2515

 104.4.3 Remotely Piloted Aircraft Training Research 2516

 104.4.4 Training Research 2518

 104.4.5 Competency-Based Approach: Mission Essential CompetenciesSM 2519

 104.4.6 Best Practices: Putting It All Together 2520

104.5 Conclusion 2522

References 2522

Abstract

The growth of medium altitude, long endurance (MALE) remotely piloted aircraft (RPA) combat air patrols increased 660 % between 2004 and 2009 (Gertler 2012). With this massive increase in the U.S. Air Force (USAF) operational requirements, the need to train RPA operators quickly and effectively is in the spotlight. This paradigm change presents an array of warfighter training and readiness issues that require solutions sooner than later. Fortunately,

L.J. Rowe (✉) • S.L. Conwell • N.P. Schill
711th Human Performance Wing Human Effectiveness Directorate, Air Force Research
Laboratory, Wright-Patterson Air Force Base, OH, USA
e-mail: Leah.Rowe@wpafb.af.mil; Leah.Rowe.1@us.af.mil; Sharon.Conwell@us.af.mil
Noah.Schill.2@us.af.mil

S.A. Morris
Air Force Research Laboratory, 711th Human Performance Wing Human Effectiveness
Directorate, OH, USA
e-mail: Sean.Morris.11@us.af.mil

there are USAF research programs that comprehensively address these types of issues. The work highlighted in this chapter is undertaken at the U.S. Air Force Research Laboratory (AFRL) in the Integrated Combat Operations Training-Research Testbed (ICOTT). In the ICOTT among many platforms, the training gaps in the RPA community are researched. Using a competency-based approach, researchers, engineers, subject matter experts, and warfighters are developing a training research testbed to evaluate curriculum, training methodologies, and best practices in training simulation. This chapter addresses the evolution of training requirements for the RPA community as well as addresses targeted solutions to support RPA training, operational readiness needs, and an exploration into the existing methods and common tools used to understand and evaluate the relative trade-off in training utility associated with alternative training systems. The overarching discussion focuses on methods for providing more rapid and efficient knowledge and skill development for RPA operators. The lessons learned from this research represent a first step to ensuring maximum RPA training effectiveness.

104.1 Introduction

The U.S. Department of Defense (DoD) is projected to spend upwards of \$26 billion on procurement, operations, maintenance, research and development, and test and evaluation of RPAs between 2001 and 2013. Comparing this figure to the \$3.9 billion spent from 1988 to 2000 demonstrates the increased attention being given to RPAs (Gertler 2012). Given this exponential growth, it is no surprise that the use of these aircraft has emerged as one of the most demanded capabilities in the joint forces (Department of Defense 2012). RPAs not only provide information to senior operational decision makers but also directly inform joint and coalition forces operating in the field and in congested urban environments. RPAs provide persistence, endurance, efficiency, mission flexibility, surveillance, and attack capability and have proven to be major players across the spectrum of global, joint military, and law enforcement operations. Unmanned technologies provide a capability that supports intelligence, command and control (C2), targeting, and weapons delivery all while providing situational awareness and keeping the warfighters operating the system out of harm's way. Unmanned technologies provide the greatest strategic utility in contested environments that are considered too high-risk for manned systems such as identification and disposal of improvised explosive devices (Department of Defense Unmanned Systems 2011). The most common operations for RPAs are intelligence, surveillance, and reconnaissance (ISR), strike, meaning the ability to deliver ordnance against a ground target, and combat search and rescue of operational personnel.

The oldest unmanned systems are underwater unmanned vehicles; nevertheless, the most utilitarian unmanned systems in the military operational environment are the RPAs. From the size of insects to the size of fighter aircraft, the RPA has forever changed the game of warfare and is now an integral part of the military arsenal. Generally, a military squadron or battalion operates several RPAs. Depending on the

size of the RPA, the usual components include several aircraft, whose size correlates to the size of the control stations. Larger RPAs such as Global Hawk and Reaper have ground stations that are the size of truck trailers, whereas micro-RPAs may be controlled with a handheld controller similar to a flying toy aircraft and a backpack antenna and laptop for data collection. The system also includes datalink antenna or “dish” that communicates by line of sight to the RPA or via satellite relay to control the aircraft and weapons. Finally there is the sensor payload equipment that controls the sensors and imaging equipment, collects the data, and provides presentation of the data to the sensor operator and/or pilot.

For the largest of the RPA, each aircraft has a pilot and a sensor operator. Larger RPAs have take-off and landing gears, while smaller RPAs have launchers to get the aircraft airborne. Typically, larger RPAs are outfitted with sense and avoid technology to help pilots navigate crowded airspace and autonomous systems that tell the RPA to return to base and land if the command and control system loses communication with the ground station. Some RPAs operate manually, that is, the pilot flies the aircraft while the others can be controlled through preprogrammed flight plans or a combination of both. Larger more sophisticated RPAs can be flown in inclement weather and have sensors suites that can see through fog, smoke, and clouds. The larger RPA can be used in remote split operations where the aircraft takes off and lands closer to theater and is flown from a safe base of operations miles if not oceans away.

Engineers have spent billions of dollars on the development of unmanned systems; however, very little research effort or research funding has gone into training the man in the unmanned system. The explosion of unmanned technologies has exceeded the U.S. military’s capacity to deal with training operators in the traditional ways that the U.S. military train flight personnel. This is especially true in the U.S. Air Force. Former predator training commander at Creech, Lt. Col. Lawrence Spinetta, commented that training U.S. Air Force RPA operators is commonly done in theater. For example, in Southern Afghanistan, RPA pilots work 12 h on and 12 h off for 120 days straight (Mulrine 2009).

In 1996, Air Combat Command, a major command of the U.S. Air Force, put forth its concept of operations for RPAs. At the time of publication, there were three platforms in the U.S. Air Force arsenal: Predator – a medium altitude, long endurance platform; Global Hawk – a high altitude, long endurance platform; and a now cancelled platform called DarkStar. Today, the DoD arsenal contains a myriad of unmanned systems. Similar to the development of weapons systems during World War II, the war on terrorism forced the DoD to develop and field unmanned systems at a rapid pace. Rapid technology changes in the private sector outpaced military acquisition, thus, forcing the department to forego standardization and interoperability in order to exploit technology improvements and field much needed systems (Department of Defense 2010). The rapid acquisition and deployment of these systems created logistical and human systems integration (HSI) gaps.

Public Law 109–364, John Warner National Defense Authorization Act for Fiscal Year 2007, Section 941 indicates that the DoD is to identify the mission and mission requirements for which unmanned systems may replace manned systems. Furthermore, there is explicit preference for unmanned systems in acquisition

programs for new systems, including a requirement under any such program for the development of a manned system for a certification that an unmanned system is incapable of meeting program requirements. In addition, Congress required the Federal Aviation Administration (FAA) to develop policy for greater use of unmanned aerial systems in National Air Space for use in homeland defense and border patrol.

This legislation set the stage for DoD to treat unmanned systems with the same regard as manned systems. However, where training is typically addressed in manned weapon programs such as the Joint Strike Fighter (Enewold and Crowley 2006), it has not been addressed well in unmanned programs. Indeed each of the U.S. military services has been left to define who operates unmanned systems and the final qualifications for operation of the systems (House Permanent Select Committee on Intelligence 2012). RPA operators run the gamut from Army-enlisted operators who engage in a nine-week course after basic training to fully qualified pilots manning U.S. Air Force drones. Congress ordered the FAA to create regulations within 2 years to integrate unmanned flight into U.S. airspace by 2015 (Peterson 2012), but the FAA has yet to announce or publish its official regulations on who can operate RPAs in National Air Space and what qualifications are needed to operate RPAs.

In addition, the GAO addressed in its 2011 military training range report the need for airspace to support training of RPA (GAO Military Training Ranges 2011). In April 2012, the House Permanent Select Committee on Intelligence reported several findings and made several recommendations to Congress regarding unmanned system training. In particular, the report faulted DoD for not having a strategic plan for RPA training and not aggressively pursuing one. The Committee recommended relaxing the U.S. Air Force and U.S. Navy standards for RPA operators from rated pilots to non-rated pilots or trained enlisted, collocate RPA training, and substitute simulation training for live training. Also, the committee recommended to upgrade RPA simulators and eliminate interoperability challenges and increase its investment in simulator training.

In July 2009, the Chairman of the Joint Chiefs of Staff provided policy guidance on the minimum training standards for joint RPA. The policy addresses the five critical skill sets required to operate and employ RPA regardless of the operational environment. The requirements for pilot/aircraft operator vary somewhat based on the type of RPA flown, micro to large airframe, and whether or not the RPA carries weapons (CJCSI 3255.01, 2009). Baseline requirements are the completion of ground instruction including weather, aerodynamics, human factors, operational risk management, and flight regulation for the types of airspace (combat and noncombat) in which the RPA operates and flight training. Baseline flight training requirements include a military-approved training program that provides in-flight and/or simulator training. Simulator training must enable flight crews to demonstrate control of a specific RPA throughout its performance parameters under all operating conditions including emergency and system malfunction conditions. The instruction also spells out currency and certification requirements.

Originally, the U.S. Air Force had required that RPA pilots be manned systems-rated pilots. However, with the increased need for RPA pilots, the U.S. Air Force

implemented two key initiatives. The first created an undergraduate RPA training pipeline with the personnel designator 18X for graduates and 1U for sensor operators. In 2012, 146 pilots graduated this program and 327 sensor operators. The second initiative increased the number of training squadrons (Department of Defense 2012). Undergraduate RPA training is composed of three courses. The first course is an initial flight screening in which RPA pilots learn the basics of commanding an aircraft. The second course is with the 558th Flying Training Squadron at Randolph AFB, Texas, where the students engage in the RPA instrument qualification course. Operators then move on to an RPA fundamentals course, gaining information on grounding in combat operations on a simplified MQ-9 Reaper simulator produced by Intific known as the PRIME Desktop Training System (Simmons 2012; Church 2011).

The Army uses enlisted soldiers to fly RPA under the 15W series. Technicians are Warrant Officers in the 150U series, and maintenance personnel are enlisted personnel under the 15E series. The exception to this is the small Raven operator who is required to complete a 10-week flight course. The 15W operator is qualified to fly the aircraft, operate the sensors, and emplace and displace the system. The 150U, fed by the 15W qualification, provides leadership and expertise in an RPA unit (Department of Defense, Under Secretary AT&L 2012). These Warrant Officers interface with higher echelons and provide safety, standardization, and maintenance programs. Currently the manpower requirement for operators of the Army's RPA fleet is 1,456 (note: Raven operators are not included in this number). By 2015, the Army predicts it will need 2,057 operators not including Raven (Department of Defense 2012).

The U.S. Navy unmanned aviation program is not separated from its Littoral Combat Ship aviation detachments. These detachments maintain and operate MH-60R/S Seahawk helicopters and the MQ-8B unmanned helicopter. Operators rotate between the platforms. As Tritons, a Global Hawk variant, replace the P-3 Orion submarine hunter, the Orion operators will transition to Triton. By 2020, the U.S. Navy will have 866 personnel supporting Triton (Simmons 2012). The U.S. Marine Corps flies the RQ-21A Small Tactical Unmanned Aircraft System (STUAS) and the RQ-7B Shadow. The three active duty and one reserve duty unmanned squadrons, or VMUs, are made up of a headquarters and three RQ-7B Shadow and nine RQ-21B STUAS. The RQ-7B unit is comprised of 51 marines and the RQ-21B of 9 marines. An additional 121 marines makeup headquarters and maintenance personnel within each squadron (Simmons 2012).

The need for well-trained RPA operator is growing. According to the Defense Manpower Data Center, the current U.S. military force with personnel system designations showing warfighters attached to unmanned systems was approximately 2,552 which may include payload operators. The DoD reported that manpower requirements for RPA pilots totaled 1,696 and sensor operators 1,194 with a 338 and 245 manning shortfall, respectively, the highest need being MQ-1 Predator operators. The predicted need by 2015 is a total of 2,060 pilots and 1,454 sensor operators (Department of Defense 2012). In an attempt to address the RPA training gap, AFRL's 711th Human Performance Wing is conducting cross-domain training research to evaluate methods and technologies to maximize communication and coordination effectiveness.

104.2 A Need for Training

Effective and efficient training is crucial for safe and successful mission execution across military domains. Budget restrictions necessitate developing innovative training strategies to compensate for fewer live flight opportunities. This limitation is significantly increased for those who operate RPAs for the military because most of the aircraft are deployed in real-world environments. The Joint Unmanned Aircraft System Center of Excellence supported a study in 2008 assessing this need. This study notes that there is a common concern among the joint military commands regarding the limited opportunities to train for force integration. The report explicitly highlights that RPAs do not operate independently and for mission success must integrate with multiple other players (Department of Defense 2012).

The Global Hawk community has struggled with determining training requirements. Based on discussions between Global Hawk operators and traditional pilots, Cantwell (2009) concluded that there are distinct differences in skill sets used by Global Hawk operators and traditional pilots. Lt Col Christopher Jella, former commander of the U.S. Air Force 18th Reconnaissance Squadron, noted that the long duration of mission and high altitudes of Global Hawk prevent operators from developing skills typically associated with pilots such as interacting with air traffic controllers, transiting controlled airspace, and taking off and landing (Cantwell 2009). Based on the rotation of operators and the duration of flights, Global Hawk operators typically interface with air traffic controllers about once every 2 months. Global Hawk's high level of automation, such as auto-takeoff and auto-land, only requires pilots to monitor the aircraft's autonomous systems, not actually fly it. In fact, according to Jella, the longer an operator is assigned to Global Hawk, the more their airmanship skills fade.

Compared to pilots of traditional aircraft, Global Hawks operators sitting halfway around the world from the aircraft must deal with a host of unusual situational problems when Global Hawk systems go awry. Situational assessment is more complex since pilots cannot feel how the aircraft is handling. Operators know only the information transmitted into the GCS instruments. For example, air turbulence can easily be mistaken for a flight control anomaly. Long mission durations complicate the matter because no one pilot is flying the whole mission; therefore, there is little continuity in the mission scenario. While pilots do debrief during control transition, it is impossible to completely capture the aircraft's performance across an entire mission. Brig Gen H. D. Polumbo, former commander of the 9th Reconnaissance Wing, noted that dealing with a crippled or misperforming Global Hawk requires a highly skilled and experienced airman to recover the aircraft safely. However, what Global Hawk operators do have available is a host of ready expertise that can be called into the cockpit (Cantwell 2009). However, this is a double-edged sword, as the services need to develop the skills of their operators, not have operators rely on the skills of seniors which could lead to a long-term degradation of skills across the community as less skilled operators move up the ranks and replace their seniors.

The tremendous demand for Predator operations, combined with training regimes and simulators not being addressed properly during the acquisition phase of Predator, placed the U.S. Air Force and U.S. Army into an on-the-job training situation in order to maintain operation tempo (Wisher et al. 2011). Cantwell (2009) reported that Predator and Reaper squadrons do not carve time out of their flying-hour program to meet training requirements. In addition, RPA operations and manned aircraft operations are different enough in terms of flight operation that RPA operators question the relevance of many of the flight currencies typically maintained by pilots. Elimination of takeoff and landing currencies, for instance, has caused significant changes to the Predator training syllabus. Less than 50 % of Predator/Reaper pilot skills do not rely on stick and rudder tactics but rather on airmanship and cognitive judgment skills regarding the mission; however, stick and rudder tactics are missing tactile and vestibular cues normally felt by pilots in manned aircraft.

Cantwell (2009) notes that within the USAF, operational tempo prompted Predator/Reaper squadrons to eliminate continuation-training sorties, but the performance gap due to lack of training is an unknown. Predator/Reaper operators fly demanding operational missions such as close air support, air interdiction, support of Special Forces, and killer scout missions in addition to typical ISR missions. Reaper operators employ a host of weapons, including Hellfire missiles and laser-guided bombs, in a multitude of possible scenarios. By contrast, attack and fighter pilots maintain carefully regulated currencies and training requirements for such engagement scenarios (Cantwell 2009).

In addition to operations tempo, restriction to national air space in the U.S. has forced the military service schools to train RPA operators through simulation. In 2012, the services operated more than 250 classrooms of PC-based RPA military training. As the war-time demand for RPA operators has increased, the U.S. Air Force responded in 2010 with its first graduating class of non-aviator Predator pilots and ramped up the numbers quickly (Jean 2010). However, with the war against terror fading, the USAF Chief of Staff will bring a set of MQs back to the U.S. to provide for organic training, with the majority of MQ moving to the Pacific and to South America where commanders need ISR for such missions as drug smuggling. As MQs move back into training operation, the USAF continues to work with the FAA to find suitable options for RPA training in the USA. Dozens of locales are bidding to be one of six spots around the country where drones will be able to fly in the same airspace as piloted aircraft. According the General Schwartz, the USAF's preference is to bring unmanned training out of restricted airspace and into the national airspace in order to train operators close to home (Beidel 2012).

Internationally, the United Kingdom and Italy have developed Reaper/Predator training programs. The Royal Air Force (RAF) embarked on a similar course as the U.S. Air Force by first converting manned system-rated pilots to RPA pilots. In 2009, the RAF implemented the Daedalus Project which included a selection battery and training program for RPA operators and sensor operators. The initial Reaper operator group was sent to Creech AFB for final training where USAF officials conclude that the operators were Predator ready, but not ready for

Reaper (Hoyle 2011). However, by 2012, the RAF concluded that the Daedalus Project had not achieved its goals and the RAF will continue the use of manned system-rated pilots to operate RPA. The UK will deploy the Israeli Hermes 450 in late 2012 and also flies Heron as do Canada and Australia. The Hermes 450 RPA has human factors and situation awareness issues that have to be addressed through training, but the RPA is being replaced with Israeli made Watchkeeper (“Inzpire Develops Human” 2012).

Provided the joint and coalition perspective provided in the previous paragraphs, it is evident that operator training should be able to encompass large teams and teams of teams. It is common for warfighters to work with joint and coalition partners. However, the introduction of distributed players in the real-time war is new and unique to the environment where our warfighters operate. In the following section, the challenges associated with collaborative training are discussed.

104.3 Training for Coordination

“Things began to look up during one shift when we received intelligence that might lead us to the man himself – Osama bin Laden. The source pointed to a tiny village along the Afghanistan-Pakistan border. Bin laden, it seemed, preferred to dwell in places where few other people would go. I headed that way, throttle open, along with a couple of A-10’s, a B-1B bomber, and a dozen marine helicopters” (Martin and Sasser 2010, p. 55). In this excerpt from “Predator the Remote-Control Air War Over Iraq and Afghanistan: A Pilots Story,” Predator pilot Matt Martin discusses one of his many missions while fighting in a war thousands of miles away from where he was. In another excerpt, he identifies the need to coordinate with a joint terminal attack controller, “I was on the radio and in the chat rooms with the on-ground forward air controller, also known as a joint terminal attack controller (JTAC), feeding him live video so he could see what I was seeing. He was ready to grant clearance as soon as I had the opportunity to engage” (Martin and Sasser 2010, p. 51). In these two excerpts, Martin discusses a multitude of ways that the Predator crew is required to coordinate with other team members who are not colocated with them.

Gorman and colleagues (2007) indicate that in cognitively demanding team environments, like RPA control, that teams coordinate to process information at the team level and that effective team coordination is a critical facet of highly effective team performance. In a study assessing a three-person team in an RPA like synthetic task environment, researchers measured team coordination and effectiveness. The researchers reported that perturbations to team coordination caused by mixing team members across team and introducing long retention intervals provide for more adaptive teams (Gorman et al. 2006), where adaptation was defined as successfully handling novel or unforeseen situational roadblocks to effective team performance (Gorman et al. 2006). Moreover, a team team provided with more obstacles (e.g., mixing team members after a long inter-mission retention interval) had an increased ability to adapt to novel situations via coordination flexibility.

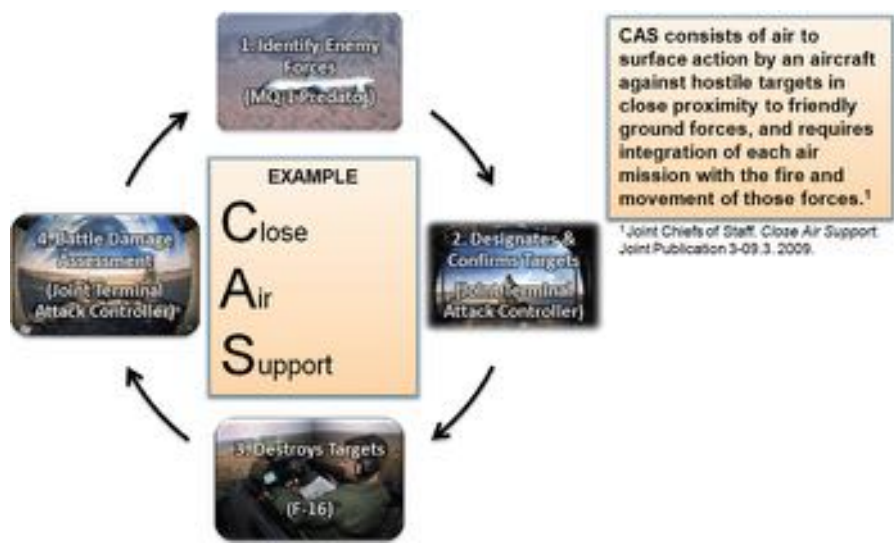


Fig. 104.1 Example of the coordination required for a CAS mission

To demonstrate the need for cross coordination, consider a close air support (CAS) mission. Common in today’s battlefields, CAS consists of air to surface action by an aircraft against hostile targets in close proximity to friendly ground forces and requires integration of each air mission with the fire and movement of those forces (Joint Chiefs of Staff 2009). Accordingly, a CAS mission requires a great deal of planning, coordination, and training for successful and safe execution. Failing to execute a CAS mission could lead to mission failure and even fratricide. Because of their long loiter times and easy deployment, RPAs are often at the forefront of these missions. Therefore, training RPA operators and RPA support personnel to properly coordinate and communicate is essential for mission and military success. Figure 104.1 illustrates an example of the multitude of players that could be involved during a CAS scenario.

104.4 Training Solutions: The Integrated Combat Operations Training-Research Testbed

AFRL is actively looking for training solutions that allow a team of warfighters to train under warlike conditions. The Integrated Combat Operations Training-Research Testbed (ICOTT) combines a family of complimentary trainers to assess training strategies for RPA crews, joint terminal attack controllers (JTAC), and warfighters within an Air Support Operations Center (ASOC). This program strives to provide alternative training strategies and methods for enhancing RPA training by offering solutions that match the level of fidelity required for specific types of training;

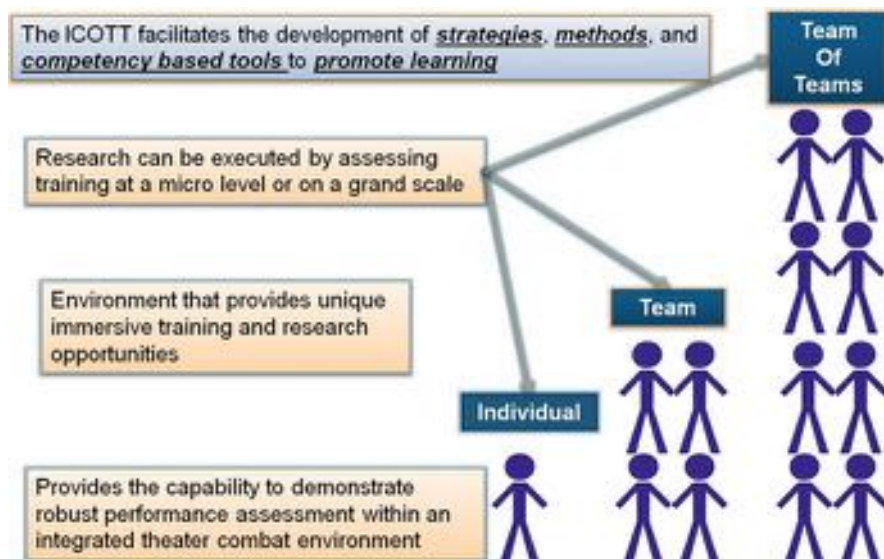


Fig. 104.2 Individual to robust team of team training

thus using training resources effectively and appropriately. In the ICOTT, the research leverages both commercial and government off-the-shelf solutions to build cost-effective, training gap, innovative solutions that allow for the research and development of unique strategies for continuous learning.

In the ICOTT, research can be executed by assessing training at a micro level (e.g., assessing an individual warfighter) or on a grand scale (e.g., assessing a team of teams) as shown in Fig. 104.2. Furthermore, the ICOTT facilitates the development of strategies, methods, and competency-based tools to promote learning within and across the RPA, JTAC, and ASOC operational domains. Findings from the research executed in the ICOTT provide research-driven recommendations for training and assessment in the U.S. Air Force, joint and coalition forces, and in the commercial RPA community. The ICOTT is designed to facilitate a team of teams training experience with interaction in an immersive environment for training research. This testbed provides an opportunity to develop and validate quantitative metrics for measuring these warfighter teams. Furthermore, the ICOTT provides the capability to demonstrate robust performance assessment within an integrated theater combat environment.

The ICOTT is unique because it provides innovative, high-fidelity training that warfighters need. However, at its core, the ICOTT is a research testbed built to assess continuous learning. The research is driven by several research questions, including:

- What are the supporting competencies, knowledge, and skills for effective team and team of teams training?
- What are the training needs, gaps, and opportunities in a networked training environment for JTAC, ASOC, and RPA teams?

- What is the most effective training structure to effectively train a team of teams in the ICOTT that will provide enhanced training capabilities to the warfighter?
- What are the methods and strategies for linking operationally relevant training devices?

Each system in the ICOTT has a unique purpose and function and is able to support Distributed Mission Operations (DMO) (multiplayer networked environments) and in the near future will aid in Live, Virtual, and Constructive (LVC) training research (allowing simulated, live, and computer-generated assets to train together). Each training research system in the ICOTT is illustrated and discussed in turn.

104.4.1 Joint Terminal Attack Controller Training Rehearsal System

The JTAC Training Rehearsal System (TRS) provides an immersive environment to investigate and evaluate training requirements, methods, and future training needs for the JTAC community and is shown in Fig. 104.3. This trainer was developed at AFRL using commercial off-the-shelf (COTS) products and is the only trainer to provide this level of fidelity to JTAC warfighters. This is accomplished by employing physics-based threat generation capabilities with semi-realistic weapon effects. The JTAC-TRS consists of a 5 m 220° field of view dome and includes high-fidelity form fit replicas of the devices that a JTAC warfighter needs to do their job.

The JTAC-TRS is fully equipped with an instructor operator station (IOS) to control the JTAC devices, manage scenario generation, and execute the command and control functions of the system. Training effectiveness is also enhanced through an integrated brief/debrief capability with audio and video playback ability. Remotely Operated Video Enhanced Receiver (ROVER) feed capability integrates full motion video from compatible RPA training systems. JTAC-TRS is Distributed Interactive Simulation (DIS) compatible and is capable of both DMO and LVC operations. These capabilities provide the capability for the JTAC-TRS to work with both live and simulated assets to complete realistic team training events.

104.4.2 Air Support Operations Center Training Rehearsal System

The ASOC Training Rehearsal System (TRS) is a proof-of-concept training research simulator providing theater-level training for ASOC warfighters and is shown in Fig. 104.4. There is little formal training and no existing simulators for warfighters assigned to the ASOC. ASOC TRS aims to fill this gap by incorporating operational command and control systems, computer-generated forces, and replicating the positions and functions of the ASOC through intelligent agent technology. By using intelligent agents as active role players, JTAGSS provides additional training capability and flexibility for this dynamic training environment. The intelligent agent configuration permits realistic computer-generated entities to replace individual ASOC team members with an interactive intelligent agent. Communication between each ASOC position and other command components will be improved by further

Fig. 104.3 Joint terminal attack controller training rehearsal system



developing voice and digital recognition technology. A completed and validated proof of concept will be transitioned to the warfighter to provide the first training system of its kind for ASOC warfighters. Using DIS protocol, JTAGSS is able to network and participate in DMO exercises with other simulators to provide team of teams training, to include the JTAC and RPA simulators in the ICOTT.

104.4.3 Remotely Piloted Aircraft Training Research

The ICOTT RPA training and assessment program provides warfighters and decision makers with research-driven recommendations for training and assessment in the operational and commercial RPA community. The ICOTT is the home of three RPA training simulators: the Reaper Mission Training Device (R-MTD),



Fig. 104.4 Air Support Operations Center Training Rehearsal System

Predator/Reaper Mission Training Device (P/R-MTD), and the Predator Research Integrated Networked Combat Environment (PRINCE).

Both the R-MTD and P/R-MTD are commercially purchased systems and are shown in Figs. 104.5 and 104.6. The R-MTD is presently used to train Air National Guard MQ-9 Reaper pilots and sensor operators. While in the ICOTT, research is being completed to determine what knowledge, skills, and experiences can best be trained using this system. Additionally the researchers at AFRL are developing a competency-based training syllabus that compliments the training that the systems are capable of providing.

PRINCE, shown in Fig. 104.7, is an AFRL built simulator in the final stages of development. The primary purpose of the system is to provide a high-fidelity environment to support ongoing training effectiveness research for standalone RPA crews as well as RPA crews collaborating with other warfighter teams such as JTACs and ASOCs. This game-based system brings the COTS technology to a new level of fidelity by using game-based technology to provide both the simulation backbone and visual system. Gaming software is capable of rendering extremely realistic environments, high-fidelity entities with articulated parts, material-specific infrared radiance data, and very detailed urban environments like the ones found in modern commercial video games.

On top of the game-based technology at the system's core, PRINCE incorporates a highly realistic Ground Control Station (GCS). Just like the live GCS, PRINCE has fully functional heads down display touch screens, a complete tracker display, and peripheral components used operationally, such as Military Internet Relay Chat, mapping programs, and radio connectivity. PRINCE is also DIS capable and is able to interconnect to other DIS systems for distributed training.

Fig. 104.5 Reaper Mission Training Device



Fig. 104.6 Predator/Reaper Mission Training Device



104.4.4 Training Research

The simulators described are a means to an end. Each of these systems is in place to execute training research. Some systems in the ICOTT are commercial and others are home grown; nonetheless, each simulator has a purpose to provide data related to the training effectiveness and continuous learning of warfighters. The ICOTT has capitalized on a training research model introduced by Schreiber and Bennett (2006) during the *Distributed Mission Operations Within-Simulator Training Effectiveness Baseline Study*. During this study, 76 teams of F-16 pilots and air battle managers participated so that researchers could assess the use of Distributed Mission Operations (DMO) to enable warfighter training on higher-order individual and team-oriented skills. The dataset used in this research represented the largest

Fig. 104.7 Predator
Research Integrated
Networked Combat
Environment



DMO effectiveness study known to exist today. During this study, several measures were collected to include objective data from the simulators, multiple participant surveys, subject matter expert ratings of performance, and knowledge structure tests. This study found that DMO training lead to significant improvements in warfighter performance between the pre-training and post-training assessments. The present efforts in the ICOTT are striving to use the findings in this study as best practices to develop a persistent training research program.

104.4.5 Competency-Based Approach: Mission Essential CompetenciesSM

The training research that is pursued in the ICOTT uses a building block or competency-based approach. Training curricula, scenarios, and systems are mapped to the knowledge, skills, and experiences that warfighters are able to provide to trainees. This work is accomplished using the Mission Essential Competency job analysis methodology. This methodology was developed by the AFRL and Air Combat Command and among many things is able to determine training requirements for personnel.

Mission Essential Competencies (MEC) are defined as “Higher-order individual, team, and inter-team competencies that a fully prepared pilot, crew or flight requires for successful mission completion under adverse conditions and in a non-permissive environment” (Colegrove and Alliger 2002, pp. 12). The MEC process results in several outcomes to include MEC statements, supporting g competencies, statements of knowledge and skills, and experiences and is illustrated in Fig. 104.8 (Alliger et al. 2012). The MEC process provides a structured and validated method

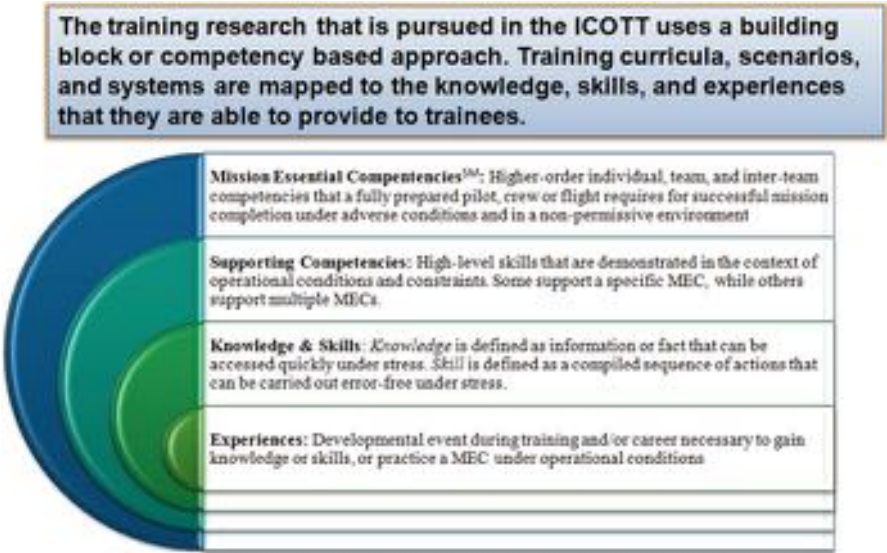


Fig. 104.8 Mission Essential Competency structure

to discover training gaps. Once training gaps are defined, training practitioners are able to engineer and design research focused on filling these gaps.

The MEC-based approach affords researchers the ability to assess training curriculum and programs to rapidly transition these new methods to the warfighters as either permanent or temporary training gap fillers. The ICOTT leverages the MECs to build research programs that are focused on competencies as defined by the warfighter. Using this approach provides a unique research opportunity. Given the present manpower demands and task saturation of warfighters, it is difficult for warfighters to participate in research efforts; however, researchers have found that this limitation can be overcome by providing a unique and valuable training opportunity not available elsewhere to warfighters.

104.4.6 Best Practices: Putting It All Together

The ICOTT uses a circular approach where objectives set the pace of the research, conditions are used to create the materials, and during the assessment stage, lessons learned are fed back to set new objectives. This approach is illustrated in Fig. 104.9.

The ICOTT objectives are set using MEC-based approaches to demonstrate the context of an actual mission using high-fidelity simulation resources. This level of mission reality provides a greater opportunity to have warfighters versus generic research subjects participate in research. Research efforts focus on scenarios that are applicable during wartime conditions, therefore, providing a direct transfer of training scenario for the warfighters.



Fig. 104.9 Best Practices and Processes in the ICOTT

To develop conditions, the ICOTT team is creating a scenario-based training curriculum that provides a building block approach to training similar to the approach taken in the Schreiber and Bennett (2006) study that was previously discussed. This allows for training research exercises that are both realistic and include a high fidelity after action review capability. The ICOTT provides for the capability to execute joint and coalition training and research exercises and allows researchers to assess common architectures to train more efficiently using live, virtual, and constructive systems. The family of trainers in the ICOTT provides the opportunity to develop and validate scenarios and syllabi that can be used across training venues and platforms.

Finally the ICOTT uses assessment strategies to close the loop for training research. To provide the most robust assessment available, the ICOTT warehouses data that in turn allows for the assessment of readiness and proficiency. This provides the ability to determine what has been learned and at what level it has been learned, the frequency in which learning needs to take place, and finally provides the ability to assess the readiness and proficiency of the trainee.

The ICOTT has had success in leveraging applied research methodologies that have been previously validated and vetted. One of the most recent successes include the development of MECs for Predator, Reaper, Global Hawk, JTAC, ASOC, and commercial medium altitude, long endurance RPAs. Presently, researchers are working to develop a MQ-1 and JTAC training research curriculum that will allow warfighter to participate in collaborative research while being provided with unique training that cannot be attained elsewhere. The team is also holding and is exploring additional agreements and partnerships with many members from the academic, commercial, federal, military, and coalition partners.

104.5 Conclusion

As illustrated in this chapter, the requirements for effective RPA training are real. There is a clear requirement for RPA pilots to work in dynamic environments while in realistic distributed locations. One line of research attempting to address some of the team and coordination training gaps for the RPA is the research that takes place in the ICOTT. The ICOTT provides a unique and innovative training-research testbed that enables the comprehensive team and team of teams training research for RPA, JTAC, and ASOC warfighters. Using a MEC-based approach to training and practices previously proven, the ICOTT utilizes scenario-based training across the range of warfighters that are researched in the ICOTT. Successes in the ICOTT have led to achieving accreditation for the JTAC TRS as well as development of MECs for all ICOTT systems and their warfighter users. The ICOTT team is dedicated to a training research mission and provides refined solutions to meet the warfighters' training requirements.

References

- G.M. Alliger, R. Beard, W. Bennett, C.M. Colegrove, Understanding mission essential competencies as a job analysis method, in *The Handbook of Work Analysis Methods, Systems, Applications, and Science of Work Measurement in Organizations*, ed. by M.A. Wilson, W. Bennett, S. Gibson, G.M. Alliger (Routledge, New York, 2012), pp. 603–624
- E. Beidel, Air Force Chief some drones won't be coming home after Afghanistan. *NDIA Bus. Technol. Mag.* (2012). Retrieved from <http://www.nationaldefensemagazine.org/blog/Lists/Posts/Post.aspx?ID=815>
- H.R. Cantwell, Operators of air force unmanned aircraft systems: breaking paradigms. *Air Space Power J.* (Summer) (2009). Retrieved from <http://www.airpower.au.af.mil/airchronicles/apj/apj09/sum09/cantwell.html>
- Chairman of the Joint Chiefs of Staff Instruction, CJCSI 3255.01 (2009). Retrieved from website: <https://extranet.acq.osd.mil/uwir/docs/3255.01.pdf>
- A. Church, RPA ramp up. *Air Force Magazine.com* 94(6), (2011). Retrieved from <http://www.airforce-magazine.com/MagazineArchive/Pages/2011/June2011/0611RPA.aspx>
- C.M. Colegrove, G.M. Alliger, Mission essential competencies: defining combat readiness in a novel way, in *Proceedings of the NATO Research & Technology Organization, Studies, Analysis, and Simulation Panel, Conference on Mission Training via Distributed Simulation (SAS-38)*, Brussels, 2002
- Department of Defense, Department of Defense, Under Secretary of Defense, Acquisition, Technology and Logistics. *Report to Congress on Addressing Challenges for Unmanned Aircraft Systems* (2010). Retrieved from website: <http://www.acq.osd.mil/sts/docs/2010-uas-annual-report.pdf>
- Department of Defense, *Unmanned Systems Integrated Roadmap FY2011-2036* (11-S-3613) (2011). Retrieved from website: <http://www.dtic.mil/ndia/2011MCSC/Thompson-UnmannedSystems.pdf>
- Department of Defense, Under Secretary of Defense for Acquisition, Technology and Logistics, *Future Unmanned Aircraft Systems Training, Operations, and Sustainability* (7-3C47E5F) (2012). Retrieved from website: http://www.wired.com/images_blogs/dangerroom/2012/06/06132012_uas1.pdf
- S.L. Enewold, D. Crowley, Department of Defense, Joint Strike Fighter Program Office, *Joint Strike Fighter Update* (2006). Retrieved from website: https://dtic.mil/dtic/index.php?option=com_content&view=article&id=2235867

- J. Gertler, *U.S. Unmanned Aerial Systems* (Congressional Research Service, Washington D.C., 2012). <http://www.fas.org/sgp/crs/natsec/R42136.pdf>
- J.C. Gorman, N.J. Cooke, H.K. Pedersen, J. Winner, D. Andrews, P.G. Amazeen, Changes in team composition after a break: building adaptive command-and control teams, in *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*, San Francisco (Human Factors and Ergonomics Society, Santa Monica, 2006), pp. 487–491
- J.C. Gorman, N.J. Cooke, P.G. Amazeen, J.L. Winner, J.L. Duran, H.K. Pedersen, A.R. Taylor, Knowledge training versus process training: the effects of training protocol on team coordination and performance, in *Proceedings of the Human Factors and Ergonomics Society 51st Annual Meeting—2007*, Maryland (Human Factors and Ergonomics Society, Santa Monica, 2007) pp. 382–387
- Government Accountability Office, *Military Training: DoD's Report on the Sustainability of Training Ranges Meets Annual Reporting Requirements but Could Be Improved* (GAO-12-13R Military Training) (2011). Retrieved from GAO website: <http://www.gao.gov/assets/590/585836.pdf>
- House Permanent Select Committee on Intelligence, *Performance Audit of Department of Defense Intelligence, Surveillance, and Reconnaissance* (2012). Retrived from website: <http://intelligence.house.gov/sites/intelligence.house.gov/files/documents/ISRPerformanceAudit%20Final.pdf>
- C. Hoyle, *UAV Trial Shows Non-pilots 'Equally Skilled', Says Raf*. Retrieved from <http://www.flightglobal.com/news/articles/uav-trial-shows-non-pilots-equally-skilled-says-raf-355001/> (2011)
- Inzpire Develops Human Factors Training for RPAs Crews* (2012). Retrieved from <https://www.inzpire.com/news/inzpire-delivers-human-factors-training-to-rpas-crews>
- G. Jean, Teaching non-pilots to fly predators requires more cockpit hours in manned aircraft. *NDIA Bus. Technol. Mag.* (2010). Retrieved from <http://www.nationaldefensemagazine.org/archive/2010/February/Pages/TeachingNon-PilotstoFlyPredatorsRequiresMoreCockpitHoursinMannedAircraft>
- Joint Chiefs of Staff, *Close Air Support*. Joint Publication 3-09.3, Joint Chiefs of Staff (2009). http://www.fas.org/irp/doddir/dod/jp3_09_3.pdf
- Joint UAS Center of Excellence Report of Results, Unmanned Aircraft System (UAS), Integration into the U.S. National Airspace System (NAS) (2008)
- M.J. Martin, C.W. Sasser, *Predator the Remote-Control Ai War Over Iraq and Afghanistan: A Pilots Story* (Zenith Press, Minneapolis, 2010)
- A. Mulrine, UAV pilots. *Air Force Magazine.com* 92(1) (2009). Retrieved from <http://www.airforce-magazine.com/MagazineArchive/Pages/2009/January2009/0109UAV.aspx>
- N. Peterson, *Unlicensed Pilots Operating Drone Aircraft in Illinois Airspace* (2012). Retrieved from <http://news.medill.northwestern.edu/chicago/news.aspx?id=205534>
- Public Law 109–364, John Warner National Defense Authorization Act for Fiscal Year 2007, Section 941 (2007)
- B.T. Schreiber, W. Bennett Jr., *Distributed Mission Operations Within-Simulator Training Effectiveness Baseline Study. Volume I: Summary Report* (AFRL-HE-AZ-TR-2006-0015-Vol I), Air Force Research Laboratory, Warfighter Training Research Division, Mesa, 2006. <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA461866>
- N. Simmons, *Innovative Simulator Solution Enhances Training, Saves Millions of Dollars* (2012). Retrieved from <http://www.af.mil/news/story.asp?id=123311972>
- Under Secretary of Defense (Acquisition, Technology and Logistics). Department of Defense, *Department of Defense Report to Congress on Addressing Challenges for Unmanned Aircraft Systems* (2010). Retrieved from website: <http://www.acq.osd.mil/sts/docs/2010-uas-annual-report.pdf>
- R.A. Wisher, F.E. Hartman, C.G. Sanders, A. Ciavarelli. *Analysis of system training impact for major defense acquisition programs*. Informally published manuscript, Naval Postgraduate School, Monterey, (2011). Retrieved from <http://edocs.nps.edu/npspubs/scholarly/TR/2011/NPS-MV-11-001.pdf>

From Research to Operations: The PITVANT UAS Training Experience

105

Maria da Luz Madruga Matos, João Vieira Caetano,
José A. P. Morgado, and J. B. de Sousa

Contents

| | | |
|---------|---|------|
| 105.1 | Introduction | 2526 |
| 105.2 | R&D Methodology and Systems | 2529 |
| 105.2.1 | Systems Engineering Iterations | 2529 |
| 105.2.2 | Subsystems Development, Integration and Testing | 2531 |
| 105.2.3 | UAS Architecture | 2532 |
| 105.3 | UAS Team Instruction and Training | 2539 |
| 105.3.1 | Requisites and Types of Qualifications | 2539 |
| 105.3.2 | Theory | 2541 |
| 105.3.3 | Simulation | 2543 |
| 105.3.4 | Supervised Flight Training | 2543 |
| 105.3.5 | Concepts of Operations (CONOPS) | 2543 |
| 105.4 | Results | 2550 |
| 105.4.1 | Personnel Trained | 2550 |
| 105.4.2 | Flight Hours and Sorties | 2551 |
| 105.4.3 | Milestones | 2551 |
| 105.4.4 | Future Goals | 2552 |
| 105.5 | Conclusions | 2553 |
| 105.6 | Annex A – Operational Checklists and Procedures: Examples | 2554 |
| | References | 2560 |

M.L. Madruga Matos (✉) • J. Vieira Caetano • J.A.P. Morgado
Centro de Investigação, Academia da Força Aérea, Sintra, Portugal
e-mail: mlmadruga@academiafa.edu.pt; jvcaetano@academiafa.edu.pt; jamorgado@academiafa.edu.pt

J.B. de Sousa
Department of Electrical and Computer Engineering, University of Porto, Porto, Portugal
e-mail: jtasso@fe.up.pt

Abstract

The PITVANT Unmanned Aircraft Systems (UAS) Research and Development (R&D) program training experience is described, including the different crew positions and responsibilities, as well as the several training phases. This is done by presenting the systems engineering methodology behind all the developments carried by the joint team of the Portuguese Air Force Academy and the Porto University, together with the PITVANT UAS architecture and main features. The training program is then described including theoretical classes, written exams, and practical instruction and evaluation. In addition, the process of developing, testing, and validating concepts of operations and the correspondent operational procedures and their training with the involvement of end users is also reported. The current operational capabilities of the systems are explained, showing that the PITVANT program has achieved a level of maturity that enables testing and validation in near real scenarios. This leads to the next step in the systems spiral development that should include, in the near future, the possibility of technology transfer to the industry. The PITVANT program is currently on its fifth year of existence, and the several air vehicles have logged about 730 takeoffs adding up to about 320 h of flight, most of which at the Portuguese Air Force Airbase in Ota, which is permanently assigned to the PITVANT program. Several operators and pilots have been trained under the scope of the program, and currently the team of 20 people operates the different PITVANT UAS on a daily basis. The extremely low accident ratio, of about 0.5 %, is a good indicator of the effectiveness of the training schemes that the research team has implemented.

At the end of the program, the team expects that the UAS have reached a level of maturity that will allow industrialization.

105.1 Introduction

This chapter describes the current state of the PITVANT Unmanned Aircraft Systems (UAS) Research and Development (R&D) program, with special focus on training aspects. PITVANT is the Portuguese acronym for Research and Technology Project in Unmanned Air Vehicles (*Projeto de Investigação e Tecnologia em Veículos Aéreos Não Tripulados*). This project, developed under a grant from the Portuguese Ministry of Defense (MoD), is the largest UAS R&D program in Portugal, with a total duration of 7 years, from November 2008 until November 2015. The main contractors are the Portuguese Air Force (PoAF) and the Porto University. Regarding the PoAF, PITVANT is carried out by its higher education school: the Air Force Academy (AFA). The College of Engineering (FEUP) is the lead partner from Porto University.

Prior to the beginning of this project, AFA had already considerable experience in unmanned aircraft design, construction, and testing. The platforms were built for the purpose of in-flight testing of new technologies, such as smart structures, and eventually they became the starting point of the autonomous flight research at AFA. Although they served the first stages of the project quite well, further research

not only demanded extended requirements concerning payload weight and volume, endurance, and performance but also new aircraft concepts. This represented an opportunity to develop and improve skills in aircraft design. Owing to the evolution of optimization algorithms and computational analysis tools in the last decade, air platform-related research has been mostly conducted on multidisciplinary optimization.

FEUP has been designing, building, and operating unmanned underwater, surface, and air vehicle systems for innovative applications with strong societal impact since 1992. FEUP received the Espírito Santo Bank (BES) Innovation National Award for the design of the Light Autonomous Underwater Vehicle (AUV). In the last 15 years, researchers from FEUP have successfully fielded unmanned air, ground, surface, and underwater vehicles in innovative operations in Europe and in the United States of America. These include some world firsts, such as the underwater rendezvous between the Aries and Isurus AUV, respectively, from the Naval Postgraduate School and FEUP, which took place in 2006 in Monterey, California, under a cooperation project between the two institutions.

The AFA/FEUP team is backed up by valuable international partners such as Honeywell, Embraer, Swedish Defense Agency (FOI), University of California at Berkeley, and the Armed Forces University, Munich. End users such as other military services, police forces, and natural disaster response authorities are also involved in the project, in order to provide inputs to the concepts of operations (CONOPS).

Hence, the four main objectives in PITVANT are:

1. To develop/test technologies for UAS
2. To develop new CONOPS for small/medium UAS
3. To test and evaluate the systems and technologies developed in civil/military operation
4. To carry out advanced training of UAS operation teams

To accomplish these objectives, PITVANT follows a systems engineering (SE) methodology. Technologies, CONOPS, and payload are developed/integrated, tested, validated, and matured on smaller and less complex UAS and then move on to larger and more complex platforms. All airframes are developed and built within the project.

Development of a specific training syllabus is not the common practice among academic R&D groups, but in PITVANT this was considered as a main objective. The fact that the PoAF is one of the main contractors, bringing into the project all its knowledge and institutional culture of flight safety, along with the expertise of FEUP in SE, naturally lead the team to the conclusion that a training program supported by strict operational procedures was the only way to perform a high number of flight hours, with a high inherent level of uncertainty, in both the experimental vehicles and in the novel control algorithms, with a low rate of accidents.

To better understand the requisites of training that the PITVANT team must deal with, the road map of development for the PITVANT is depicted in Fig. 105.1.

The first stage of the program addressed the upgrading of the existing remotely piloted aircraft into autonomous flight capable unmanned air vehicles (UAV). This stage included the integration and configuration of autopilots into different

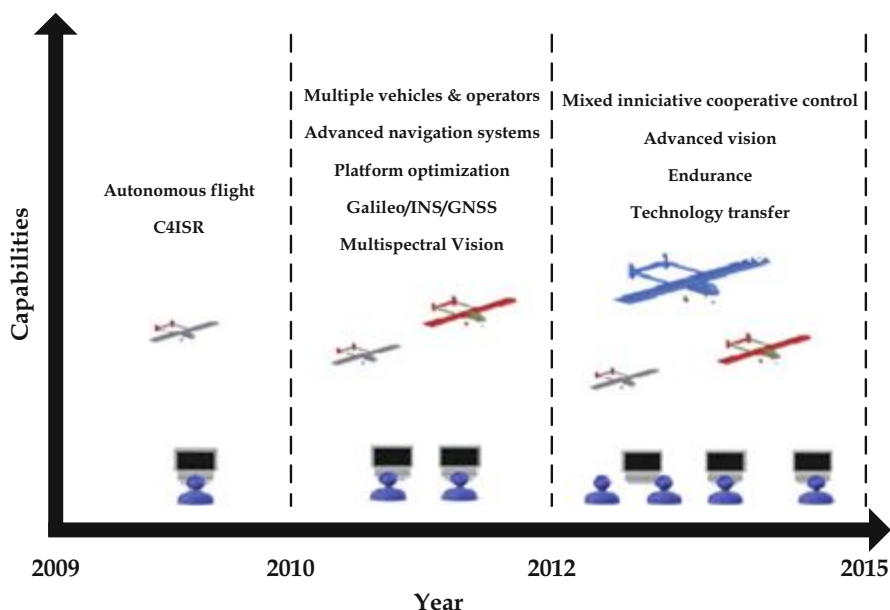


Fig. 105.1 PITVANT development road map

platforms, as well as the manufacturing of improved new platforms, needed for the cooperative control algorithm testing that would take place in latter stages of the program. At the same time, command and control (C2) systems were also being developed. By the end of 2010, the whole process from manufacturing of the platform to autonomous flight was fully developed and matured.

The second stage of the program is now in its final phase and comprises a set of activities that enable the autonomous platforms to perform relevant missions to both civil and military end users. The testing of novel control algorithms for target tracking, obstacle avoidance, and atmospheric thermal currents tracking are some examples of the work developed by the team. Flight testing of these algorithms was only possible due to the experience obtained at the first stage. Multi-UAV operation, characterized by having several UAV operating within the same confined airspace with a centralized C2 structure, was also undertaken, paving the way for cooperative control. Communications are currently being tested, to ensure the necessary range for the endurance tests that follow. Different sensors were integrated into the platforms to provide the data needed for these tests. These include different sorts of stabilized vision sensors, as well as onboard computing systems.

PITVANT's final stage will be devoted to mixed-initiative cooperative control, as well as to endurance tests with larger platforms. Advanced vision systems will allow the UAS to deliver relevant data to end users. By the end of the program, the UAS should be at a stage of development that will allow industrialization.

This chapter is divided in three main sections. Section 105.2 introduces the SE methodology, which is the backbone of the project, and describes the UAS architecture. Section 105.3 focuses on the training aspects. In this section the instruction and training for personnel assigned to different roles within the operational team are described, as well as their responsibilities. The training for maintenance of the systems is not considered in this chapter. The process of CONOPS development is also presented here, as it bears direct impact toward the training requirements. Samples of checklists and procedures that operators must follow for the different CONOPS are also presented. Section 105.4 presents the results achieved by the team to demonstrate how well adapted the training process and operational procedures are to the project goals. This includes the number of personnel trained since the beginning of the program, in-flight statistics and milestones, as well as future goals. Conclusions are then presented to summarize the most remarkable aspects of the systems under development as well as the training program, to highlight flight results and milestones achieved.

105.2 R&D Methodology and Systems

One of the key aspects leading to the PITVANT's good flight results is the use of a SE methodology. The present section will cover the PITVANT's R&D approach from a SE point of view.

The project is divided in two parallel phases (Campos et al. 2010), as shown in Fig. 105.2. The first one is the development of new technologies in a scientific environment. The second one concerns the systems' integration and the operational testing of the developed technologies. A SE approach is applied to both phases, ensuring the correct input-output relation and the maximum effectiveness.

105.2.1 Systems Engineering Iterations

One of the goals of this project is to integrate a wide variety of systems and subsystems into a more complex system, capable of performing autonomous missions in different operational environments. Hence, the concept of system of systems (SoS) will be used to better address the subject, on an SE approach.

The baseline for the definition of the requisites and the functional architecture of each subsystem is defined on the first iteration of the SE process, applied to the SoS. This encompasses the requisites' analysis, functional analysis, and synthesis.

A matrix of design parameters is also developed in parallel, connecting them to the various subsystems as well as the subsystems to each other. This way, should a subsystem need an update, the matrix will indicate the tree influence that specific subsystem has on other subsystems, allowing for the developer to have the information needed to consider compatible updates.

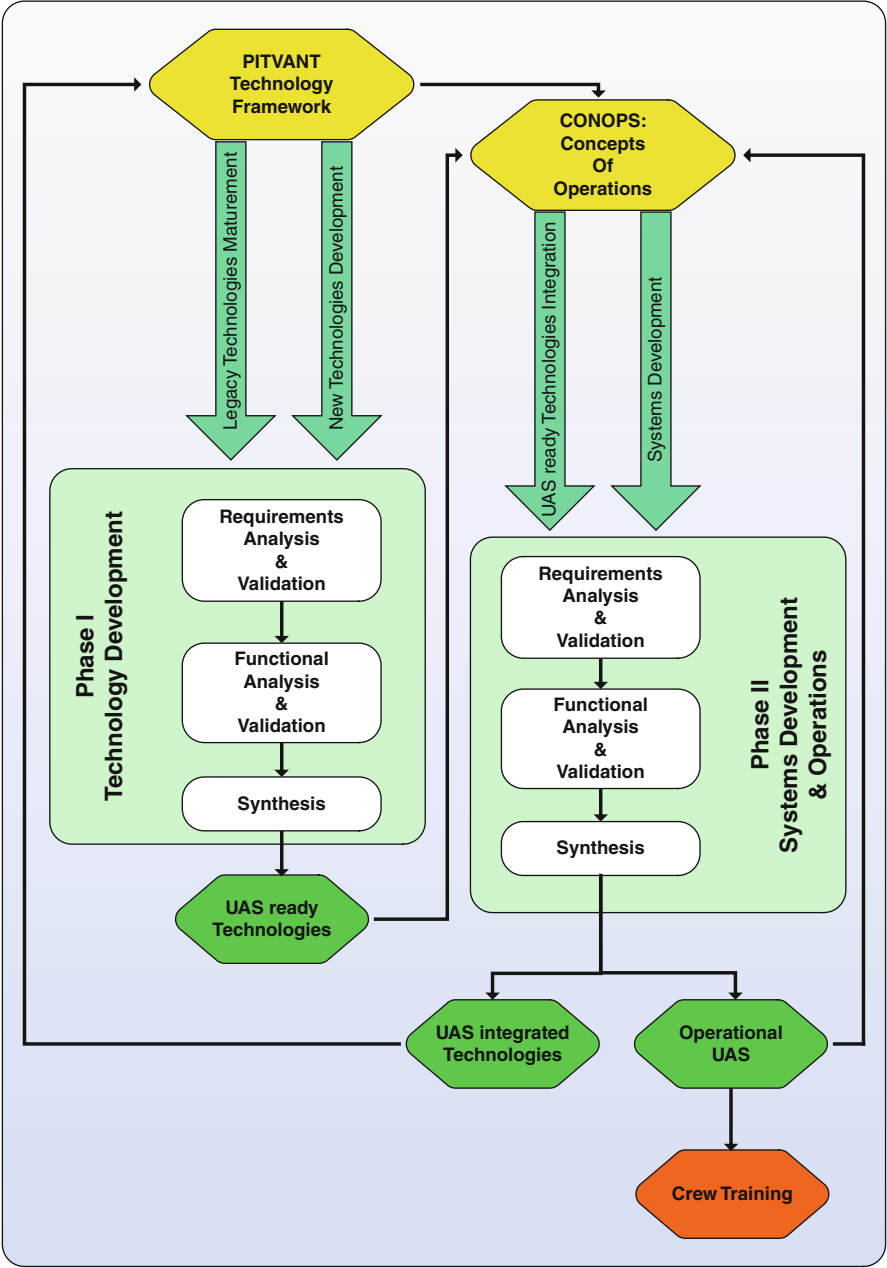


Fig. 105.2 Two-phase SE methodology applied to the PITVANT

Starting with these, seemingly basic, initial definitions, the SE process will be applied to the functional framework in an iterative way, covering all subsystems. At the end of each iteration, a new version should be created, tested, and documented.

The following paragraphs will describe briefly the three most important steps carried out for the system and for each of the subsystems in every iteration.

105.2.1.1 Requirements' Analysis

Referring again to Fig. 105.2, one can see that the requirements' analysis is the first step of each iteration for both phases. The main goal of this step is to define the (sub)system's capabilities, performance, working environments, human-machine interfaces, and, finally, the aspects that affect the development of the system. These considerations have a baseline of requisites that defines the goals and the means of efficiency, further used for the following steps of the process.

105.2.1.2 Functional Analysis

The functional analysis is the second step in the process. Its primary goal is to interpret the requisites and convert them into a functional architecture, dividing each role of the system into sub-roles of the dependent subsystems.

105.2.1.3 Synthesis

This final step defines the physical and the design solutions, identifying the subsystems that are actually capable of satisfying the requisites. Hence, the functional architecture is converted into physical architecture, presenting the first arrangement of the subsystems, their interfaces, and even design restrictions that have already been identified.

105.2.2 Subsystems Development, Integration and Testing

After the synthesis, the designer is able to develop the matrix of design parameters that will serve as the input for the following steps. These are similar to the three steps explained before, applied to each subsystem. Each of the iterations on the subsystems should output the support tools, the integration methods and procedures, as well as the versioning updates.

The development phase of new subsystems culminates with their individual testing being followed by their integration and final systems testing. In order to guarantee that nothing is left aside, the team develops test and integration procedures for each step and version of the subsystem. After passing all tests, training procedures are developed and carried out in an iterative way, to allow the systems to be tested on the field and in-flight, assuring that the test conditions are as similar as possible to the final operational theater.

This step ends with the subsystems' hardware-in-the-loop testing. The new technology and algorithms can be tested in a hardware testbed that is specifically developed for this purpose, simulating the real operational environment as much as possible.

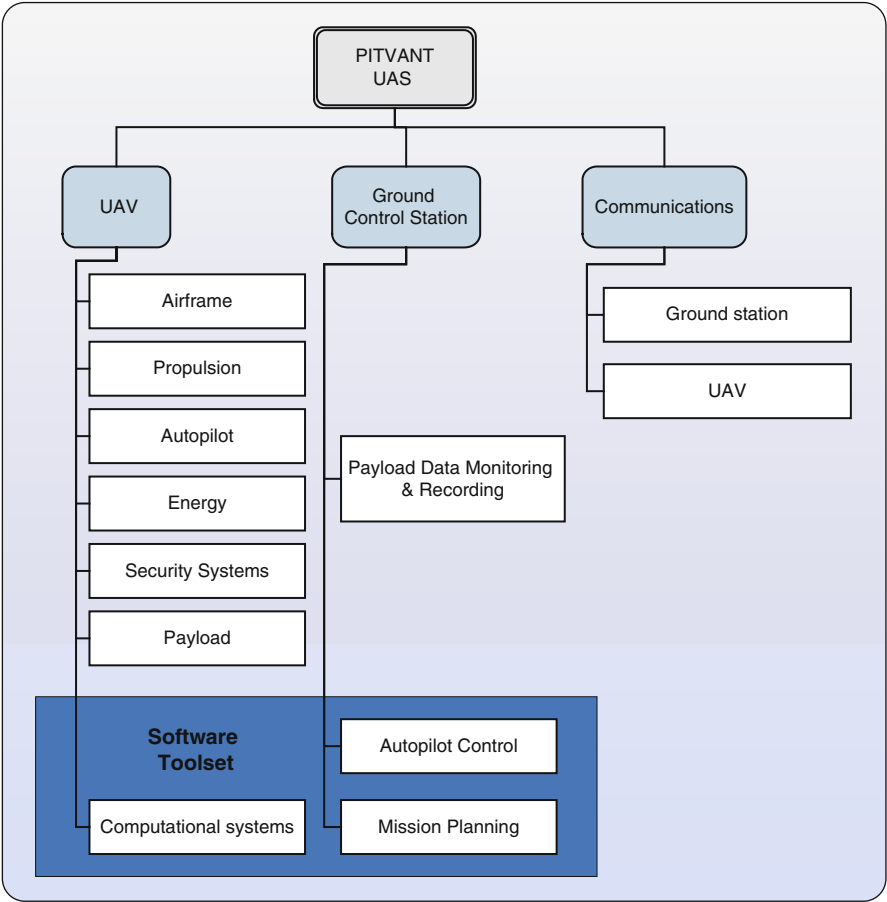


Fig. 105.3 PITVANT UAS system breakdown structure

105.2.3 UAS Architecture

This section presents the PITVANT project’s operational architecture, as it bears a direct impact with training and operational aspects of the UAS deployment in the field. It will cover the three main subsystems presented on the systems breakdown structure (SBS) of Fig. 105.3: UAV, ground control station, and software toolset.

Figure 105.4 presents the UAS functional architecture, for both training or operation environments. As the PITVANT is a research project, the team is always in constant training and procedures’ development: new technologies require new concepts, and UAS technologies require fail-proof safety and airworthiness.

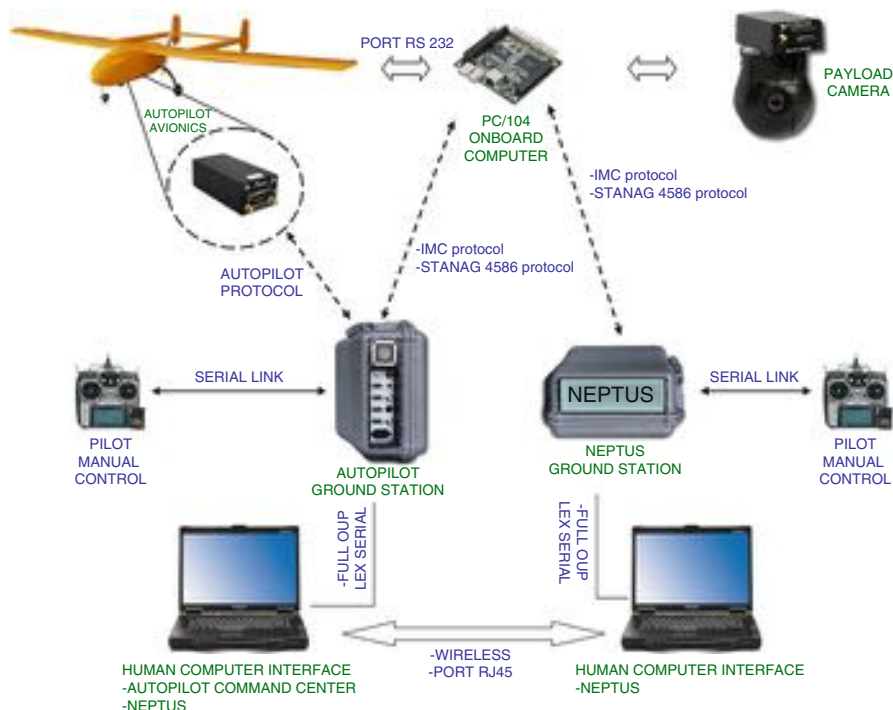


Fig. 105.4 PITVANT UAS functional architecture

105.2.3.1 Ground Control Station

The ground control station (GCS) is composed of the UAV ground station, the off-board control computers, the UAV operator, and the onboard systems' operator hardware, such as receivers, data links, and tracking stations/antennas. The communication between the UAV and the GCS is assured via a dedicated data link to the autopilot via the dedicated ground station (GS). It can have two different configurations, depending on the GCS – fixed or portable. The GS is connected to the operator's computer and the operator interface. The GS is also connected to a pilot console, in order for the pilot to fly the UAV, when in manual mode. Fig. 105.5 shows the portable version of the GCS used in PITVANT.

105.2.3.2 Unmanned Air Vehicles

The PITVANT's objectives lead to the need for developing several platforms for testing of different CONOPS and payloads. The PITVANT comprises different UAV that can be categorized according to the United States Air Force Tier categories. All of the listed UAV were fully designed and built at the AFA Research Center (CIAFA). CIAFA is not an industry-related entity, nor is it an aeronautical company.

Fig. 105.5 Fast deployable campaign GCS



Table 105.1 Summary of the PITVANT’s UAV characteristics

| | Flying wing | Mini-UAV | ALFA | ALFA Extended | ANTEX |
|--------------------|------------------|------------------|-------------------------|-------------------------|------------|
| Wing span (m) | 2 | 2 | 2.4 | 3.5 | 6 |
| MTOW (Kg) | 3 | 4 | 13 | 25 | 130 |
| Payload (Kg) | 0.9 | 1.9 | 4 | 10 | 30 |
| Engine | Electric | Electric | Electric/ combustion | Electric/ combustion | Combustion |
| Endurance (h:m) | 01:00 | 01:15 | 02:00 | 03:00 | 05:00 |
| Max speed (kts) | 50 | 50 | 60 | 65 | 80 |
| Range (km) | 40 | 60 | 130 | 190 | 390 |
| Number of aircraft | 02 | 01 | 07 | 01 | 02 |
| Obs | Hand launched | Hand launched | Catapult launched | – | – |

This fact makes it even more important to have a good training methodology and procedures. The operational teams are instructed in order to be able to operate and maintain all UAV in the fleet. The following sections will cover all UAV types operated in the PITVANT as technology demonstrators and testbeds.

Depending on the UAV configuration, there can be one or more onboard computers in parallel with the autopilot. These computers monitor the tasks the autopilot is not able to perform and can also be used for parallel monitoring or data-link settling and control. Their primary task is to serve as dedicated imagery processors and to allow the tests for the cooperative control of multiple platforms, autonomous target tracking, and obstacle avoidance algorithms. Here, the developers can also implement the software toolset mentioned in the following section.

A summary of the main platform characteristics is presented on Table 105.1.

Tier N/A: Micro/Small UAV

The PITVANT has developed two different configurations of this type of UAV: a flying wing and a conventional airplane configuration, with wing and tail.



Fig. 105.6 Flying wing being launched and in flight

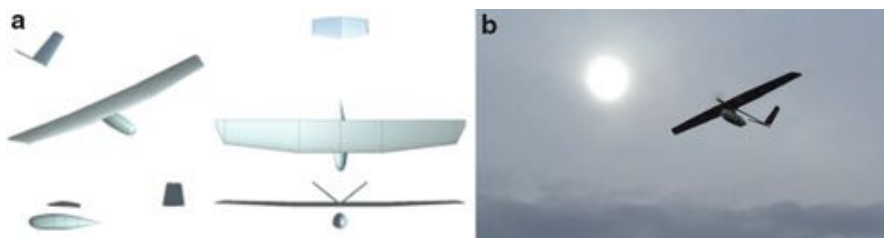


Fig. 105.7 Mini-UAV

Figure 105.6 shows the flying wing UAV subclass. It is a 3-kg maximum takeoff weight (MTOW) UAV. It can be equipped with two fixed cameras and live-stream video links, carrying the autopilot.

The UAV presented on Fig. 105.7 was specially developed for the war fighter in the field. It can be carried in a $0.6 \times 0.45 \times 0.25$ m backpack, set up, and hand launched in less than 5 min.

With 4.5-kg MTOW and a maximum payload mass of 1.900 kg, it can fly for over 1 h at a speed of 15 m/s. It can be configured with a wide range of electro-optic (EO) sensors with a minimum field of view (FOV) of 2° and infrared (IR)-stabilized sensors (FOV down to 12.3°), with digital encrypted video data link with 8 Mbts/s and onboard HD recording. The UAV recovery can be done at the push of a button with three configurable landings, (i) net recovery, (ii) belly landing, and (iii) deep stall, as the main parts separate from each other to dissipate energy. It can be operated in three different control strategies: (i) launch and forget (mission configuration prior to launch), (ii) autonomous flight (control with introduction of references), and (iii) fly by wire. It can also be operated in line or beyond line of sight.

Tier 1: Tactical UAV

In the Tier 1 category, the PITVANT counts with a total of ten UAV, with three different configurations. These differ in size and performance. The smallest within this class is the ALFA, with 5-kg of payload mass for a 13-kg MTOW UAV.

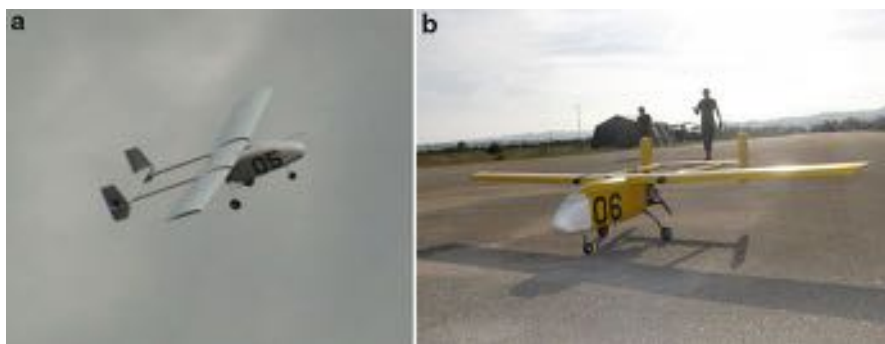


Fig. 105.8 ALFA #05 in flight and ALFA #06 after landing

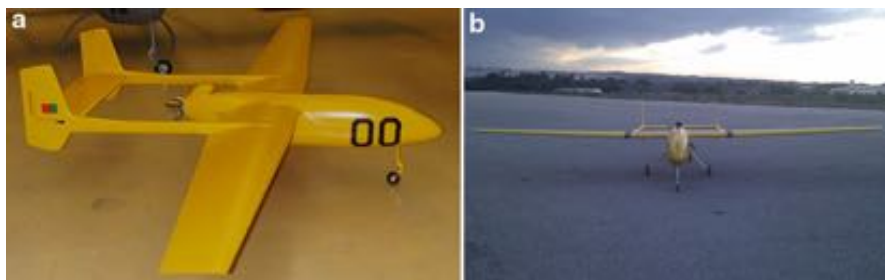


Fig. 105.9 ALFA Extended

This high-payload small UAV was developed with the main goal of serving as the primary testbed for the first iterations of the subsystems. There are seven UAV similar to the ones presented in Fig. 105.8.

It can also be operated with the same control strategies of the previous UAV, as well as point-to-coordinate fly by camera.

In terms of weight and size, the following UAV in the PITVANT is the ALFA Extended shown in Fig. 105.9. This is an optimized version of the ALFA. It was developed for high-efficiency flights and greater performance. It can carry 9 kg of payload for a total of 25-kg MTOW. This was the first PITVANT UAV to fly with a class C transponder. As it is the first of its class with these requisites and is designed to be the testbed for most of the forthcoming developments, it was painted yellow, to ensure a good visual signature.

The larger platform, the ANTEX, has 5.5-m span and MTOW of 130-kg. It has been developed for medium-altitude medium-endurance missions, such as Intelligence Surveillance and Reconnaissance (ISR), and is intended as a maritime search and rescue asset. The PITVANT has two of these platforms under tests and certification. This UAV is shown in Fig. 105.10.



Fig. 105.10 ANTEX #01

The PITVANT's engineers are developing a newer optimized 7-m span medium-altitude medium-endurance UAV to be used at the latest phases of the project and to be deployed in teams of maritime search and rescue UAV with mixed-initiative cooperative control.

105.2.3.3 Software Toolset

The PITVANT UAS use the Neptus/IMC/Dune software toolset, developed at FEUP, for the development of the UAS command and control applications. This toolset is easily configured for deployment in heterogeneous UAS to carry out mixed-initiative missions. This is because of both the modular design and the use of the publish/subscribe framework for communications.

Neptus is a command, control, communications, computer, and intelligence (C4I) framework. The interactions with human operators are categorized according to the phases of a mission life cycle: world representation, planning, simulation,



Fig. 105.11 Neptus user interface

execution, and post-mission analysis. The planning of a mission is supported by the mission planner. It provides a map editor that allows the construction of a virtual representation of the mission site. Maps are composed of several geo-referenced elements: marks to denote points of interest, geometric figures, surface images (with optional elevation map), generic 3D models, and paths (composed by line segments). For the execution phase, Neptus provides support for the creation of operator consoles that can be configured in terms of displays and controls. An example of such a console is presented in Fig. 105.11. After mission execution, all collected data can be analyzed using the Neptus' mission review and analysis tool. The collected data can be displayed in different ways such as tables, plots, and color maps. This feature is also available for mission replays. Moreover, an embedded web server can also be activated in order to provide web access to real-time mission data and post-mission analysis.

Neptus supports different communications protocols by having a specific communications layer that gives support to message-based protocols, regardless of the chosen transport protocol. It also supports distributed networked vehicles operating in wide areas, through the help of the inter module communications (IMC) message protocol that defines a common control message set, understood by all types of systems (vehicles, consoles, or sensors) in networked environments. IMC provides support for interacting with other autonomous vehicles, such as submarines or surface vessels, and has inbuilt functions that allow communication using the NATO Standardization Agreement STANAG 4586. This important feature makes it

possible to better manage the deployed systems in the field, allowing the operators to have access to information from each other's systems, as UAS or, for instance, unmanned underwater vehicles.

105.3 UAS Team Instruction and Training

This section will focus on the training method and syllabus used in the PITVANT. It will also present CONOPS that the team has developed for the UAS described in the previous section. A sample of the tests the UAV operators, configurators, and flight managers/tactical coordinators have to pass to complete their qualification are included in annex A.

105.3.1 Requisites and Types of Qualifications

The operational team is currently composed of the following positions:

- (a) *Pilot (PIL)* – Person responsible for remote-control flying of the UAV within vision range. This person will be responsible for taking off and landing the UAV in regular test conditions, whenever the mission requires manual flying instead of automatic launch and recovery. The pilot's role is even more important and noticeable in case of in-flight emergencies or subsystems malfunctioning.
- (b) *UAV operator (OP)* – Person responsible for managing the UAV flight, in autonomous mode, and monitoring and controlling the mission objectives in all CONOPS.
- (c) *Systems' operator* – Person responsible for the onboard systems, whether these systems have already been fully integrated in the UAS or are under integration process other than the autopilot. He/she is also responsible for the UAV's collected video imagery, onboard computer behavior, and other data links, as video, parallel telemetry data, or running algorithms. This position can be occupied by an operator if he/she is also a developer of the new onboard subsystems.
- (d) *First-line maintenance (FLM) technician* – These team elements differ in number depending on the UAS and mission being performed. On a regular operation, the FLM is occupied by a single person. This element is responsible for the UAV engine start-up and runway checklists, until the UAV is authorized to takeoff/be deployed.
- (e) *Flight manager (FM)/tactical coordinator (TACO)* – This position is occupied by a senior OP, trained and approved for the position. In case on multiple UAV and air traffic controller-/air command-dependent operations, he/she assures the correct mission performance and coordination of the UAV teams with the land or manned aircraft crews.

- (f) *Configurator* – Responsible for the correct configuration of the UAS in all CONOPS and for all new tested subsystems.
- (g) *Instructor* – The last stage of operational responsibility in the PITVANT. This person is responsible for the correct instruction, training, and evaluation of the operator trainees.

The team's structure reflects the fact that the UAS are under development, therefore the engineers must be a full part of the team, as suggested in Austin (2010). Another key point concerning the training program is the type of UAV operation. Active piloting of the UAV is only performed by the PIL for a very short time, during takeoff and landing, if the CONOPS don't require fully autonomous operation. When in autonomous mode, the UAV flight is managed by the UAV operator, which does not have a direct control over the platform flight dynamics. This means that the UAV operator does not need to be qualified as a manned aircraft pilot, therefore reducing the overall training costs. In fact, the manned aircraft component of the UAV operators training is the most expensive one (Hoffman and Kamps 2005).

Each candidate to UAV operator, pilot, and maintenance technician should fulfill a set of (pre)requisites:

- (a) Possess personal qualities and technical and professional qualifications – Entering the PITVANT's team is difficult, as the candidates are required to possess technical and personal qualities that are evaluated for several years in the PoAF or thoroughly evaluated at FEUP. Since some of the work can be classified, the PITVANT project demands for highly honorable team members. From the PoAF side, the team members can be officers, sergeants, or corporals. Officers are researchers in the CIAFA and can arise to tactical coordinators (TACO), instructors, and configurators. Sergeants can be operators and pilots. Corporals mostly serve as FLM technicians. From the FEUP's side, the team members normally have an engineering or computer science background and go through systematic trials and interviews. After entering the PITVANT, they focus on the engineering and research parts of the project, having the same qualifications as the PoAF officers.
- (b) English level B1, according to the Common European Framework of Reference, equivalent to the NATO Bright Evolution Level 2.0–2.4.

As the PITVANT focuses on developing new technologies and subsystems instead of training the operational teams to be combat ready, the syllabus is adapted to the project's needs, specially addressing safety and systems airworthiness. This is in line with the usual procedures for developing new systems, as the training should be formatted in order to fill the gap between the actual crew skills and the ones required for the systems' thorough operation (Blanchard and Fabrycky 2006).

Table 105.2 presents the time fractions spent in each phase of training and is in line with the findings of the European-funded study concerning the assessment of technology needs for a combat UAV (Ribeiro 2008).

The time percentage for each phase translates into the minimum requirements in terms of simulator, supervised training, number of flights, and flight hours per operator category that are presented in Table 105.3.

Table 105.2 Percentage of time allotted to each training phase

| | | |
|----------------------------|----------------------|------|
| Theory | Principles of flight | 10 % |
| | Operation | 30 % |
| Simulation | | 20 % |
| Supervised flight training | | 40 % |

Table 105.3 Operator categories requisites

| | Operator | Systems' operator | Flight manager/TACO | Configurator | Instructor |
|-------------------------------|----------|-------------------|---------------------|--------------|------------|
| Simulator hours | 10 | 10 | 20 | 25 | 25 |
| Supervised training hours | 10 | 10 | 15 | 20 | 30 |
| Supervised UAS configurations | – | – | – | 20 | 30 |
| Number of flights | 20 | 20 | 40 | 50 | 60 |
| Real-time operational hours | 10 | 10 | 25 | 25 | 35 |

The instruction and training steps described in the following paragraphs are applicable to the positions of operator (both UAS and systems), instructor, and TACO.

105.3.2 Theory

As presented in Table 105.2, the theoretical training comprises principles of flight and operational theory. This is compliant with the STANAG 4670, except for the air traffic control (ATC) procedures and rules of the air. Since PITVANT UAV only fly in segregated airspace, coordination with manned aircraft or ATC is not necessary. However, whenever needed, the PoAF supports the PITVANT operations with their ATC personnel. The principles of flight theory used in PITVANT are adapted from the theoretical knowledge requirements for the Airline Transport Pilot License (ATPL) as stated by the Joint Aviation Authorities (JAA). The topics that are covered are:

- (a) *Principles of atmospheric behavior*
 - (i) Atmosphere
 - (ii) Winds
 - (iii) Rain
- (b) *Fundamentals of aerodynamics for fixed-wing aircraft*
 - (i) Aerodynamic forces and moments
 - (ii) Tridimensional effects
- (c) *Flight stability*
 - (i) Longitudinal static stability
 - (ii) Directional static stability

- (iii) Lateral static stability
- (iv) Longitudinal dynamic stability
- (v) Lateral-directional dynamic stability
- (d) *Maneuvers and flight performance*
 - (i) Propulsion
 - (ii) Gliding flight
 - (iii) Leveled flight
 - (iv) Ascending and descending flight
 - (v) Coordinated turn
 - (vi) Pull-up
 - (vii) Flight envelope
 - (viii) Basics of control
 - (ix) Longitudinal control
 - (x) Lateral control
 - (xi) Directional control

After the first set of courses, the operator trainees proceed to operational-related courses. As presented in Table 105.2, the operational instruction comprises almost one third of the time an operator takes to be certified in UAS. This instruction covers the following subjects:

- (a) Chain of command
- (b) Steps to autonomous flight
- (c) Autopilot user's guide
- (d) Integration guides
- (e) Platforms
 - (i) Characteristics
 - (ii) Capabilities
 - (iii) Flight envelope
 - (vi) Limits
 - (v) Weather conditions
 - (vi) Platform checklists
- (f) Operational procedures
 - (i) Rules of conduct
 - (ii) General terminology
 - (iii) Phonetic alphabet
 - (iv) Critical situations
 - (v) Regular operational conditions (per CONOPS and platform)
 - (vi) Emergency procedures
- (g) Operational checklists for all CONOPS
 - (i) Flight readiness
 - (ii) Preflight checks and rechecks
 - (iii) In-flight
 - (vi) After-flight checks
- (h) Mission planning
- (i) Multiple team coordination

After all theory has been lectured, the candidates are tested. A minimum of 75 % is required in each test. Examples of checklists and procedures that are part of the testing related to the operational theory are presented in Annex A.

105.3.3 Simulation

After the theoretical instruction, the operator trainees go through the simulation training. In this phase, the candidates make use of the theory they have learned in the previous weeks in a mission simulator. The simulator is able to replicate the UAV flight dynamics, most of the operational environment difficulties, as well as some of the CONOPS. It is also possible to simulate turbulence, emergencies, and multi-crew training.

The simulation phase takes about a fifth of the full instruction course. After several hours of training, the operators subject themselves to the instructors' tests. These comprise general knowledge questions, operational/procedure-related questionnaires, and simulator-in-the-loop proficiency tests (execute procedures and checklists under a predetermined time). Operators must achieve more than 75 % approval in order to pass each test.

An example of a test the operator trainees have to pass is shown in Table 105.4.

105.3.4 Supervised Flight Training

Just before being certified to operate as part of the operational team, the operators in training go through supervised flight training. Here the trainees have to perform all tasks in the field, from UAV checks to postflight checks, passing by flight operation. This phase is complemented by a final assessment test that also includes stress reaction and behavior analysis.

Examples of evaluation topics for different flight phases are shown in Tables 105.5 and 105.6.

As mentioned on Sect. 105.3.1, there are several types of operational roles. After being certified as UAS operators for all CONOPS, the operators can proceed with instruction and training to become tactical coordinators and instructors/configurators.

105.3.5 Concepts of Operations (CONOPS)

The PITVANT mixes the R&D aspects of UAV development with their inherent operational nature in the field in a symbiotic environment. The development of new CONOPS takes an important role in development of all SoS in the PITVANT, making it paramount to show the importance and need of the use of UAV for several military and civil missions, in different scenarios.

CONOPS are defined as sets of technologies, procedures, and methodologies that are carried out in a specific operational scenario and conditions, with a predefined system.

Table 105.4 Example of simulator-in-the-loop questions

| Simulator in loop | Pass (V) | No pass (X) |
|---|----------|-------------|
| <i>System's windows</i> | | |
| <i>Aircraft</i> | | |
| Detect autopilot | | |
| Set pilot | | |
| Set active | | |
| What are the previous "set" for | | |
| <i>Ground station (ground station panel)</i> | | |
| Normal "GS serial" state? | | |
| Average power? (~ 4 W) | | |
| Min voltage? (13 V) | | |
| Request communication settings | | |
| Set channel | | |
| Set communication power | | |
| Normal RSSI? | | |
| Meaning of Ant, 3D, 2D, Acq? | | |
| What is the normal state of Ant? | | |
| Meaning of PDOP? (position dilution of precision) | | |
| What is "get spectrum" for? | | |
| <i>System panel</i> | | |
| Request communication settings | | |
| Set channel | | |
| Set communication power | | |
| <i>Telemetry</i> | | |
| What is the normal state of Ant? | | |
| Min voltage? (11,5 V) | | |
| Min servo voltage? (4,8 V) | | |
| Wind reading (where is it?) | | |
| Maximum temperature? (80°C) | | |
| <i>Preflight (meanings)</i> | | |
| Control surface test? | | |
| Control of (D)GPS? | | |
| Zero air data | | |
| Throttle hold and min? | | |
| <i>Mission limits (meanings)</i> | | |
| GPS time out? | | |
| Comm time out? | | |
| Pilot time out? | | |
| Lost comm number? | | |
| Land plan number? | | |
| <i>Advanced windows</i> | | |
| Command loops, substitute for which window? | | |
| Surface calibration (meaning)? | | |
| Sensor config – how to adjust? | | |
| Controller configuration? | | |
| <i>MAP</i> | | |
| Set up layers | | |
| <i>Set custom telemetry</i> | | |

Table 105.5 Before-flight proficiency test sample

| Evaluation topic | Pass (V) | No pass (X) |
|---|----------|-------------|
| Preflight checklist execution in less than 12 min | | |
| Preflight re-checklist execution in less than 4 min | | |
| Random quiz test on the procedures | | |

Table 105.6 In-flight proficiency test sample

| Evaluation topic | Pass (V) | No pass (X) |
|---|----------|-------------|
| Monitor behavior and stress | | |
| Response to malfunctions or emergencies | | |
| Response to simulated: | | |
| –Foreign aircraft incursion | | |
| –Power loss | | |
| –Erratic altitude tracking | | |
| –Others | | |
| Procedure follow-up | | |
| Overall evaluation | | |

105.3.5.1 CONOPS Development Process

As shown in Fig. 105.2, the CONOPS development is included in the SE methodology, as a response to the needs that arise from the development of technologies and subsystems. Another source for the development of CONOPS is the set of inputs from the end users that are participating in the program. A more detailed scheme for the CONOPS development process, linked with crew training, is presented in Fig. 105.12.

Any new test is carefully evaluated and discussed by the team and procedures and checklists are settled, so the operators can train beforehand. A very significant difference from operational systems can be perceived: the PITVANT team is not trained to operate the UAS in one defined way, but the type of operation is constantly evolving. This requires constant updates of the procedures and the improvement of the trained skills.

105.3.5.2 Simulation and Procedures Development Process

The safe operation of the UAS is the high end of all the work and research done backstage. A considerable part of this work is dedicated to the training of new CONOPS and the improvement of the existing ones.

All starts with the simulation of the future CONOPS in a controlled environment, with the inputs of the development team. At this phase, all integrated subsystems must be working flawlessly and must fulfill the requisites. After simulation, the initial procedures are outlined and thoroughly checked by the developer and operational elements of the team.

For some cases, it might be impossible to simulate the full CONOPS without having to actually fly the UAV. For instance, in catapult launch deployment, UAV mock-ups are launched before the real UAV to check for correct functioning of the

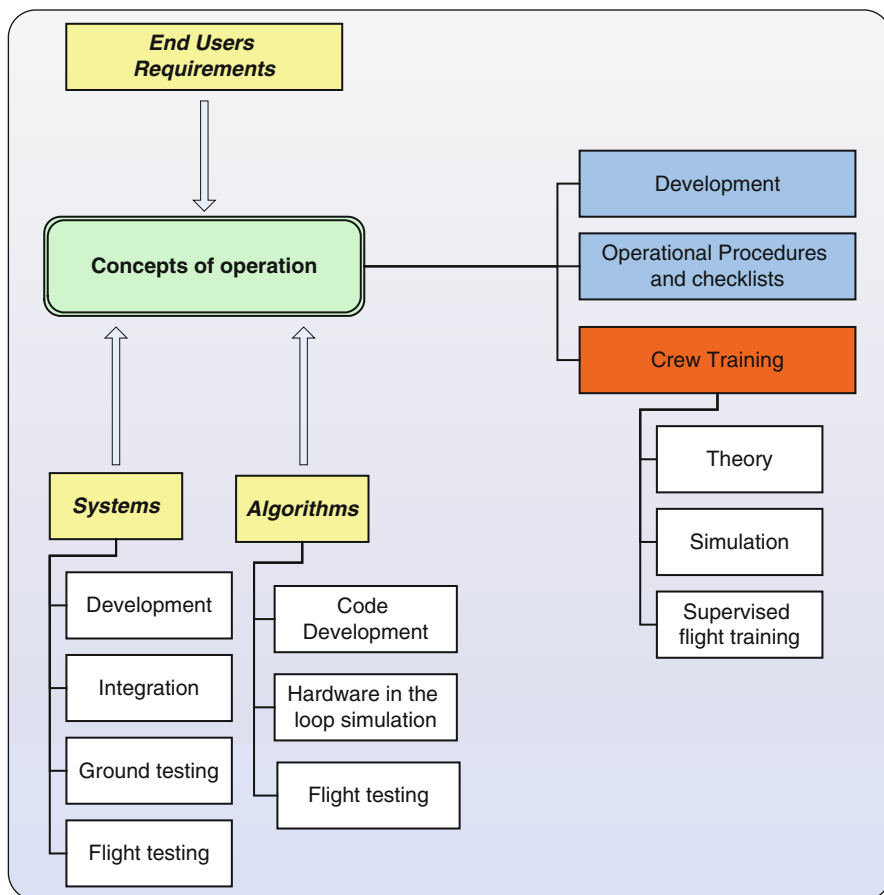


Fig. 105.12 Concepts of operations development methodology

catapult and coupling of the UAV. In the real UAV, acceleration tests are made, to simulate the launch and to detect high-tension points in the structure.

After the development of the operational procedures, checklists are elaborated, and both are executed and updated for an extended period of time, until all subsystems and operators are capable of executing the flight test, assuring the flight safety and airworthiness. Along the iterations, a set of conduct, communication, and emergency procedures are thoroughly tested and updated. This will minimize any collateral damage, should something go differently from what was planned.

105.3.5.3 Operational Testing

As presented on the lowermost blocks of Fig. 105.12, the last step is the operational flight testing. This phase is the most demanding one, despite being the tip of the sword in the UAS field. At this point, all aspects related to subsystems, communication, and operation must be at their final version and well implemented

in the field. By procedure, the senior operators and pilots (instructors) are the ones who carry out the first operational tests. Since November 2008, none of the 730 flights in the 320 flight hours resulted in any damage in the systems or UAV on the testing of a new CONOPS. A summary of the most important CONOPS, including a sample of requisites and procedures, is presented next. To keep this chapter to its intended size, CONOPS D to H are presented without their requisites and procedures.

Day Flight of Single UAV with Autonomous Landing

This operation is characterized by the daylight flight of a single UAV. Within it there are different operational scenarios. It is most commonly used for subsystems' testing and development. In the latest case, the operation will have a higher risk when compared to regular operation with subsystems that have already been fully tested and integrated.

Requisites:

- (i) UAV must be well configured with no error on altitude or indicated air speed (IAS) tracking.
- (ii) UAV must have been previously tested with virtual landing plans with excellent results – no lateral or height deviation from path.
- (iii) UAV must have the autoland path well configured and previously tested.
- (iv) Calibrated static sensor on the landing point, at runway altitude.
- (v) Certificated operator for autoland operation.

Before landing sequence:

- (i) Preflight window or altimeter window: uncheck control to GPS.
- (ii) Confirm touchdown waypoint altitude correct for autoland.
- (iii) Confirm UAV altitude is near altitude of land circuit entry.
- (iv) Request TACO authorization to autoland and inform landing runway.
- (v) Zero air data the barometric sensor to landing altitude.
- (vi) Update barometric reference on the active UAV.
- (vii) Select LAND NOW.

When the landing plan is activated:

- (i) Verify correct flight plan tracking.
- (ii) Inform/report TACO UAV final approach.
- (iii) Monitor flight path accuracy.
- (iv) If flight path track is good and autopilot does not abort.
- (v) When UAV passes decision waypoint, look out to see UAV attitude and altitude:
 - (a) If UAV lands, OP commands the pilot to transition to manual.
 - (b) If UAV reaches touchdown waypoint without landing and altitude offsets below 2 m, OP sets kill engine.
 - (c) Else, OP aborts landing.

Catapult Launch

Catapult launch is used to deploy a UAV without the need of a runway. So far, the PITVANT is using this CONOPS for UAV up to 13-kg MTOW, as shown in Fig. 105.13. This concept is not as efficient in larger UAS as it is on smaller ones,



Fig. 105.13 ANTEX-X02 #03 ready for catapult launch

as the larger normally require airfield landing or heavy launchers that are not suitable to be taken into the frontline. Another aspect is that larger UAVs have more endurance and fly faster than smaller ones making them more suitable for aerodrome operation, as opposed to the smaller ones that can be fitted for landing on unprepared ground.

This CONOPS allows for numerous tactical advantages, such as the deployment of UAV from ships, vehicles, and locations without a runway.

Night Operation

Night operation translates into UAV flights under reduced visibility due to sunset or night. It mainly differs from the daylight flight as it requires full automatic takeoff and landing, instead of the manual control that can be carried out in case of emergency or sudden change of flight conditions, as wind or safety. This CONOPS demands for new procedures and exhaustive training from the operational team.

Requisites:

- (a) GCS light level should be reduced before takeoff.
- (b) Engine final checks are performed by the frontline mechanics (aircraft illumination by a small light).
- (c) Runway lights close to the PIL should be off (40-m radius).
- (d) PIL has to be on a low light level for some time before starting the takeoff run (5 min).
- (e) A light should be near the flight manager/TACO so that the frontline mechanic holding the aircraft can see the hand signals.

- (f) Light or lantern should be used to signal the frontline mechanic.
- (g) If light is on, mechanic should hold the UAV for final checks.
- (h) When light is turned off, the mechanic lets the UAV free.
- (i) Constant update of the static pressure on the ground (for possible emergency landing).

Automatic Target Tracking/Following

This CONOPS is used for automatic target following based on a reference, and it can be added to the previous CONOPS. The reference can either be GPS beacon or video imagery based.

Nowadays it is quite common to have a vision sensor that is capable of following a target that corresponds to a marking on the image. However, in most of the UAS, the operator/pilot has to adapt the UAV flight path to the moving target's behavior. In this case, a tracking algorithm receives the GPS coordinates or image data from the target and automatically adapts the flight path to the target's ground speed and elevation and even corrects for sudden stops or changes in attitude. This can be used to track and follow targets, either friend or foe on land or sea.

This concept was fully developed by request from one of the end users participating in the project. Hence, it followed the development and testing standards mentioned in Sect. 105.2, introduced by the SE methodology. New procedures were developed, and all versions of the target tracking algorithm followed the block diagram in Fig. 105.13.

Intelligence, Surveillance, and Reconnaissance (ISR)

In this CONOPS, the UAV is equipped with vision sensors and real-time video broadcasting subsystems. The communication protocol allows for STANAG 4586 integration with other systems on the ground, sea, or air. For demanding missions, as persistent surveillance of fires or illicit activities, it is possible to equip the platforms with forward-looking or short wave infrared payloads.

The ISR missions must be very well planned and executed in order to ensure that the UAV does not fly in restricted or nonsegregated airspace. Should this need arise, the mission's TACO on the UAS side coordinates the activities with the air traffic controllers in the area. The UAS are also equipped with a class C transponder for this purpose.

UAV Handover

This CONOPS is characterized by the UAV control being passed to another operational team, deployed in the field. It is very useful for having UAV traveling long distances without being affected by communication losses as well as handing over the system to an operational team that will execute a sensitive mission in the field.

Beyond Line-of-Sight

Beyond line-of-sight (BLOS) operations imply the use of satellite communications, instead of the direct radio frequency data links. Sometimes, due to the distance

between the operational team and the UAV, it is necessary to use satellite communications for telemetry, control, and video images.

This CONOPS involves complex logistics and requires a great deal of experience from the operational team, being the CONOPS that requires more training and evaluation of all. This is the last step before operational deployment of the systems in a high-risk or dangerous theater of operations.

Multi-UAV Operation

Multi-UAV operation is an ever evolving CONOPS in the PITVANT. Referring to the last phase of Fig. 105.1, this CONOPS is evolving from two UAVs in the same airspace with separate GCS and operating teams. It is being upgraded and tested to have several heterogeneous/different platforms in the field being controlled by a single operational team. This difficult and highly demanding objective is intended to be one of the final goals of the PITVANT and is the one that uses all of the CONOPS mentioned before.

The extensive list of technology developments, procedures, and training hours should lead to the point where a mixed-initiative (UAV operator) cooperative control of a team of UAV will help the tactical and strategic commands to better decide and fulfill a mission. The applicability of this concept is very wide, going from simple fire surveillance over wide areas to search and rescue or ISR missions.

105.4 Results

Since the project kickoff on November 2008, the team has achieved several important results and milestones. The training program, combined with the careful development of the CONOPS and related procedures and checklists, as described in the previous section, allowed the team not only to grow into a considerable number of trained people but also to log an impressive number of flight hours and sorties for a non-industry-related R&D group.

Flight testing is mostly conducted in the Ota Air Base, now fully committed to UAS testing by the PoAF. A secondary test site is the Santa Cruz Municipal Airfield. Conveniently located near the shoreline, this test site is used for the ISR CONOPS, as it allows flying over the ocean. The PITVANT UAS always fly in segregated airspace. Nevertheless, class C transponder integration and testing was already done, paving the way for integration in controlled airspace whenever the legal framework will allow it and following the recommendations of ICAO Circular 328.

105.4.1 Personnel Trained

The present crew size of 20 people allows the rotation of personnel in order to keep flight operations on a daily basis, in parallel with the scientific and technological developments done at FEUP and AFA. Table 105.7 shows the number of individuals trained for the different roles within the PITVANT team.

Table 105.7 Number of staff trained/in training

| Position | Elements trained | Elements in training |
|------------------------|------------------|----------------------|
| Pilot | 7 | – |
| UAV operator | 14 | 3 |
| Systems operator | 12 | – |
| Flight manager/TACO | 5 | 4 |
| First-line maintenance | 18 | 3 |
| Instructor | 2 | – |
| Configurator | 2 | – |

Table 105.8 Number of sorties, flight time, and mean time between alternative landings per year

| Year | Number of sorties | Flight time (hh:mm) | Mean time between alternative landings (hh:mm) |
|----------------------------|-------------------|---------------------|--|
| 2009 | 98 | 30:50 | 00:43 |
| 2010 | 118 | 44:53 | 02:14 |
| 2011 | 209 | 93:58 | 03:21 |
| 2012 | 278 | 142:00 | 08:54 |
| 2013 (until February 28th) | 27 | 07:59 | N/A |
| Total | 730 | 320:13 | – |

105.4.2 Flight Hours and Sorties

The missions, as the operational part of the development process, are thoroughly planned at the beginning of each year, with most of the primary objectives already defined. Over the years, there has been a considerable increase of mission time and flight hours. Table 105.8 presents the flight hours and sorties per year.

Keeping in mind that PITVANT is an R&D project with operational testing, so far, only three accidents occurred that resulted in the full loss of the UAV and its subsystems. This represents a rate of accidents of approximately 0.5 %. A committee was appointed to investigate the causes for each accident, concluding that the first two were related to failures in the autopilot (which is an off-the-shelf product). The last accident, on September 2010, was a result of a chain of human failures, related to integration of a new subsystem. Once the chain of human errors was identified, the procedures were revised and updated accordingly, resulting in no more accidents so far.

105.4.3 Milestones

The principal milestones achieved by the PITVANT team are presented in the Table 105.9.

Table 105.9 Pitvant milestones

| Year | Milestone |
|------|---|
| 2009 | Autonomous flight |
| | Automatic takeoff and landing |
| | ANTEX-X02 improvements |
| | NEPTUS/IMC operationally used |
| 2010 | Mini-UAV development |
| | ANTEX-X02 extended |
| | Stabilized video sensor subsystems |
| | Night operations |
| | Multi-UAV operations |
| | Automatic target track/follow |
| | UAV handovers |
| | BLOS/satellite operations |
| 2011 | Automatic obstacle avoidance |
| | ANTEX flights |
| | One hundred flight hours |
| | Digital encrypted data links |
| | Successful transponder integration |
| 2012 | Catapult launch |
| | Automatic flight formation of two UAVs |
| | Flight formation of two UAVs over a military parade |
| | Communications test over the ocean (40 Km) |
| | Flying in a GPS-mitigated environment |
| | Three hundred flight hours |

105.4.4 Future Goals

As the PITVANT team is preparing to enter the final stage of the project, steps are being taken to achieve important goals as a wrap-up of all the work and progress done so far.

105.4.4.1 Mixed-Initiative Cooperative Control

This CONOPS was already developed, as described in Sect. 105.3. At this point, the team is ready to operate two UAVs flying in the same airspace. Control algorithms are being developed to allow those UAVs to perform their mission in a cooperative way, allowing for both the human operator and the platform, through the onboard computer, to make decisions along the mission.

105.4.4.2 Maritime Surveillance Operations

Portugal has one of the largest economic exclusive zones (EEZ) and also has search and rescue responsibilities over the Santa Maria flight information region (FIR), one

of the largest FIR in the world. The maritime resources are so important for Portugal, that the government is taking steps toward the definition of an integrated maritime policy (Cunha 2011). In this context, maritime surveillance is an obvious and strategic application for UAS. The largest platforms, ANTEX, are being prepared for autonomous flight, in order to demonstrate to the Portuguese authorities the feasibility of including UAS in the national maritime surveillance effort.

In terms of crew training, there are some challenges related to this operation. Procedures and checklists must be developed for the ANTEX. As new test sites are likely to be used, namely, in Azores and Madeira archipelagos, local procedures must also be developed and trained by the team to keep the accident level as low as it is now.

105.4.4.3 Reliability Program

The flight records of the operation done so far allow for the beginning of a reliability program, as done for manned aircraft. This goal implies a new task for the team members that will be instructed to collect and register specific data to feed the reliability program.

105.4.4.4 Industrialization

As presented along this chapter, PITVANT is an R&D program that combines scientific and technological R&D with intense flight testing, including a detailed training program for the crew. As no Portuguese end user is currently operating UAS, the team intends to reach the end of the project at a level of maturity that will allow industrialization of the UAS developed. This would ensure that all the knowledge gathered during the PITVANT life cycle could be transferred to the Portuguese end users (and other country end users as well) along with the training program, CONOPS, and operational procedures. This is the reason why end users are involved in the project from the beginning, to ensure that their operational requirements are taken into consideration, easing the industrialization of UAS to fit their needs.

105.5 Conclusions

The PITVANT is an R&D project in the area of unmanned systems, following a two-phase systems engineering methodology: the development of technologies and the operational testing and validation of the new systems and concepts. The UAV fleet consists of two micro/small UAVs plus ten Tier 1 tactical UAVs ranging from 13 to 130 Kg of MTOW. The UAS also comprise the GCS and the software toolset that allows for mission planning and flight management, and also support the mixed-initiative cooperative control algorithms.

The SE methodology backed by a well-defined training program has allowed the team to achieve an impressive flight record for a non-industry-related R&D group. From the beginning of the project in November 2008 up to February 28th, 2013, the PITVANT UAS have logged about 320 h of flight in 730 sorties with only three accidents, a rate of accidents of less than 0.5 %. Upon investigation, the first two accidents were attributed to autopilot failure, while the last one was due to human failure. Procedures were revised and updated, and since September 2010 the PITVANT team never registered another accident.

The team members that carry the flight testing are selected and then subjected to a training program that comprises theory, simulation, and supervised flight training. For each phase of the training program, candidates are tested and evaluated. The team includes (i) pilots for visual line-of-sight remote control flying the UAV in manual mode, (ii) UAV operators to manage the flight in autonomous mode, (iii) systems operators for payload data management, (iv) TACO for coordination between two teams and with the ATC, (v) first-line maintenance technicians for UAV preparation before flight, and (vi) instructors and configurators. At this time, the operational team has 20 people, allowing for daily operation in parallel with the scientific and technological developments made at FEUP and AFA. The CONOPS developed so far meet a double objective: to test technologies and systems developed by the team and to answer real requirements of potential end users that are actively participating in the project.

So far, the trained team members have been able to assure the correct execution of the planned missions, through a correct planning of the developments and a good prioritization of the objectives. All missions are carefully prepared, including active logging of the events that occur, in order to debug and improve the subsystems, the procedures, and the training methods. Only this way have the team members been able to considerably increase the number of sorties and tests every year while decreasing the number of failures, emergency landings, and unexpected maintenance.

The team expects to reach the end of the project at a UAS level of maturity that will allow, through a process of industrialization, for the transfer of all the know-how related to the UAS development, technology, architecture, and operation to future end users. If the PITVANT UAS, training program, CONOPS, and operational procedures are put to use in real operations in the future, the PITVANT objectives will truly have been achieved.

105.6 Annex A – Operational Checklists and Procedures: Examples

This annex presents examples of checklists and procedures the operators have to know and execute thoroughly.

Checklists (the checklist's format was adapted to fit the present document)

UAV Operator - Flight Readiness Checklist

| | | | |
|----------------|----------------------|----------|--|
| Mission: | <input type="text"/> | Operator | <input type="text"/> |
| Plane | <input type="text"/> | Pilot | <input type="text"/> |
| Piccolo Number | <input type="text"/> | Date | <input type="text"/> / <input type="text"/> / <input type="text"/> |
| Flight number | <input type="text"/> | Time | <input type="text"/> : <input type="text"/> |

System Checks

Ground Station hardware

| | | | |
|--|----------------------|--|----------------------|
| Power | <input type="text"/> | GPS antenna | <input type="text"/> |
| OI computer connection (RS232) | <input type="text"/> | Handset cable | <input type="text"/> |
| UHF antenna (beware not to connect it on the place of the Iridium sat phone) | <input type="text"/> | Perfect Connection from GCS to pilot handset | <input type="text"/> |
| | | Iridium sat phone | <input type="text"/> |

Ground Station window

| | | | | |
|--|----------------------|------------------------------------|----------|----------------------|
| Verify ground station diagnostic | <input type="text"/> | Set communications power: | [0.01 W] | <input type="text"/> |
| Verify ground station voltage | [>13 V] | Verify ground station GPS 3D fix | | <input type="text"/> |
| Verify communications channel (request) [Ch _____] | <input type="text"/> | Verify ground station GPS PDOP(<3) | | <input type="text"/> |

System panel

| | | | | |
|--|----------------------|---------------------------|----------|----------------------|
| Verify communications channel (request) [Ch _____] | <input type="text"/> | Set communications power: | [0.01 W] | <input type="text"/> |
|--|----------------------|---------------------------|----------|----------------------|

Aircraft panel

| | | | | | |
|-------------------|------------|----------------------|---------------------|------------|----------------------|
| Set pilot address | [SN _____] | <input type="text"/> | Set active aircraft | [SN _____] | <input type="text"/> |
|-------------------|------------|----------------------|---------------------|------------|----------------------|

Pre-flight panel

| | |
|------------------------|----------------------|
| Verify correct vehicle | <input type="text"/> |
|------------------------|----------------------|

System panel

| | | | |
|-------------------------------|----------------------|--|----------------------|
| Verify firmware version match | <input type="text"/> | AP power input (Volt>11.5 V & [3 W<Pwr<5 W]) | <input type="text"/> |
| RSSI: -79 or -71 (3) | <input type="text"/> | Servo power input (Volt>4.8 V & Curr<0,100 A]) | <input type="text"/> |
| Link: 100 | <input type="text"/> | All servos moving voltage: [_____ V; _____ A] | <input type="text"/> |

Main Panel

| | | | |
|--|--------------------------|--|--------------------------|
| Electrical motor - "Engine off" active | <input type="checkbox"/> | Insert lost comm waypoint (WP 80-89) | <input type="checkbox"/> |
| Pre-launch mode active | <input type="checkbox"/> | Insert runway waypoints (WP 96 & 97) | <input type="checkbox"/> |
| Map(s) set up | <input type="checkbox"/> | Insert A2M transition flight plan (Alt 150-200m) | <input type="checkbox"/> |
| Takeoff direction set [_____]° | <input type="checkbox"/> | Insert emergency land plan (WP 90-95) | <input type="checkbox"/> |
| Insert flight boundaries (WP 70-77) | <input type="checkbox"/> | Insert flight paths (WP 0-69) | <input type="checkbox"/> |

Limits panel

| | | | |
|---|--------------------------|---------------------------------|--------------------------|
| Minimum altitude set [_____] m | <input type="checkbox"/> | Comm times out checked | <input type="checkbox"/> |
| Maximum altitude set [_____] m | <input type="checkbox"/> | GPS times out checked | <input type="checkbox"/> |
| Flight timeout set [_____:_____:_____] | <input type="checkbox"/> | Close Throttle uncheck | <input type="checkbox"/> |
| Land if timeout and lost comm checked | <input type="checkbox"/> | Drop Deadman Line uncheck | <input type="checkbox"/> |
| Lost Comm Waypoint Set [WP ____; Alt: ____ m] | <input type="checkbox"/> | Aerodynamic Termination uncheck | <input type="checkbox"/> |
| Landing Waypoint Set [WP ____; Alt: ____ m] | <input type="checkbox"/> | Deploy Parachute uncheck | <input type="checkbox"/> |

Takeoff security settings - Command Loops

| | | | |
|---|--------------------------|---|--------------------------|
| IAS speed reference (min speed + 2 m/s) [____] m/s | <input type="checkbox"/> | Cmd loops: Activate runway endpoint (96/97) | <input type="checkbox"/> |
| Set altitude reference (gnd altitude +100 m) [____] m | <input type="checkbox"/> | | |

Aircraft control Checks (Checked aircraft; Next to the ground station)

Communications flight settings

| | | | |
|--------------------------------------|--------------------------|---|--------------------------|
| Set GS communications power: [0,5 W] | <input type="checkbox"/> | Set Autopilot communications power: [1 W] | <input type="checkbox"/> |
|--------------------------------------|--------------------------|---|--------------------------|

Communications check at distance (About 300m from the ground station). RSSI signal should indicate close to the maximum reading (3)

| | |
|----------------------|--------------------------|
| RSSI: -79 or -71 (3) | <input type="checkbox"/> |
|----------------------|--------------------------|

Communications flight settings

| | | | |
|---------------------------------------|--------------------------|--|--------------------------|
| Set GS communications power: [0,01 W] | <input type="checkbox"/> | Set Autopilot communications power: [0,01 W] | <input type="checkbox"/> |
|---------------------------------------|--------------------------|--|--------------------------|

General

| | | | |
|--|--------------------------|---|--------------------------|
| Select manual control and verify it's displayed | <input type="checkbox"/> | Verify control surface trims. Adjust as necessary | <input type="checkbox"/> |
| Verify control surface movement directions. (Manual) | <input type="checkbox"/> | | |

Control surfaces (Preflight panel - "AP On" mode)

| | | | |
|---|--------------------------|---------------------------|--------------------------|
| Ailerons right: Right aileron up, Left aileron down | <input type="checkbox"/> | Steering wheel right/left | <input type="checkbox"/> |
| Ailerons left: Right aileron down, Left aileron up | <input type="checkbox"/> | Throttle open/closed | <input type="checkbox"/> |
| Elevators up/down | <input type="checkbox"/> | Flaps down/up | <input type="checkbox"/> |
| Rudders right/left | <input type="checkbox"/> | Zero Air Data (1) | <input type="checkbox"/> |

Sensors Telemetry

| | | | |
|---|--------------------------|------------------------------|--------------------------|
| Dynamic pressure (<15Pa with pitot tube protection) | <input type="checkbox"/> | INS test (2): Y acceleration | <input type="checkbox"/> |
| INS test (2): Z acceleration | <input type="checkbox"/> | INS test (2): Pitch up/down | <input type="checkbox"/> |
| INS test (2): Roll right/left | <input type="checkbox"/> | INS test (2): X acceleration | <input type="checkbox"/> |

| | | | |
|----------------------------------|--------------------------|--|--------------------------|
| Telemetry panel | | | |
| Verify Autopilot GPS 3D fix | <input type="checkbox"/> | AGL - Sensor on | <input type="checkbox"/> |
| Verify Autopilot GPS PDOP (<2,5) | <input type="checkbox"/> | AGL - Verify AGL Alt.: "From Sensor" & "Installed" | <input type="checkbox"/> |

| | | | |
|---|--------------------------|---------------------|--------------------------|
| Gains look sensible ("AP on" and "Flying" modes) - Command loops | | | |
| Blow test: 25 m/s + | <input type="checkbox"/> | Yaw damper (online) | <input type="checkbox"/> |
| Roll controller (Bank set to 0 deg/s) | <input type="checkbox"/> | | |

| | | | |
|---------------------------------|--------------------------|------------------------|--------------------------|
| Engine start preparation | | | |
| Fuel tank full/Batteries full | <input type="checkbox"/> | Ignition battery full | <input type="checkbox"/> |
| Refuelling cap on | <input type="checkbox"/> | Pre-launch mode active | <input type="checkbox"/> |

| | | | |
|---------------------------------|--------------------------|-------------|--------------------------|
| Operation team readiness | | | |
| Cellphones off | <input type="checkbox"/> | IM SAFE (5) | <input type="checkbox"/> |

| | | | |
|---|--------------------------|-----------------------------|--------------------------|
| Communications with TACO | | | |
| Flight info to TACO (duration, plan and mission type) | <input type="checkbox"/> | Request Engine Start (TACO) | <input type="checkbox"/> |

| | | | | | |
|---------------------------------------|---------|--------------------------|-------------------------------------|-------|--------------------------|
| Communications flight settings | | | | | |
| Set GS communications power: | [0,5 W] | <input type="checkbox"/> | Set Autopilot communications power: | [1 W] | <input type="checkbox"/> |

| | | | |
|--|--------------------------|---|--------------------------|
| Engine Checks (Aircraft seized) | | | |
| "Engine on" active | <input type="checkbox"/> | Test ground station kill (Commands panel) | <input type="checkbox"/> |
| Request Engine Start (mechanic/pilot) | <input type="checkbox"/> | Test pilot kill | <input type="checkbox"/> |
| Engine start time | : | Check sensor readings for signs of excessive noise due to engine vibration at different RPMs (Gyros readings should be under 20deg/s) | <input type="checkbox"/> |
| Mixture tuning | <input type="checkbox"/> | | |
| Check response to sudden changes in throttle | <input type="checkbox"/> | | |
| Set lower throttle limit (Limits panel) | <input type="checkbox"/> | | |

| | |
|---|----------------------|
| Aircraft readiness to flight (signature) | <input type="text"/> |
|---|----------------------|

| | | | |
|--|--------------------------|-------------------------------|--------------------------|
| Runway Checks (Aircraft rolling to start position of takeoff) | | | |
| Pilot handset memory check | <input type="checkbox"/> | Request Taxi: TACO & Pil | <input type="checkbox"/> |
| Zero Air Data (1) | <input type="checkbox"/> | RSSI: -79 or -71 (3) | <input type="checkbox"/> |
| Remove Pitot tube protection | <input type="checkbox"/> | Autopilot voltage: 11.5 V + | <input type="checkbox"/> |
| Communications power: GS and Autopilot | <input type="checkbox"/> | Start voltage supervision | <input type="checkbox"/> |
| Confirm Takeoff Direction & Runway clear (w/ pilot) | <input type="checkbox"/> | Request Takeoff (inform TACO) | <input type="checkbox"/> |
| Takeoff info to TACO (direction and altitude) | <input type="checkbox"/> | Takeoff warning | <input type="checkbox"/> |

| | | |
|---------------------|---|----------------------|
| Takeoff time | : | <input type="text"/> |
|---------------------|---|----------------------|

In flight Checks (Aircraft flying)

| | | | |
|--|--|---|--|
| Aircraft Flying mode | | Check V-rate [>0m/s] | |
| Navigation mode: INS/GPS or INS/SBAS | | Activate M2A waypoint | |
| Airspeed > Vstall | | Request M2A | |
| Altitude > 70m AGL | | Request TACO to proceed with mission plan | |
| Update Lost comm waypoint for mission safety | | | |

Pre-landing Checks (Aircraft flying)

| | | | | |
|--------------------------------|--|--|--|--|
| RSSI: -93 to -71 (3) | | | Flight info: (IAS, Engine, RPMs, Altitude, Wind) | |
| Link: > 80 | | | Confirm runway clear (w/ pilot) | |
| Autopilot voltage: > 10.5 V | | | Request A2M & checks (inform TACO) | |
| System: Servo voltage: > 4.8 V | | | Activation of runway endpoint or landing path | |
| | | | Request: Downwind, Base leg & Final info | |
| AUTOLAND | Verify GPS (3D fix) and PDOP(<5) | | Landing Warning | |
| | Navigation mode: INS/GPS or INS/SBAS | | Landing time : | |
| | Sensor readings - Gyros (<20deg/s) | | Pre-launch mode active | |
| | Static pressure update | | Electrical motor - "Engine off" active | |
| | Request landing authorization from TACO | | Engine kill time : | |
| | Confirm runway clear (w/ pilot) | | Lower communications power: [0,01 W] | |
| MAN | Activate automatic landing | | Gasoline engine - Turn ignition off | |
| | Motor-altitude supervision (check for flame-out) | | Any Failure / emergency? | |
| | Perfect Connection from GCS to pilot handset | | Emergency landing? | |
| | Request pilot landing direction | | Fill the flight log | |
| | Request landing authorization from TACO | | | |

Procedures

Auto to Manual (A-to-M) Transition

In-flight requirements:

- (a) All A-to-M transitions must assure that the Auto and Manual engine regimes are as close as possible and near the optimum flight speed.
- (b) All A-to-M transitions must be made in leveled flight.
- (c) In “Manual test transition” and “Manual transition for landing” – transition should be carried out at about 600 ft high, 300 ft away horizontally from the ground station, while in downwind, near the optimum flight speed.

Sequence of procedures:

- (a) The OP informs the PIL of the UAV state: attitude, altitude, velocity, and engine percentage input.
- (b) The OP informs the PIL that he can make the transition.
- (c) OP tracks WP 95 or 96 – the one ahead of the UAV.

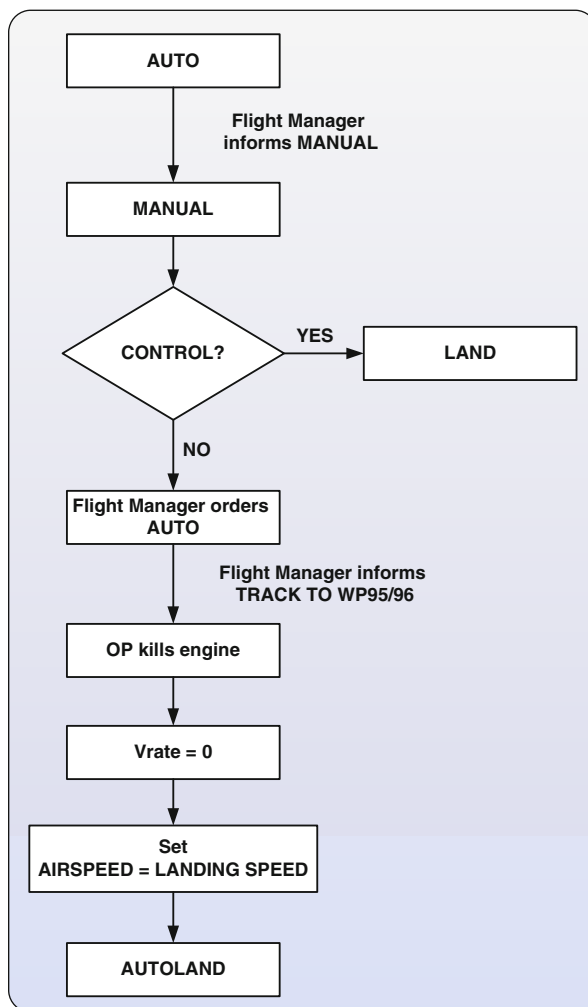


Fig. A1 Procedures for in-flight mechanical emergency

- (d) The PIL activates Manual and reports the transition.
- (e) The OP confirms that the transition has occurred, by checking the console.
- (f) PIL tests the UAV's response to pitch/camber, roll, yaw, and engine (in this sequence).
- (g) The OP should monitor the UAV attitude on the HUD and report if something is out of limits.
- (h) The crew monitors the UAV attitude changes. If no change occurs, OP orders PIL to activate Auto.
- (i) If PIL detects a failure in the transition to Manual, activate Auto immediately.

- (j) If the PIL and OP lose control over the UAV, OP shuts the GS off if ordered by the FM.
- (k) If control is assumed over UAV, initiate land plan.

Emergency – Auto to Manual with mechanical emergency on the UAV (except for elevator Failure)

In-flight requirements:

Before *manual* activation, all operation crew must be well informed of the emergency circumstances and limitations, if the time available for the response is enough.

Sequence of procedures:

- (a) OP requests PIL Manual activation.
- (b) PIL activates Manual, executes the Manual test procedures (3.1), and reports the conclusions to the OP.
- (c) OP informs Manual take over on the console.
- (d) If Manual is not assumed:
 - i OP orders PIL to activate Auto.
 - ii Depending on the response, OP shuts the GS off if mandated by FM.
 - iii FM requests OP to track WP 95 or 96, kill engine, set $V_{rate} = -0,2$ m/s, and force the airspeed to the landing setting.
 - iv OP monitors communications. If link below 40 or UAV below 60 m/200 ft, move WP 89 to the end of the runway, forcing UAV to be above the runway
- (e) If Manual is assumed:
 - (i) If in control, PIL lands the UAV.
 - (ii) If the UAV endangers the crew, buildings, or the material, PIL crashes the UAV on the safest crash area (on a predefined area, if possible).

References

- R. Austin, in *Unmanned Aircraft Systems*, ed. by I. Moir, A. Seabridge, R. Langton (Wiley, Chichester, 2010), p. 241
- B.S. Blanchard, W.J. Fabrycky, in *Systems Engineering and Analysis*, 4th edn. ed. by W.J. Fabrycky, J.H. Mize (Prentice Hall, New Jersey, 2006), pp. 505, 506
- H. Campos, G. Gonçalves, C.E. Oliveira, *Plano de Gestão de Engenharia de Sistemas-PITVANT PGES_v1.8* (Faculdade de Engenharia da Universidade do Porto, Porto 2010), p. 2
- T.P. Cunha, *Portugal e o Mar* (Fundação Francisco Manuel dos Santos, Lisboa, 2011), p.60
- J.C. Hoffman, C.T. Kamps, in *Air & Space Power Journal* (United States Air Force, Spring 2005), p. 36
- ICAO, *Unmanned Aircraft Systems, Circular 328* (International Civil Aviation Organization, Montréal, 2011), p. 29
- A. Ribeiro, *Assessment of Technology needs for UCAV – EUROPA MOU RTP 115.031 Final Report* (Instituto Superior Técnico, Lisboa, 2008), p. 30
- STANAG 4586, in *Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability*, prom., 2 edn. ed. by J.A. Moreno (NATO Standardization Agency, Brussels, 2007)
- STANAG 4670, Ed. 1: *Recommended Guidance for the Training of Designated Unmanned Aerial Vehicle Operator (DUO)*, prom., 1st edn. ed. by J.A. Moreno (NATO Standardization Agency, Brussels, 2009), p. 3

Section XIX

UAV Logistics Support

***Fulvia Quagliotti and
Nafiz Alemdaroglu***

Kimon P. Valavanis and George J. Vachtsevanos

UAV Logistics Support refers to UAV lifetime operation support; scheduling issues; delivery of goods and services; maintenance, testing, and fielding of UAVs; design and operation for reliability, safety, availability, and maintainability; logistics for ground station support and mobile UAV platforms; and potentially human operator support.

► **UAV Logistic Support Definition** by Chiesa and Fioriti presents considerations and hypotheses about RAMS characteristics and logistic support options for operational fleets of UAS. At first, an outline of why logistic support is relevant for UAS in comparison with other complex systems is presented, followed by considerations why reliability, safety, and maintainability are relevant aspects for UAS, explaining how such characteristics can be achieved by design. Synthetic considerations regarding transportability, a characteristic strictly connected to safety, reliability, maintainability, and logistic operations, are discussed. It is stated that considering (in addition to technical features) targets, requirements, constraints at the origin of a fleet of UAS, fleet sizing, and related system support (elements and strategies), a useful tool to face such complex problems is stochastic simulation and in particular a Monte Carlo methodology, which is also discussed.

► **UAV Logistics for Life-Cycle Management** by Karaagac, Pakfiliz, Quagliotti, and Alemdaroglu provides the reader with general knowledge on UAV Logistics for Life-Cycle Management, focusing on current logistics challenges and possible

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

future UAV logistics trends that may improve UAV deployment. Although UAVs do not have a pilot onboard, they require nearly the same logistics support as most manned aircraft. That is, logistics support covers support for all components including the unmanned aircraft and the ground control station. The objective is to design a system with a reduced logistic footprint, enabling fast deployment and high mobility.

Future integration of UAS into the national airspace will require clearly defined rules and procedures for UAV logistics support during every phase of the UAV life cycle, including being proactive in predicting and accommodating possible problems before they become serious issues.

Sergio Chiesa and Marco Fioriti

Contents

107.1 Introduction 2566

107.2 UAS Operational and Logistic Scenario..... 2567

107.3 UAS Support Systems Elements 2571

 107.3.1 R.A.M.S. 2573

 107.3.2 Logistics..... 2577

 107.3.3 Transport and Transportability 2583

 107.3.4 UAS Logistic Peculiarities 2587

107.4 Integrated Logistic Definition 2592

References 2599

Abstract

The chapter will present several considerations and hypotheses about R.A.M.S. characteristics and logistic support options for operational fleets of UAS.

First, a short review will outline how logistic support is relevant for UAS in comparison with other complex systems. It focuses on how logistic support must be designed and how such a design must be driven up jointly with the system design. The relevance of the scenario in defining logistic support will be discussed.

Next is a consideration of how reliability, safety and maintainability are relevant aspects for all advanced systems, and in particular for UAS, explaining how these characteristics can be achieved by design. The maintenance program will always be defined in the design phase and, consequently the logistic support will also be defined. Several tools useful for these purposes will be presented and shortly described.

S. Chiesa (✉) • M. Fioriti
Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy
e-mail: sergio.chiesa@polito.it; marco.fioriti@polito.it

Some synthetic considerations will be presented regarding transportability, a characteristic strictly connected to safety, reliability, maintainability and logistic operations, also of particular relevance for UAS.

Other peculiarities of UASs, mainly deriving from the flight segment and ground segment coexistence, will be presented.

Having in mind the targets, requirements, and constraints at the origin of a fleet of UAS, in addition to the technical features of the UAS, the fleet sizing and the related system support (elements and strategies) definition is a very complex problem. A useful tool to face this is stochastic simulation, in particular Monte Carlo methodology. A possible feature of a UAS fleet simulation model will be exposed and discussed.

107.1 Introduction

It is an established concept that in every technique, any complex system can operate for an extended time in a satisfactory way only if it is properly supported by a totality of elements that provide various support tasks, among which are:

- Providing the system (energy supply for operation devices and/or consumables necessary to perform a stated mission).
- Supplying the equipments, spare parts, staff training and technical handbooks to perform the aforesaid operations.
- Supplying the transport of the system, if necessary, and the above mentioned elements (at least the most necessary ones) in cases when the area in which the system works needs to be changed.
- To house, in suitable facilities, the activities, people and hardware involved, as well as, if necessary, the primary system.

With reference to Fig. 107.1, which takes as a typical example of a “complex system” a modern fighter aircraft, it is easy to understand the importance or the critical state of the logistic support system, and this under two aspects:

- The inefficiencies of the logistic support system strongly influence the results obtainable by the primary System.
- The cost of the logistic support system is very high, of the same size of the primary system costs.

In military and advanced industries, a potential cause of criticality has been found to arise as a consequence of placing primary system design before design of the logistic support system; as a matter of fact, in this way, the definition of the logistic support system is bound by the already-defined characteristics of the primary system, while modifications to improve efficacy and efficiency of the logistic support system cannot be made without alteration of relevant performances of the primary system, which could alternatively easily be introduced during the design phase of the system itself. Figure 107.2 shows how the sequential approach, first the primary system design and then the logistic support system, is obsolete,

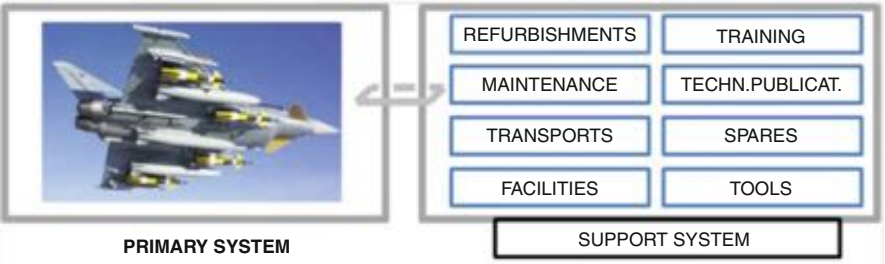


Fig. 107.1 Example of primary system and support system

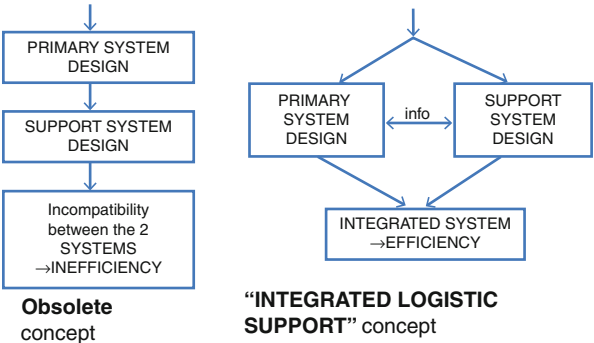


Fig. 107.2 Integrated logistic support (ILS) concept

in comparison with the suitable technique of developing both systems at the same time, with a continuous exchange of information.

The design process is undoubtedly more complex, but the advantage of obtaining an “integrated logistic support-ILS” is extremely more efficient, and this completely justifies the increased engagement in the design phase (Chiesa and Gianotti 1996).

Figure 107.3 points out that the importance of the logistic support system for any complex system is certainly valid for UAS as well, as they are typically complex systems; in particular, the concept of ILS is fundamental, even if it is not easily realized in a new field, at present in an evolving phase, and with peculiar characteristics. In the next section (Sect. 107.2), peculiarities of this system will be briefly discussed.

107.2 UAS Operational and Logistic Scenario

To achieve the best logistic support for any product or complex system involves a range of subjects, methodologies, and means to carry out in the best way everything

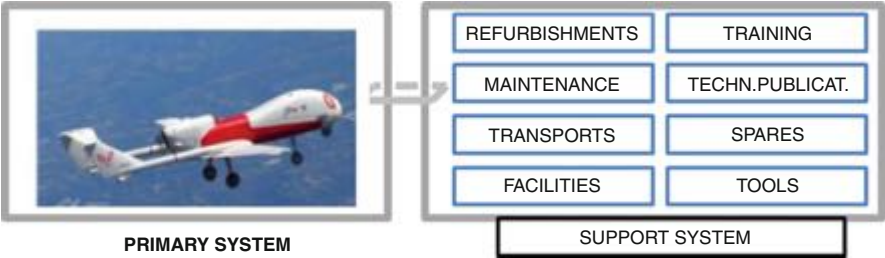


Fig. 107.3 UAS and related logistic support system

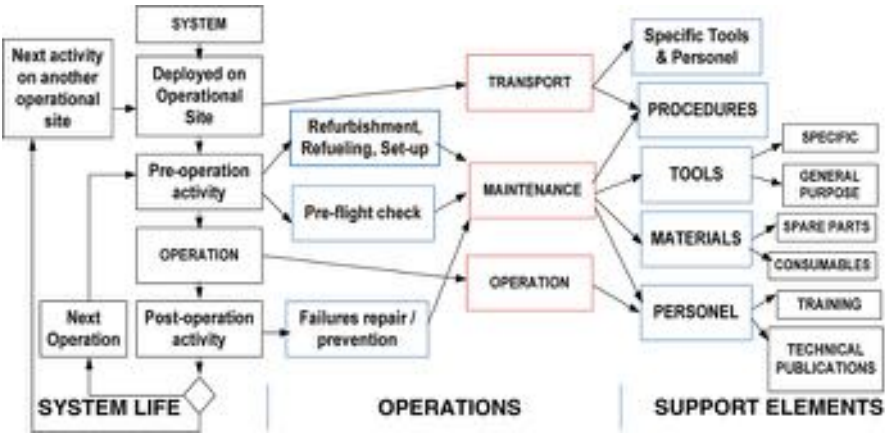


Fig. 107.4 UAS operational life and logistics

that is necessary to allow the product to perform continuously during its lifetime; Fig. 107.4 shows the key criteria for a UAS system to work correctly:

- (a) It must be moved to the place where it performs;
- (b) It must be prepared before starting working or, in particular, before starting a mission; the suitability to perform a mission must obviously be checked before the starting of the same mission.
- (c) It must be operated (or “piloted”) during its performance.
- (d) It must be checked after an expected period of performances to prevent a deterioration; in other words, possible failures are to be corrected or, better, prevented and addressed in advance. In this way, the product is ready for a new cycle of functioning after a necessary preparation.

The result of the support logistic includes operational system deployment in the intended environment, planning of efficacy and maintenance, and all else necessary for operation: staff (for guiding and maintenance), tools (particular for the system and of “general purpose” or “standard”) and equipment (consumable and spare parts).

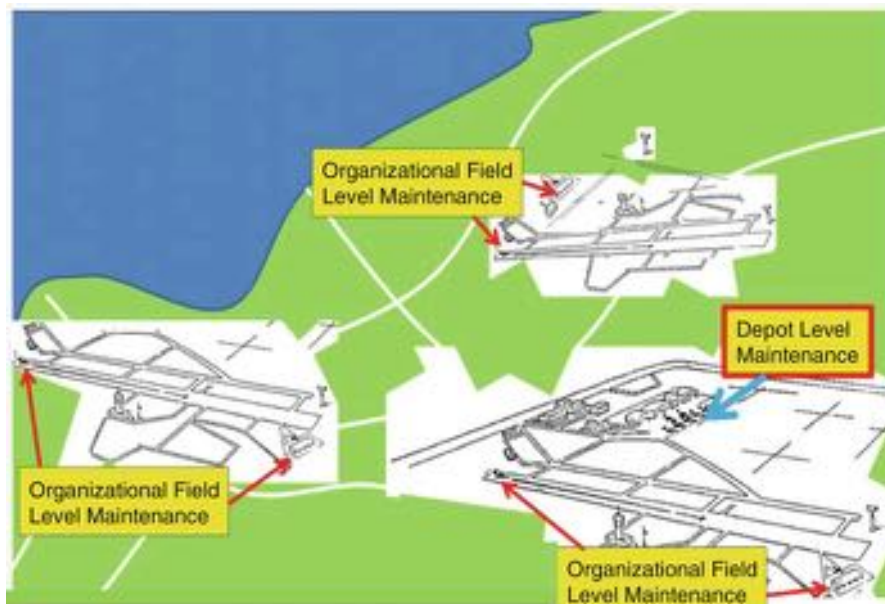


Fig. 107.5 Multi-bases operational and logistic scenario

Figure 107.4 highlights other problems, for instance, that operations as well as maintenance staff must be properly trained and supported by suitable technical handbooks; it is also clear, even if not shown in the figure, that staff, materials and equipment must be kept in dedicated facilities and that the system must be managed and information (internal and external) must be exchanged with a special computerized system.

Next, Fig. 107.5 shows how the operational and logistic scenario can get complicated if the UAS system is (as is very probable) deployed in a vast region, with more than one operation base.

Indeed, this fact offers the opportunity of optimizing the logistic processes, in particular maintainability, by diversifying the typology of activities on the various bases.

This requires organization of the whole maintenance process on many levels, depending on the difficulty of operations. In Fig. 107.5, for example, a possible subdivision of maintenance operation of a UAV fleet on two levels is shown: the “Organizational Field-Level Maintenance” scheduled for all the bases and the “Depot-Level Maintenance” scheduled only for a lower number of bases (in Fig. 107.5 the example of a single base).

In particular, whereas the more general and frequent activities (i.e., refurbishment and, generically, flight setting activities, as well as before and after flight controls) are carried out in all the bases (with reference to Fig. 107.5), which are sites of organizational field-level maintenance, the more complex, and usually less

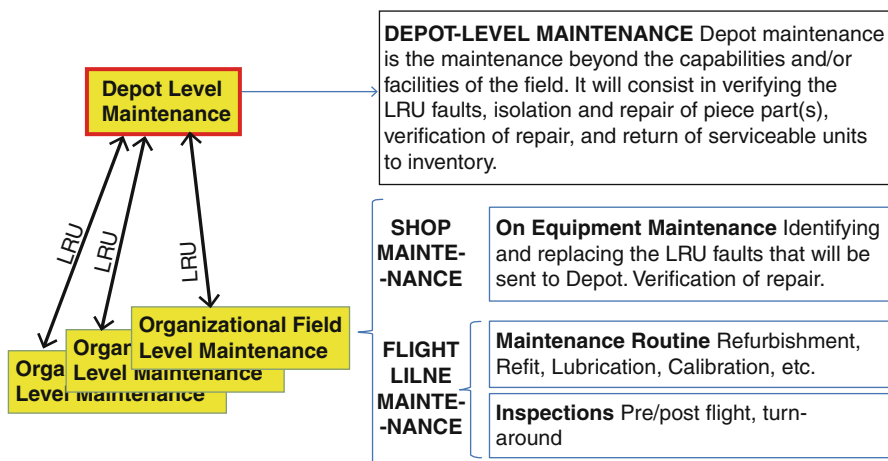


Fig. 107.6 Maintenance levels

frequent operations are carried out only in a lower number of bases (one only in Fig. 107.5).

This fact makes basically uneconomical to deploy at all bases the resources (staff, tools and facilities) for specialized operations, as these resources would be underutilized.

In particular, as to maintenance, in “organization field-level maintenance” the aircraft’s broken units LRU are replaced with other correctly working units; in this way the following advantages are obtained:

- The aircraft downtime is strongly reduced, with a quick replacement of the broken LRU with a working one; the operation is carried out by a staff with generic training that can easily replace any kind of LRU, while the indication of which part of the LRU is out of order is usually given by an automatic diagnostic system.
- The operation of repairing the unloaded LRU, which are generally specialistic and complex activities, are delegated to depot-level maintenance, where specialized resources are conveniently utilized as they receive LRUs from all the bases, whereas if they are included at all the bases they will be utilized in a reduced way.

The aforesaid advantages that derive from organizing maintenance on more than one level (as the scheme in Fig. 107.6 shows) are considerable (so that LRU costs and time of transports to and from the depot maintenance level are minimized), but it is clear that all these considerations are influenced by the fact that the system project has already foreseen the level maintenance organization, in particular by providing the above mentioned automatic diagnostics, and that the project is modular, that is with LRUs ready to be easily removed and reassembled.

It is clear then that the optimization of maintenance and, consequently, the logistic support system has as main assumption, which cannot be overlooked, the

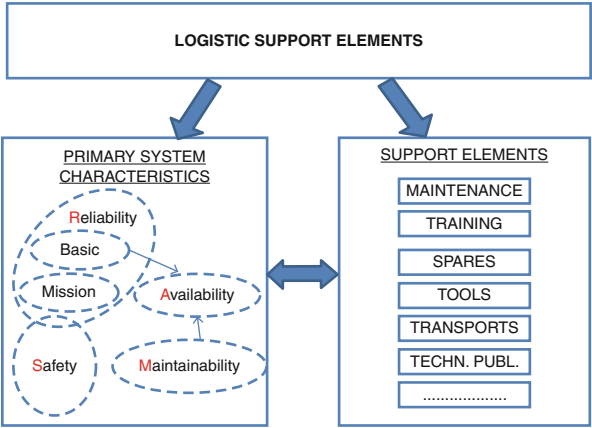


Fig. 107.7 Logistic support components: primary systems characteristics and support system elements

primary system project and its logistic support as carried out at the same time from the ILS point of view.

After such a short characterization of complex system logistic and an operative scenario depicting a possible system for a UAS fleet, Fig. 107.7 shows again the support elements already seen, along with primary system characteristics like reliability and maintainability.

A very preliminary peculiarity of these two characteristics could be considering reliability as indication of how the system tends towards failure or not, and maintainability as an indication of increased or decreased ease to be placed under maintenance.

It is evident that such characteristics are basic for the definition of logistic support, and this explains why, in Fig. 107.7, they are indicated as “elements” of logistic support; in the same figure it is said in advance that reliability and maintainability are characteristics strictly linked to others, like availability, mission reliability, and safety.

These extremely important characteristics are to be treated together, as is seen in the following section. In the same Fig. 107.7 the explanation of the acronym R.A.M.S. is shown.

107.3 UAS Support Systems Elements

R.A.M.S. refers to “Reliability, Availability, Maintainability, and Safety.” Considering these topics together is widely accepted due to their strict relationships, as for instance:

- The stochastic nature of safety binds it with reliability, as the probability not to suffer any failure in any part of the system; so on the basis of these

probabilities (the reliabilities of each component) and how the components are interacting, safety, mission reliability, and basic reliability are derived. Safety is understood as the probability of having no failure (or in many cases “groups” of concurrent failures of various components) to prevent the fulfilment of critical functionalities; mission reliability means the probability of having no failures (or groups of concurrent failures) to prevent the fulfilment of functions needed for the mission success. Finally, basic reliability or simply reliability includes all the possible failures occurring on the system.

- Basic reliability obviously affects the entity of maintenance needed to delete the effects of failures as they occur, so the time required for the maintenance itself; maintainability also, being the feature of a system to receive maintenance actions more or less easily, is a concurrent issue in defining such time. It seems clear the binding between reliability, maintainability and availability, being this last a feature depending on the time spent under maintenance by the system.

Having demonstrated that reliability, availability, maintainability and safety are linked together, there is still “unavailability,” by which we mean that availability does not depend only on maintenance, that is, the system is not available not only when it is under maintenance, but also when unavailability is imposed by operational needs, in particular due to transports in case of clearing in other sites. In the field of maintenance, availability is also dependent not only on maintainability but also on the maintenance organization, on its sizing, on the skills of maintenance personnel, on technical documentation, on the tools used, on the amounts of spare parts available and, where needed, again on the transport activities.

This leads to the concept of logistic support systems and of the multiple elements that are part of it; clearly, the logistic support system is a particular to the system and it is affected greatly by its features of reliability and maintainability, and even, for the needs of “transportability,” on the division into different modules (to be easily integrated or separated) that the system has to include. A basic step is to understand that all the features imposed on the system by design, as already seen for reliability, and all the maintenance operations and logistic support systems, such as technical publications and the skills of maintenance personnel, are all features that affect safety.

Figure 107.8 shows synthetically the concepts explained.

Discussing the relevance of the above described features, it is worth notice how safety, apart from obvious ethical reasons, is the basis of certification and is thus a needed condition within any system, particularly in aeronautics, while reliability, maintainability, availability and logistics affect operational capability and, heavily, costs (purchase and operational costs) of a system, so affecting its effectiveness and efficiency.

This topic will be investigated at a deeper level in the following, but for now we point out that the competitiveness needed to develop successful aerospace systems today clearly show how strong an attention has to be placed on it.

The state of the art in the field of R.A.M.S. and logistics is defined by some consolidated results reached in previous times (Chiesa 2007); in particular, the following points can be reiterated:

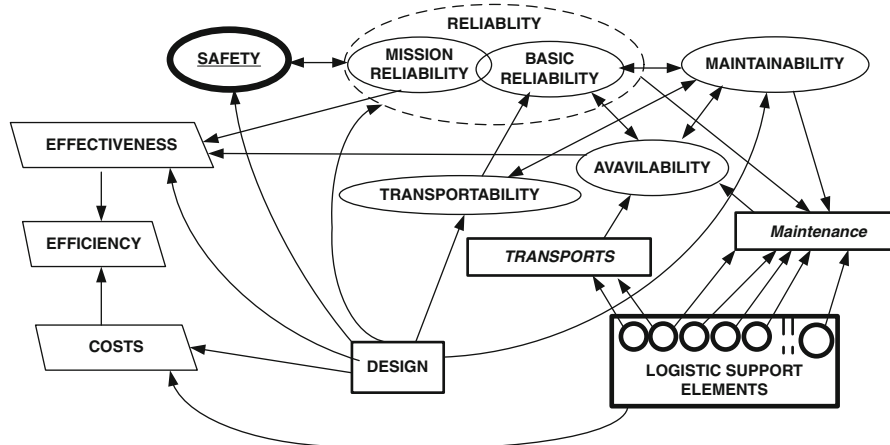


Fig. 107.8 Relationships between R.A.M.S., logistic support elements, design and other related issues

- Safety, reliability and maintainability are design features, as their respective values, in a given product, are derived by the design of the product itself;
- The logistic support system has to be designed not subsequently but at the same time as the primary system design, in such a way that the two systems will be developed with no problems of compatibility between each other, and in such a way to achieve the highest level of efficiency (this is the concept of Integrated Logistic Support – ILS (ILS Army Regulation 2007))
- Particularly with reference to maintainability (MIL-STD 471 and 472) it is worth notice that in this field a particular development was obtained with the techniques of automatic diagnostics (Johnson 2005, Keller et al. 2001, Xiaoyang 2012), with their own processes in both HW and SW; as a direct consequence, the feature of “testability” has increased in relevance, encompassing both the testability the easiness of the System to be tested (MIL-STD 2165 and MIL-HDBK 1814); the relevance of testability is such that sometimes, instead of R.A.M.S., R.A.M.S.T. acronyms can be found, in order to show the importance of testability, which is considered a self-grown discipline and not only a branch of maintainability.

107.3.1 R.A.M.S.

Returning to a deeper level of investigation it's useful before showing how R.A.M.S. disciplines can be developed in design phases to notice how all of these disciplines are “crossing characteristics,” features for which nearly all components of the system concur to define their own level. A direct consequence of this is that in the design team of a system only specialist designers of all part components are able to, and have to, design the part components under their responsibility in order to

satisfy R.A.M.S. requirements, while at the same time they have to satisfy specific “crossing characteristic” requirements with respect to other parts and the system as a whole, such as overall weight or overall cost; the activity of specialist designers has therefore to be addressed, driven, and supported by discipline specialist that, in order:

- Provide specialist designers the requirements, for their own part components, in the various R.A.M.S. disciplines;
- Control continuity of designs of the various parts are developed with respect of the above mentioned requirements;
- Provide specialists supports for what concerns the aspects of various R.A.M.S. disciplines, in particular for what is related to the improvements whose needs was put in evidence by the control activity above mentioned, but also for planning and execute demonstrations of R.A.M.S. level reached at the end of design development.

This organizational strategy is shown in the following Fig. 107.9.

In order to complete the overview of “design to R.A.M.S.” it can be useful to recall some of the techniques currently in use today; in particular:

- In the field of prediction techniques it’s worth remembering MIL-HDBK 217 “Reliability Prediction for Electronic Components” and several data bases (among which is RAC - Reliability Analysis Center) for mechanical design to reliability (Dudley 1997).

Other techniques, even for the other R.A.M.S. disciplines beyond reliability, are available in the exhaustive set of MIL rules (MIL-HDBK 217F N2), at least for the activities concerning design development. On the contrary, for what concerns the early design phases, in which taking care of R.A.M.S. aspects is not easy due to the still poor definition of the design, but yet extremely useful since the features achieved by the future product through the conceptual design phase will tend to be maintained, some methods, developed mainly in academic environment (Chiesa 2007), are available.

- Analysis Techniques: in the reliability field, a great relevancy is held by F.M.E.A. (failure modes and effect analysis, MIL-STD-1629A), used in various shapes, starting from the original definition MIL, later on relevantly readapted in several cases. For what concerns safety, the most known technique is probably F.T.A. (faults tree analysis), often mentioned together with F.M.E.A., as complementary techniques (Fig. 107.10), one being based on a bottom-up approach, the other being top-down, while, in particular if there is an aim to catch the effects on Safety of the installations of the systems on the cells, the “Qualitative Safety Techniques” are worth to be mentioned, of which a clear example is given by the “Safety Specification” of ESA (European Space Agency) and yet other methodologies determined by the academic world (Chiesa 2007).

As a further consideration on safety, it has to be kept in mind that, for unmanned air vehicles, the fact of not carrying crew or passengers onboard leads to a major concern not only on that failures that may cause a crash, but above all care must be taken to avoid damages for people on the ground. In particular, the concept of ELS (expected level of safety) was elaborated, such as:

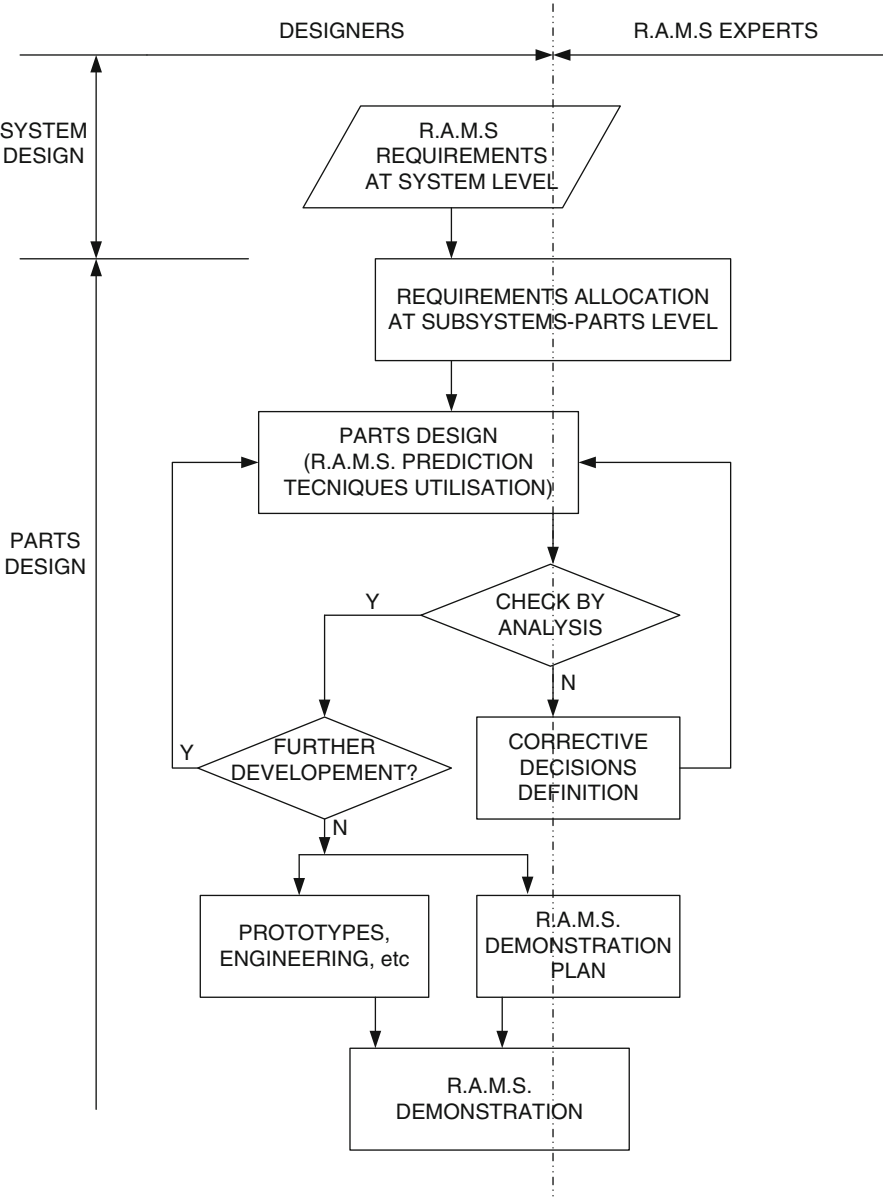
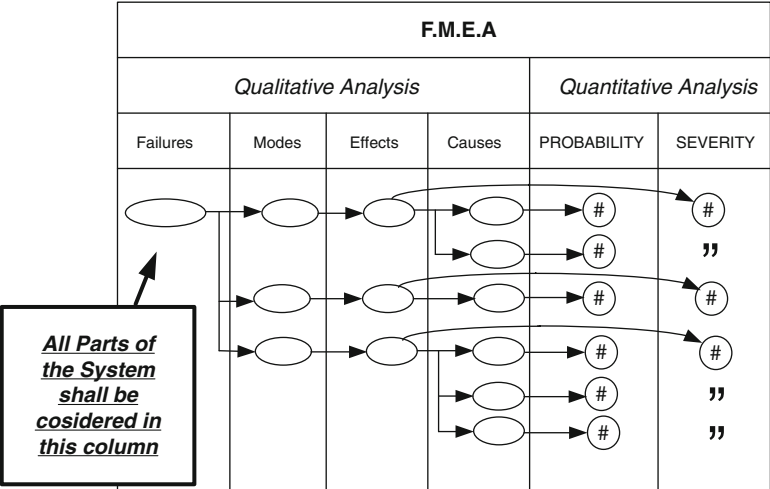


Fig. 107.9 Task distribution in R.A.M.S. oriented design



All Parts (Componets) are considered, with their various possible modes; for all the effects on System are analyzed
→ **BOTTOM - UP APPROACH**

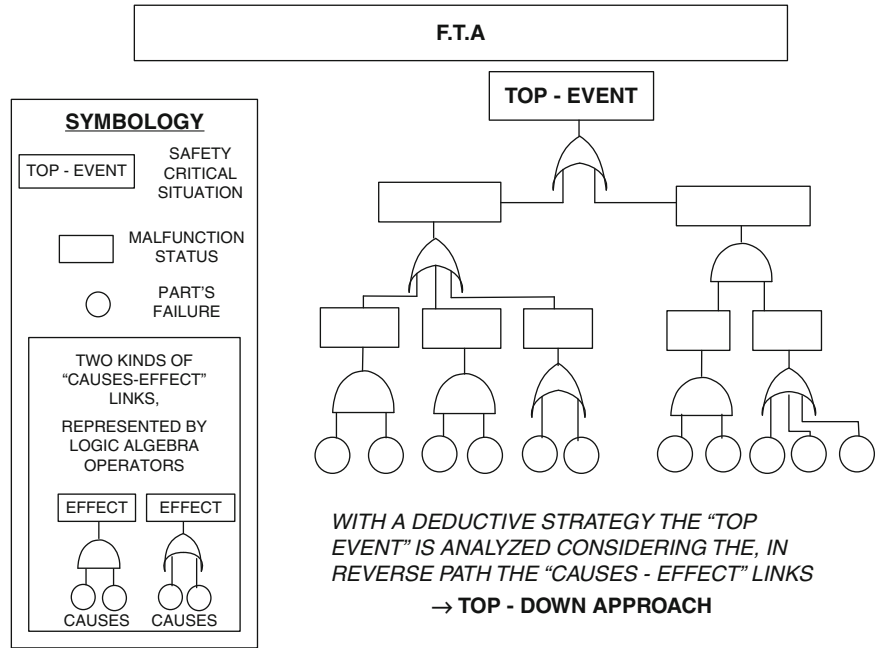


Fig. 107.10 F.M.E.A. and F.T.A. approaches

$$ELS = \frac{A_{\text{exp}} \rho P_{\text{pen}} (1 - P_{\text{mit}})}{MTBF},$$

where,

- $(1/MTBF)$ is the failure rate of safety, so the estimated rate of “crash”
- A_{exp} is the territorial area damaged by the crash
- ρ is the population density of the area on which the UAV is operating
- P_{pen} is the probability for people to be hit by debris, even if they are protected by houses, buildings or vehicles in which they eventually find themselves in the timeframe when the crash occurs.
- P_{mit} is the probability that crash effects are decreased by systems of safe flight termination (i.e., ballistic parachutes, autonomous guidance systems on uninhabited places, and so on)

This point of view, particularly innovative, requires a great attention.

A brief consideration of the following phases of design leads us to consider how in the phases of construction of a product (production, but also prototype realization) R.A.M.S. features have to be taken into account; before all else, it has to be recognized that, as already mentioned before, the design is assumed to be oriented to R.A.M.S. features, for which the most critical issue is that, through production, there should not be an unacceptable decrease in the values achieved by design. This is particularly relevant for reliability, as design choices for maintainability (such as, accessibility, systems of automatic diagnostics, modularity, use of connections “quick connect/disconnect”) and partially design choices for safety (in particular for redundancies, and installation choices, in spite of already mentioned qualitative safety) result in design “macro-choices” that cannot be overcome by construction; on the contrary, reliability of equipments and installed devices (with clear effects on safety, mission reliability, and basic reliability) can be reduced with respect to design levels for integration mistakes, rough handling, etc. There is a clear need for attention to avoid, or at least decrease, this reduction of reliability caused by construction process; a very useful tool for this is the “process FMEA,” that is FMEA applied no more on all the “part components” of the product but on all the consequent operations in the processes of construction and production.

107.3.2 Logistics

As to the logistics aspects, they are focused on developing, as already mentioned, in an as far as possible way, designs integrated with the system development (I.L.S. concept Jones 2006, see Fig. 107.11), that is called “System Support.”

The system support is constituted by the integration of the various elements concurring to the system support, that is:

- Support and maintenance tools and instruments
- Technical publications
- Personnel training
- Spare components and consumables

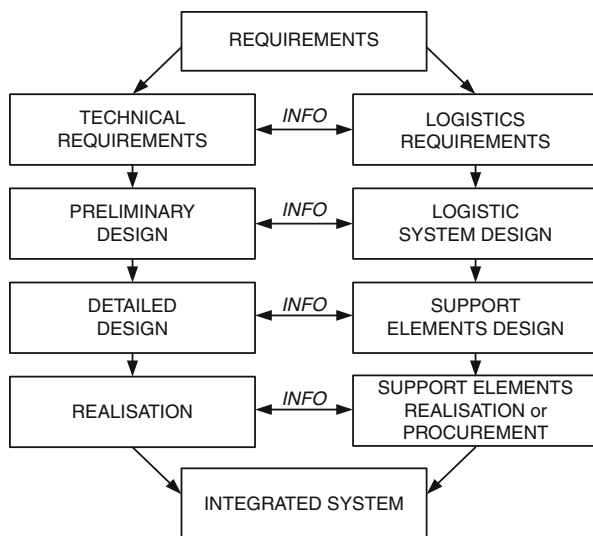


Fig. 107.11 From requirements to integrated system, meaning product and logistic support system

- Facilities
- Management procedures and SW
- Transportation

This list, purely indicative and not exhaustive, helps understanding as system support is constituted by various and heterogeneous elements, including elements such as “personnel,” “HW,” “SW,” in which there is a concurrence of human factors, technical and economic management aspects, with strict interfaces with the main system, in particular for the levels of reliability and maintainability and the operational needs; there is a strong evidence of the amount of resources included and so of costs, usually in the same order of magnitude of the main system.

Another important and challenging task is to define and refine the sizing of the system support and how it has to be performed. The currently common strategy puts on the first place the definition of a maintenance program deriving the requirements of the system support (in both qualitative and quantitative terms) from maintenance tasks that are defined (as tasks and frequency) by just defining the maintenance program: this is a list of the various maintenance tasks that have to be foreseen to prevent anomalies and failures (preventive maintenance) or to repair occurred failures (corrective maintenance). It's clear that for tasks of preventive maintenance, which can be either “routine tasks” such as refuelling, cleaning, lubrications, controls, servicing, etc., or scheduled replacements of parts or components, or periodical controls on parts, there is not only a need to define the nature of the tasks, but also to define the frequency and a time schedule.

An obvious consequence is that the definition of the maintenance program becomes a very important task to be executed with great accuracy and in the most favourable timeframe; these two aspects are examined in detail in the following.

Currently, the definition of the maintenance program can be performed in the best way possible by using the RCM approach (reliability-centered maintenance) (MIL-STD-2173) with the use of techniques such as, to recall the most well-known, MSG-3 standard. This methodology was developed, in the last decade, by a critique of the traditional approach, which essentially considered the various parts-components of the main system, defining for them the maintenance tasks estimated as the most useful. The key flaws of this strategy were:

- A certain difficulty in imagining the failure modes of the various parts, in particular those due to interactions of more than one parts, to define the most suitable maintenance action, preventive or corrective;
- A complete separation from the design activities (often already completed previously, that is opposite with respect to ILS approach)
- The subjectivity of the decisions taken

In summary, the RCM approach prescribes the following guidelines:

- Define a maintenance program based on FMEA statements; for each one of these an eventual maintenance task has to be decided; please note that in this way the first two criticalities of the traditional approach are deleted, avoiding the separation from the design and using FMEA, thus providing an completed activity where problems, such as failures due to multiple part interactions, have already been solved (moreover, they have been solved by whomever seems to be the most appropriate person to do so);

Use an appropriate decision logic to decide, for the various FMEA points, if a maintenance task has anticipated; this decision logic practically eliminates the threat of subjectivity; it is constituted by a series of decisional diagrams that force the writer of the maintenance programs to ask him/herself the correct question in the correct order, avoiding the mistake of not considering one or more solutions eventually possible; it has to be noticed that the decision logic is organized in a way to guarantee first of all, safety, and then to minimize costs. A further contribution brought by MSG3 is to provide a very good and exhaustive “traceability” of how and why the various decisions, leading to the maintenance program, have been taken. Finally, it has to be noticed that the decision logic can lead, for each point of FMEA, to foresee servicing or scheduled checks (to decide if a part is to be removed or not) or scheduled substitution of parts or even decide to wait being ready for, eventual, corrective maintenance; it has to be noticed, on this point, that, in case a failure involves problems of safety and no preventive maintenance tasks result possible, the decision logic puts as the only alternative a need to redesign.

In Fig. 107.12, the RCM methodology is summarized; in particular, the figure illustrates how the RCM inputs are the whole failure modes reported by FMEA. For each of them the decisional logic is applied; the figure shows that the decision logic is spread on two levels, the first of which classifies the failure mode from the point of view of criticality on the system, and the second responds with different strategies, depending on the result of the first level, and chooses the more convenient

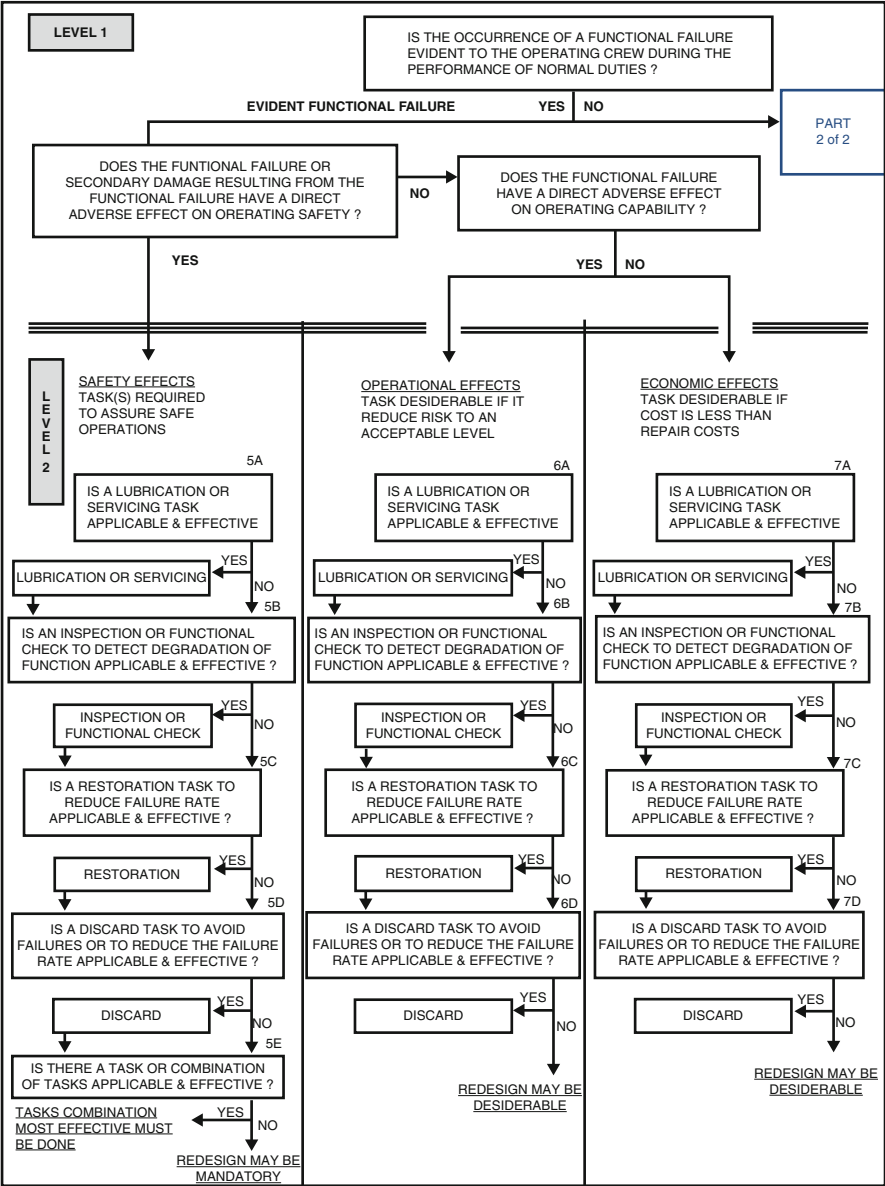


Fig. 107.13 Example of a system logic diagram

led to the sequence (apart from obviously possible iterations and feedback on the previous activities) shown in Fig. 107.14, fully agreeing with ILS approach.

In order to complete the overview on the currently available instruments, the already mentioned Fig. 107.14 points out three more, the L.S.A. – Logistic Support

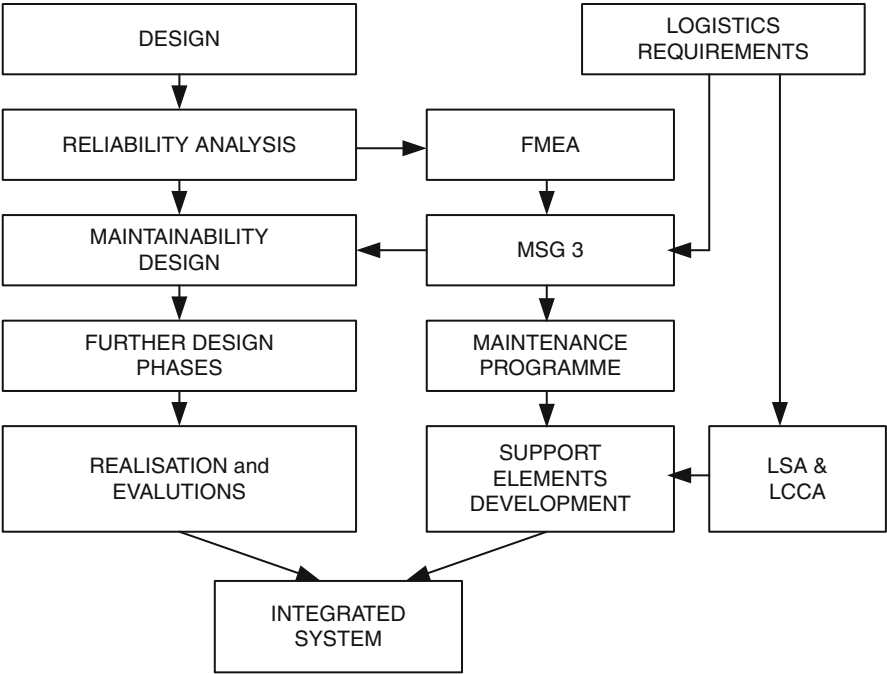


Fig. 107.14 Activities in integrated logistic system definition

Analysis (MIL-STD-1388-1A), the L.O.R.A. – Level Of Repair Analysis (MIL – STD 1390D) and L.C.C.A. – Life Cycle Cost Analysis (Blanchard 1978).

The first one, the LSA, is in summary a guideline for the development of ILS, pointing out the several data that have to be produced and put in evidence in the so called LSAR (LSA reports). The related MIL standard or other equivalent regulations list all activities to be done, during an integrated logistic system development in order to avoid any weakness in logistics definition. To give idea of high level of detail that LSA requires, in Fig. 107.15 the logistic requirement considered at every phase of the development are shown.

The second tool indicated in Fig. 107.14 is, on the contrary, a quite specialised tool, very useful in definition of procedures describing how a maintenance task has to be driven up, if the task involves the identification of a failed part; in such a case, the LORA, as Fig. 107.16 (MTain Inc. 2011) shows, indicates if the part has to be discard or has to be repaired, and, in this case, provides a definition of the maintenance level in which the repair has to be performed. As the figure shows, the LORA forces the maintenance planner to make the economically more convenient choices.

Finally, the third tool indicated in Fig. 107.14 is the LCCA (life cycle cost analysis) that plays the role of optimisation driver for the whole integrated logistic system development process.

| Logistic Support Requirements | | |
|---|--|--|
| <div>Test and Support Equipment</div> <div><ul style="list-style-type: none">• Type and Quantity of Equipment• Location of Equipment• Utilization of Equipment</div> | <div>Spare / Repair Parts (Supply Support)</div> <div><ul style="list-style-type: none">• Type and Quantity of Spare/Repair Parts• Location of Spares• Inventory Requirements• Provisioning Factors</div> | <div>Personnel and Training</div> <div><ul style="list-style-type: none">• Personnel Skill Levels• Quantity of Personnel• Assignment (Location)• Training Programs• Training Aids, DataEquipment and Facilities</div> |
| <div>Transportation and Handling</div> <div><ul style="list-style-type: none">• Packing and Shipping Requirements• Environmental Requirements• Handling Equipment</div> | <div>Operational and Maintenance Facilities</div> <div><ul style="list-style-type: none">• Space Requirements• Environmental RequirementsUtilities• Warehousing• Capital Equipment</div> | <div>Technical Data</div> <div><ul style="list-style-type: none">• Operating and Maintenance Procedures• Installation and Testing Procedures• Modification Instructions• Drawings/Microfilm/Lists</div> |

Fig. 107.15 Logistic support requirements in logistic support analysis

It is logical to think about life cycle cost concept, with the famous allegory of the “iceberg,” illustrated in Fig. 107.17, that defines all the cost components to be considered during the life cycle of the system, and it is extremely pertinent to other complex systems and with long life cycles. The LCC concept is useful in order to establish a standard, clearly defined and accepted definition of the very many costs items (Blanchard 2003). The LCCA (life cycle cost analysis), operated in a standard way by evaluating LCC for the various alternatives that grow through the development of the ILS, becomes a very good instrument for decision making and, as already said and as illustrated in Fig. 107.18, a precious optimisation tool.

107.3.3 Transport and Transportability

In a preview of the development of future advanced systems, in particular ones comprehensive of UAS, it appears mandatory to pursue a minimization of logistic footprints and of costs in general. The obvious choices use the best that state of art can provide in fields of R.A.M.S. and logistics, but there is another topic on which studies and researches have to be addressed; the transports.

Considering Fig. 107.4, it is clear that, for UAS, deployment in different sites can happen many times during operational life. The scenario (the related relevance has been discussed) can be considered as geographically variable if the UAS has to be deployed, when requested, far from its usual operational site; in this case, it is clear that the logistic support, or a part of it (for example the first level of maintenance), has to be moved; further considerations about this aspect will be presented in the following point. As the transport will be, probably, not specific for UAS, the compatibility with the available transport system has to be pursued

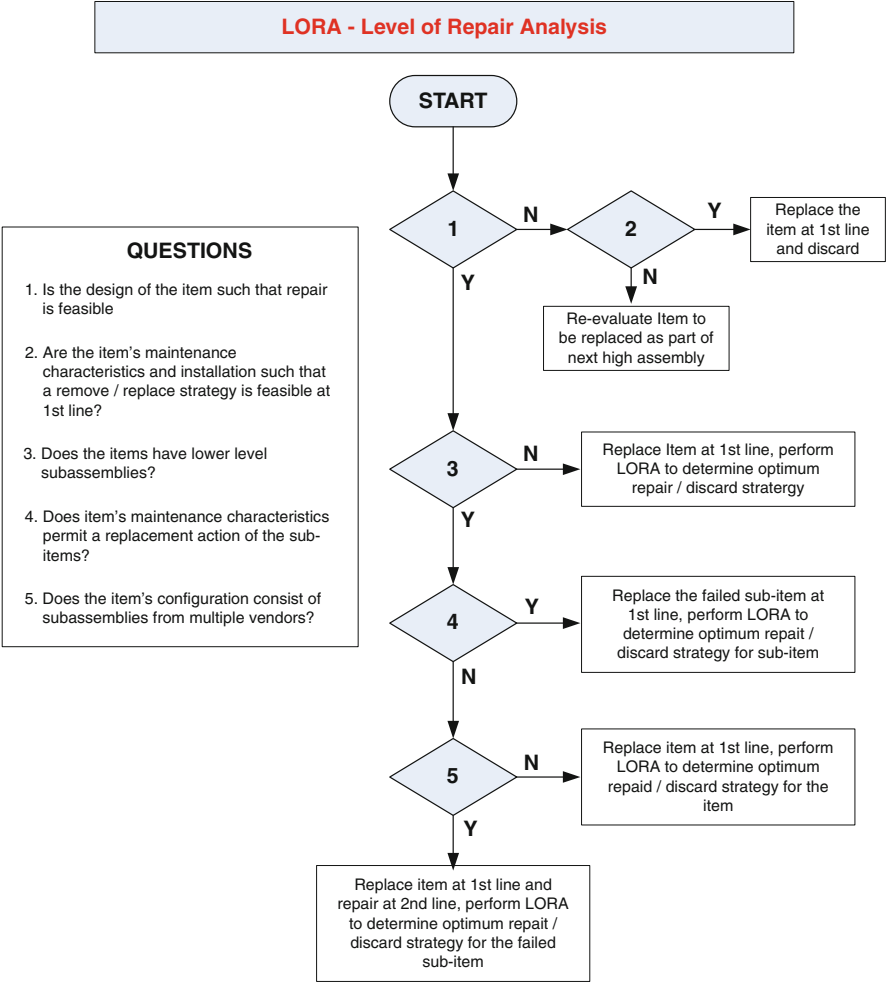


Fig. 107.16 LORA – level of repair analysis

during logistic support elements definition. Similar considerations can be done for moving the GCS (ground control station) of the UAS.

The problem can be interesting when speaking of the aerial vehicles, i.e., the UAVs; in this case, an interesting trade-off analysis to be performed is whether it is better for the UAVs themselves to reach the new operational site in flight, versus being transported by transport system, as happens for GCS and support elements. Please note that the first alternative is usual for manned aircraft, but the second could be better for UAVs, but only if they are “designed to be transportable.”

But what does “design to be transportable” mean? Concisely, it means that the UAV can be easily and quickly disassembled and reassembled, and that the various

Fig. 107.17 LCC – life cycle cost allegoric representation

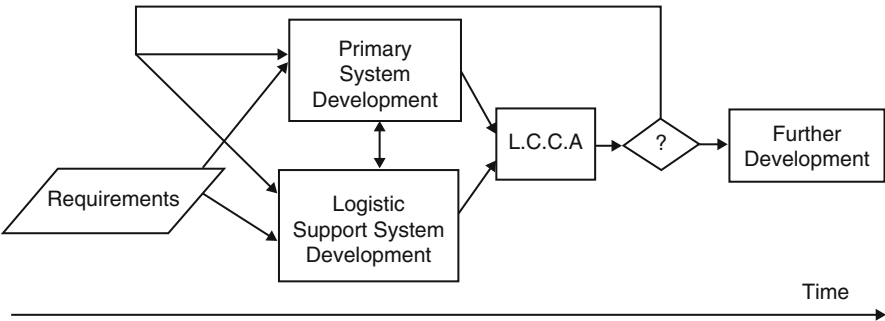
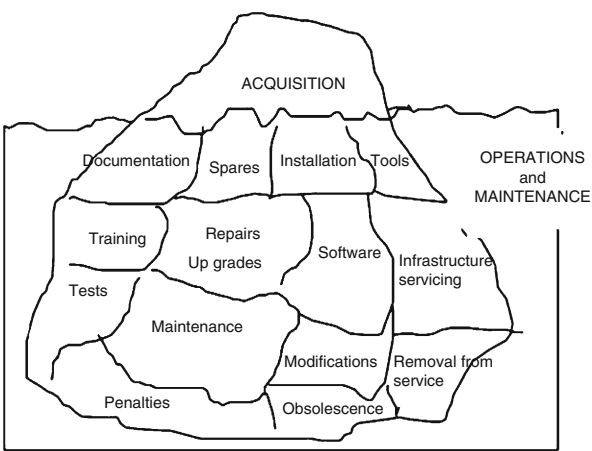


Fig. 107.18 L.C.C.A. – life cycle cost analysis as optimisation tool of ILS development process

parts in which the aircraft is disassembled can be easily stored, in particular in containers with standard dimensions (compatibles with the transportation systems) and of very few kinds (better if the same containers utilised to transport the system components other than UAVs).

So the optimization of the break-down of the vehicles becomes of primary relevance, and has to be defined in the early phases of design, taking into account that, apart from the fact that it concerns many technical aspects, such as layout of assembly line and thus the entire process of manufacturing, and also affects maintainability and logistics, for UAS (much more so than for manned aircraft) there is the question of being stored in containers for transportation by means of ground or sea path.

Figure 107.19 shows the concept of vehicle breakdown. It is important to understand that the vehicle breakdown has to be defined early in design, due to the following reasons:

- The whole project of development and the following detailed design are highly affected by the kind of decomposition, in terms of design and structural sizing,

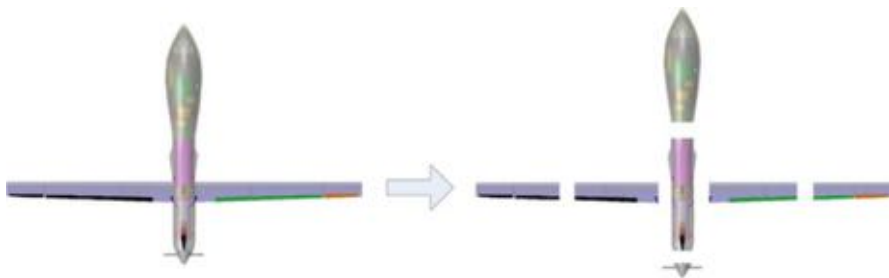


Fig. 107.19 Concept of aerial vehicle break-down

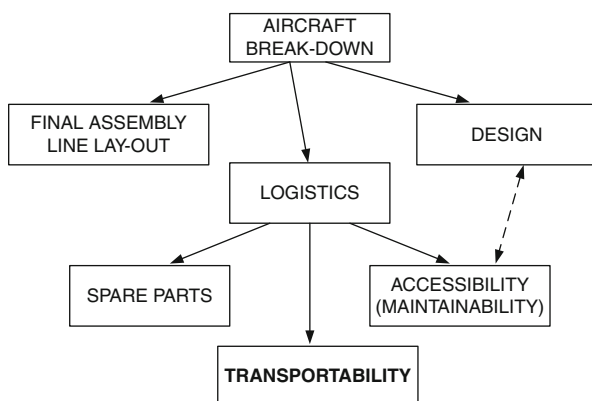


Fig. 107.20 Transportability

but also of design of systems that would have to be split in several “segments” into which the aircraft will be divided and, as it will be discussed later, these choices should be studied with care.

- The breakdown will be functional, and so it will affect some other activities and features of the vehicle that will affect on their own side effectiveness and costs of the system; all of these considerations are displayed in Fig. 107.20;

In this figure it can be noticed how the division of the aircraft has impacts on logistics for several aspects, among which is “Transportability,” that is, the attitude of a system to be transported by means of air, sea, or ground paths. It seems evident that this feature, generally not important since it is relevant only for the assembly tasks (transport of segments of vehicle from the production site to the final assembly site), or carrying segments to be used as spare parts for great maintenance or repair campaigns, it nevertheless takes on a great relevance for UAS. In fact, transport not by air but by means of ground paths are quite usual for unmanned vehicles, both for operational reasons linked to transfer flights, and because, differently from “manned” vehicles the aircraft requires a ground control station which may also be in the process of being moved itself.

Table 107.1 Load factors during transportation

| Medium/mode | Longitudinal load factors, g | Lateral load factors, g | Vertical load factors, g |
|-----------------------|------------------------------|--------------------------|--------------------------|
| Water | $\pm 0,5$ | $\pm 2,5$ | $+2,5$ |
| Air | $\pm 3,0$ | $\pm 1,5$ | $\pm 3,0$ |
| Ground | | | |
| Truck | $\pm 3,5$ | $\pm 2,0$ | $+6,0$ |
| Rail (bumping shocks) | $\pm 6,0$ to $\pm 30,0$ | $\pm 2,0$ to $\pm 5,0$ | $+4,0$ to $+15$ |
| Rail (rolling) | $\pm 0,25$ to $\pm 3,0$ | $\pm 0,25$ to $\pm 0,75$ | $+0,2$ to $+3,0$ |
| Slow-moving dolly | $\pm 1,0$ | $\pm 0,75$ | $+2,0$ |

This fact makes it necessary to consider, during the design of the vehicle, the sizes and shapes of the containers needed to contain the segments in which UAV has to be split, and to conclude that “designed to be transportable” requires an integrated study of the vehicle breakdown, and consideration of the features of the various transporters considered, both their transport capability (weights and volumes) and the environment in which the elements (UAV and GCS) will be hosted; see the following Table 107.1 for example:

From this short consideration, it is clear that a global strategy to define the vehicle breakdown will have to be implemented in the design of future UAS. Such a strategy will have to foresee a strict integration of design activities involving the features shown in Fig. 107.20, in a way that all the needs will be satisfied as much as possible. Particularly interesting is the need that the division of the various systems that have to be distributed in the various segments has to be optimized by means of the choice of the positions on the vehicle (and so on the various segments) of the devices; among other criteria, the following ones can be noted:

- To minimize the number of lines (cables, pipes, etc.) to be interrupted when separating segments;
- The parts of a system on the same structural segment shall be as much as possible complete from a functional point of view (with the purpose to allow eventual testing operations on those system parts separately).

The whole transportability strategy has to be considered during the primary system development and in a collaborative way with it, exactly as already seen for other elements of the system support, and so the “transportability design” (a concise example is given by Fig. 107.21) will be a relevant aspect during the operative way of the system (see again Fig. 107.4).

107.3.4 UAS Logistic Peculiarities

Further logistic peculiarities of UAS derive from the coexistence of two different segments, one concerning flights, and one concerning the ground.

As Fig. 107.22 shows, the link between the flying segment and the ground one is a fundamental element that must always be present in every phase of the mission;

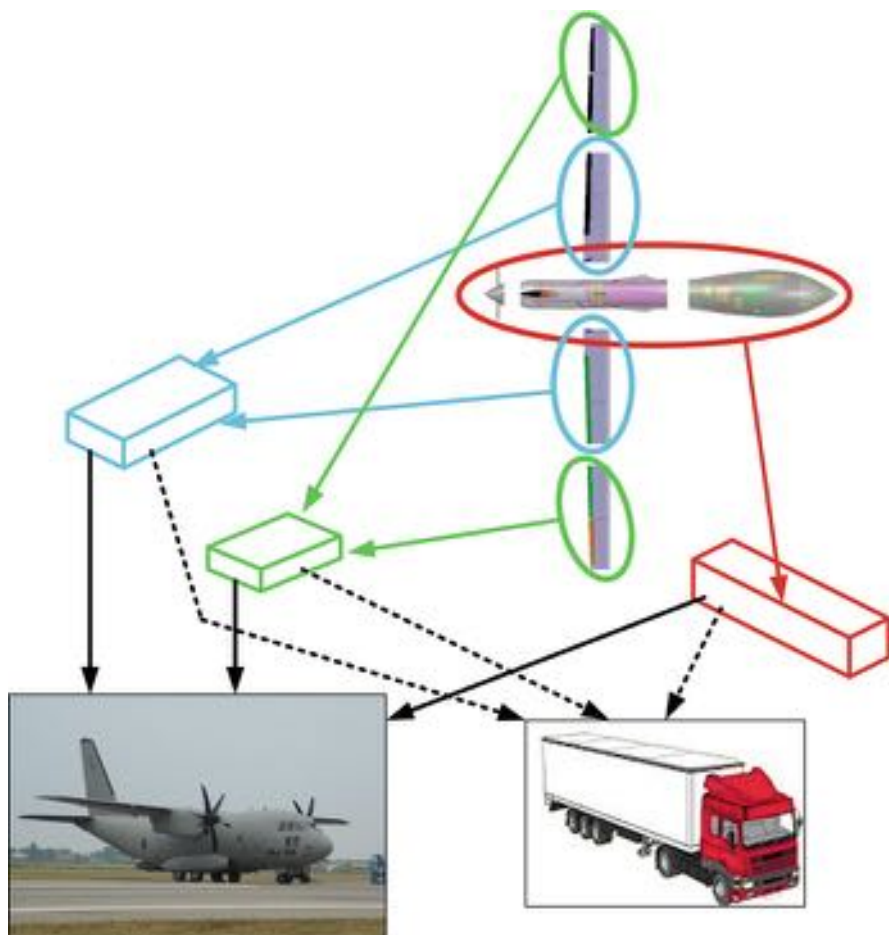


Fig. 107.21 Example of transportability design

even in the cases when the system has a high ability for operating autonomously, the possibility of setting the link between aircraft and the GCS must always be present.

In comparison to traditional aircraft, for which the loss of the link with other “systems of the systems” elements represents only a reduction of efficiency and/or efficacy, as such piloted aircraft have enough decisional autonomy to complete the mission or at least to return safely to the base, for UAS the loss of the link is considered a serious problem.

The more complex UAS systems (such as Global Hawk, Predator A and B) can carry further problems from the logistic point of view, when deployed far from the usual base. In fact they are made up from a ground segment (GCS, ground control system) that controls one or more UAV (normally 1 to 4). The GCS in its turn is

Fig. 107.22 Flight segment
– ground segment Link



made up from a LRE (launch and recovery element) station and a MCE (mission control element) station (Austin 2010).

The LRE station is utilized for the taking off and landing operations, and it is usually located in the same base where the flying segment operates. The communication established between LRE and UAV is a LOS (line of sight) link and it usually extends for a range of 200 nautical miles, in consideration of the Earth's bending and the UAV altitude. On the contrary, the MCE controls UAVs during all the remaining of the mission. Unlike the LRE station, the MCE is not bound to operate from the same base as the flying segment, so it can be positioned in any place on the Earth. Indeed, the link between the MCE station and the UAV (one or more) is a BLOS (beyond line of sight) link, by means of satellites or carried out by means of radio relay ground stations, and it is possible that the control station MCE is positioned in the main base of the UAS, while the UAV and LRE are deployed in the operation site, which could be located even in another continent (Fig. 107.23).

In fact, from the R.A.M.S. point of view, the link can be considered as an element that is in series with the other UAS segments. During some phases of the mission, such a link can be redounded, as can happen when the UAV communicates with an LOS link and is not performing landing or take-off phases, as the reliability block diagram in Fig. 107.24 shows. In more advanced UAS systems, the critical situation following the loss of the link can be compensated by setting up a waypoint in advance in the UAV mission computer, which must be reached autonomously when this failure occurs, and on which the UAV will stay until the link is restored.

Please note that the link reliability depends on components that can also be external to the UAS system and that are different according to the LOS or BLOS link.

In particular, the LOS link depends on the reliability of the UAV receiver/transmitter, and also on the control station, as well as the ether electromagnetic waves transmission that can be deviated by obstructions or jammed by atmospheric perturbations. In case of BLOS transmission, besides the above mentioned elements, satellites or the ground radio relay ground stations must be taken in consideration.

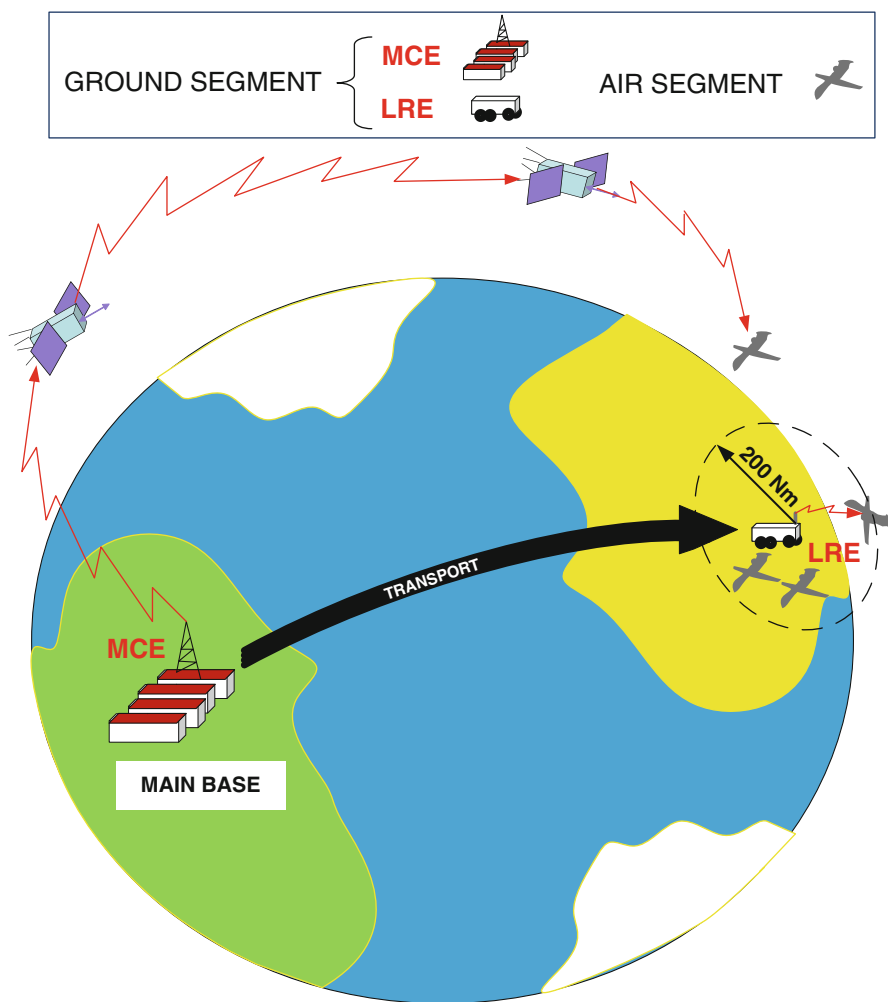


Fig. 107.23 Deployment in far settings

Other UAS peculiarities that produce effects on the RAMS characteristics are analyzed in the U.S. DoD “UAV Reliability Study” document (Unmanned Aerial Vehicle Reliability Study UAVs Design 2004). In consideration of the necessity of low acquisition cost for such systems, many failure causes derive from the choice of low reliability equipment, little redundancy of subsystems, and design inaccuracy. Other failure causes are more related to UAV design peculiarities, and are:

- Rain infiltration in the avionic bay: for simplicity of manufacture, in the absence of a cockpit, and in order to improve maintainability, UAVs generally have avionic equipment spread on a single stub plane between the nose of the fuselage

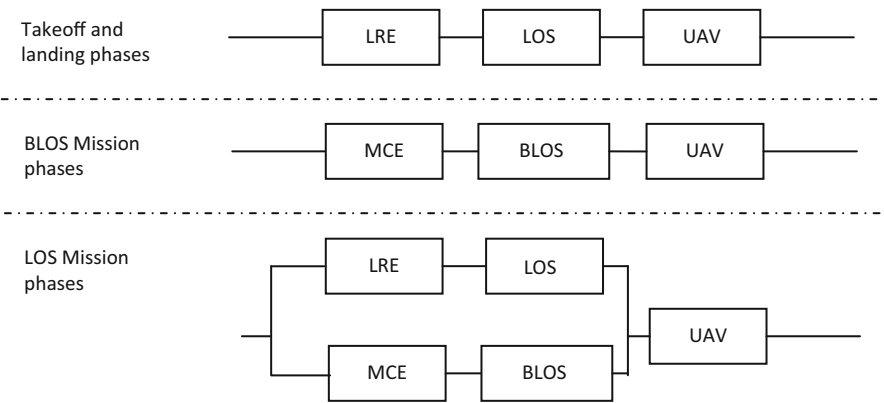


Fig. 107.24 Link between UAV and ground segment reliability block diagrams

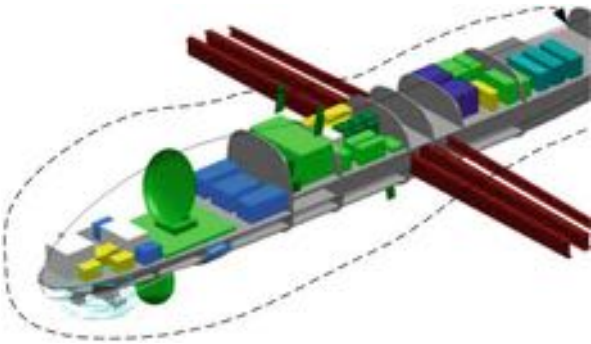


Fig. 107.25 UAV avionics installation

and the wing cross. All the fuselage over the stub plane is a single removable fairing that is not easily sealed because of its size (Fig. 107.25).

- Low wing load: existing surveillance UAVs are designed to have a high aerodynamic efficiency and operational altitude, and these facts cause a low wing load. This fact, in addition to a low cruise speed, leads to a high sensitivity to gust. The flight in turbulent air may cause a reduction of sensor accuracy, possible loss of satellite link and flight control loss when the turbulence is so high that the flight control system is not able to restore the flight level (i.e., safety conditions).

In a case such as the one shown in Fig. 107.23, the logistic support of the UAS system, at least the one called “organizational field level maintenance” would be physically separated (part with UAVs and LRE, part with MCE) with reduction of possible synergies that, on the contrary, would be possible in operating and maintaining all the parts of the system in the same base. The synergies that would be lost could be: the transport with the same means utilized for spare parts, the

utilization of the same maintenance staff, and the utilization of part of the system operating team for other activities.

In addition, the UAS system shows peculiarities also for the maintenance and system operation staff. The UAS system needs a high number of operators; for instance, for the REAPER system (4 UAV, LRE and MCE), about 170 people would be employed (Deptula 2009). In comparison to traditional aircraft, it is necessary to attend to the maintenance not only of the flying segment, but also of the GCS; which means considering the GCS complexity, as it is to operate in autonomous way, with autonomous power generation, and with multiple piloting and control (up to 4 UAVs contemporarily). Moreover, in view of the single pilot that drives the traditional aircraft, it is necessary to consider, for UAS systems, not only the person that operates it, but also the staff that operate the sensors and the satellite link, and the staff that plans the mission, sometimes in real time. In future times, the staff can be reduced by increasing the decision autonomy.

It is necessary also to take into account that missions can last up to 30h, and teams of operators taking shifts during the mission are needed.

107.4 Integrated Logistic Definition

R.A.M.S. analysis and logistics have an extremely relevant impact on any aeronautic systems, and this is particularly true while talking about UAS.

In general, for future UAS achieving a satisfactory level of safety, apart from obvious deontological reasons, has to be considered as a mandatory condition, since it is mandatory to obtain flight permission; this fact is particularly true for UAS devoted to operation in a civil field. In this ambit, a very interesting aspect could be to focus attention on the aforesaid concept of ELS, considered especially when the plan crashes on ground; in particular, research activities are desirable in order to accurately assess, in design phases, values such as A_{exp} , P_{pen} , and P_{mit} ; while for this last, the problem will be simply to estimate the probability of success of various kind of safe flight termination systems that will be surely equipped on the various UAVs; for the other two parameters, a methodology of estimation would have to be elaborated. An initial idea for a reliable assessment, not only statistical, for A_{exp} and P_{pen} is being able to find a correlation between those values and design choices, which would be set according to a strategy such as the one described in Fig. 107.26. Useful ideas and methodologies can be, at least partially, taken from methodologies for studies of “survivability” that are usual for flight vehicles. We also see some synergy with “safety zonal analysis,” previously already assessed.

Safety, in design phase, shows very strong interactions with reliability and, in general, with the large field known as “R.A.M.S. techniques”; the relative dependence between safety and logistics is also strong, and can be reviewed as logistics depend from design through reliability and maintainability.

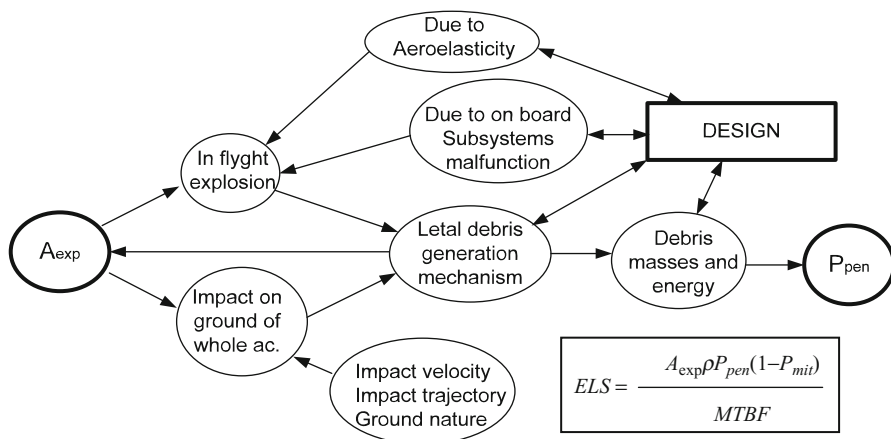


Fig. 107.26 Relationships between parameters of ELS

$$ELS = \frac{A_{exp} \rho P_{pen} (1 - P_{mit})}{MTBF}$$

The development of an integrated logistic system for UAS, in addition to the peculiarities examined for transportability, could be similar to the one used for other aircraft based systems; this is partly true, and it is demonstrated from the fact that methodologies like ILS, FMEA, RCM, LSA, LORA, LCCA are perfectly usable for UAS; a peculiarity of UAS is that the potential for use in generally not well defined and quickly changing places is extremely easy to waste. In addition, as UAS are still in a development phase, with a poor statistical data background, it has been observed that in many case traditional methodologies have stressed serious limits of applicability.

In such a situation, although the methodologies and the tools described before are to be considered generally valid, the uncertainties due to a lack of knowledge of quantity levels and/or provisional and nonstandard utilizing procedure, the possibility of having on hand a good simulation model of the system and of the operation and scenario, with in their evolution in time, is very important for evaluating possible alternatives, as well as for having a view of different results derived from different situations not predictable in advance. The model must obviously be precise enough to point out the differences of parameter values and/or qualitative differences (i.e., operative and logistic or, more generally setting procedures) that are not possible to be reliably evaluate in advance (Chiesa and Kostandinidis 1987; Chiesa and Olivero 1985).

A simulation tool must be able to simulate the operational environment mentioned above, and as a result forecast the right size for the considered UAV fleet and for their related logistic support (always in terms of operating bases and maintenance needs).

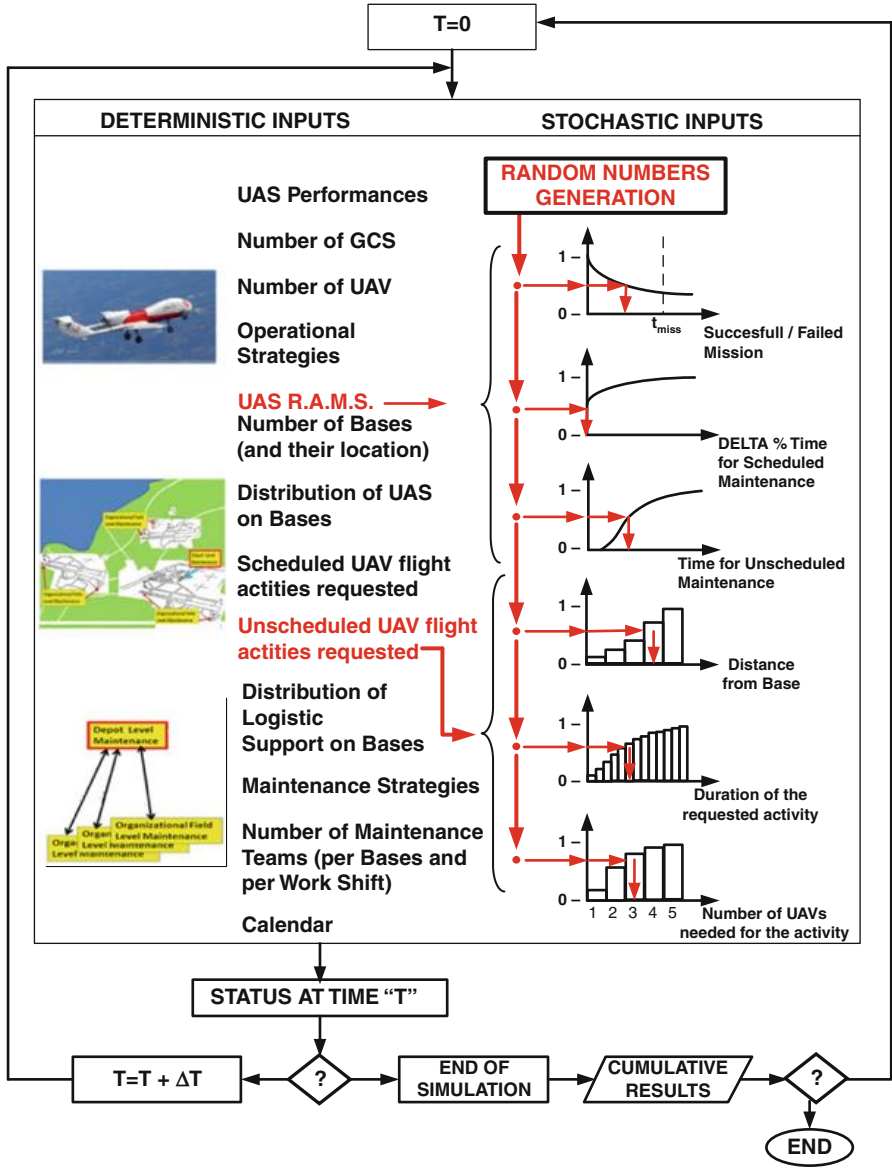


Fig. 107.27 Simulation model

Such a model must be able to consider as inputs all available data concerning the system, the scenario and the operations, summarized in Fig. 107.27. In this figure, it is shown that “inputs” can be of several kinds:

- *Deterministic Data*, that is data on which it is possible to assign an extremely precise numerical value, i.e.: number of UAV present, number of bases active

and number of GCS, number of maintenance teams present in each base for every work shift scheduled (shifts can be different day and night and between weekdays and holidays), calendar, with working days and holidays, for long enough period of time to include a planned operational cycle or a year, for instance for land monitoring systems, where seasons effects can be seen (i.e., summer engagement against bushfires mainly, winter engagement against bad weather damages), scheduled activities (for times and for activity extent, i.e., number of hours flown by a certain number of UAV) that are, in case of land monitoring systems, under seasonal influence, for instance different kinds of crop control.

It is necessary to consider not only numerical but more complex data like operative “procedures” (for instance when UAV of a certain base are requested to operate in another, they fly or are transported by land means) as well as maintenance ones (for instance, in the case that only one maintenance team is available and two aircraft need support, two options are possible: the first to be repaired is the one that can be more quickly reestablished, versus “the first arrived the first served”)

- *Stochastic Data*, that is, data depending on high randomness, for instance when a certain mission is successful or not, the time occurring for an unscheduled maintenance operation, the possible increase of already foreseen time to complete a scheduled maintenance operation, the possible arise of a flight activity necessity (for instance, for land monitoring systems again, a sudden flood) as date, length and strength, and number of UAV wanted to be in flight contemporarily. In Fig. 107.27 it is shown how the randomness of such events can be simulated, obviously by knowing the statistical distribution (that, as to RAMS characteristics, brings to the knowledge of parameters MTBF and MTTR) and matching to them the random numbers automatically generated; this technique is widely utilized and it is known as “Monte Carlo method” (Fishman 1996; Chiesa et al. 2010).

Please note, as Fig. 107.27 shows, that the data inputs set have to be updated at every simulation step, i.e., following the advancing time, and it is shown that at every time the status of the system is calculated, obviously on the basis of inputs data and of previous status.

The calculation of the new system status is operated by the part of the software that makes up the real system simulation model. To understand the working simulation model well, Fig. 107.28 shows a possible organization of it, based on the verification that, at time T , UAV takeoff is requested (for scheduled activities or for an arising request of nonscheduled activity, or to continue an activity already started).

All the UAV and all the maintenance teams taken into account in the simulation are now examined, by verifying the condition that has already been defined in the previous simulation step. Afterwards, they can be modified (or not), in relation to a possible request for takeoff, and the UAV status is “available,” the same UAV passes to the status “perform a new mission” (in reality “it will be taking off, starting the new mission”), while if a UAV is in the status “needs maintenance” it can pass to the status “under maintenance” only if at least one maintenance team is “available” (and consequently such a maintenance team passes to the status “working”). Please also

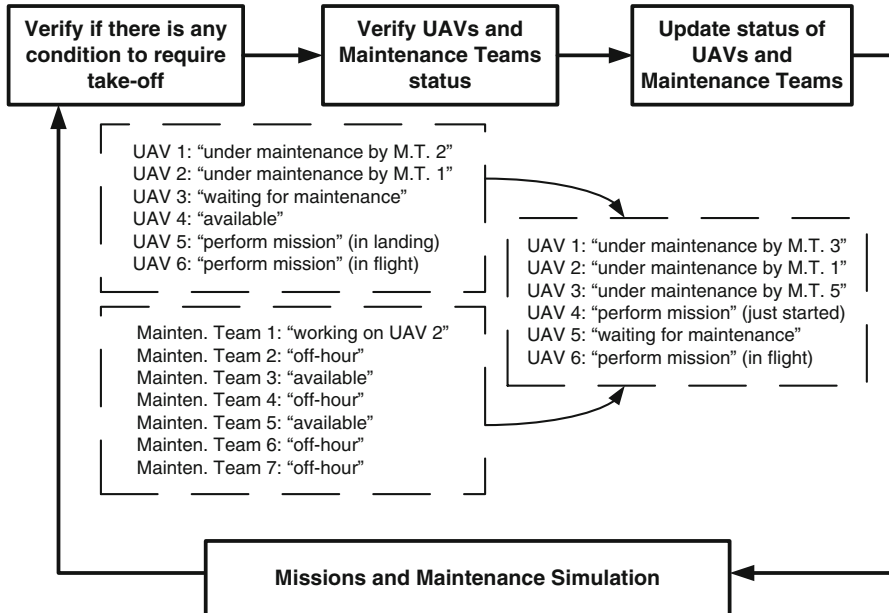


Fig. 107.28 Main loop flowchart

note that a UAV in status "under maintenance by team 'i'" can change its status if the aforesaid Maintenance Team "i" passes to the status "off-hour"; the transit could be to the status "under maintenance by team 'j'" or to "needs Maintenance" in the case that no maintenance teams were available. In particular, the dashed boxes in Fig. 107.28 show examples of these situations.

The model simulates the "active" status (not the "available" and the "waiting for Maintenance" ones) of the UAV (for those in status "under maintenance," considering also the maintenance team working on such UAVs). For both UAVs, with a possible active status ("perform mission" or "perform maintenance"), the simulation, during the step, can cause a changing of status; the possible changes are:

- From status "perform mission" to status "perform mission" if the mission has not been completed.
- From status "perform mission" to status "waiting for maintenance" if the mission is completed.
- From status "perform mission" to status "emergency re-entry" if a failure occurrence has happened.
- From status "under maintenance by maintenance team 'x'" to a status "available" if the maintenance activity has been completed in the time (casually) generated when it had started. At the same time, the status of maintenance team "x" becomes "available" unless it has reached the status "off-hour" at the same time.

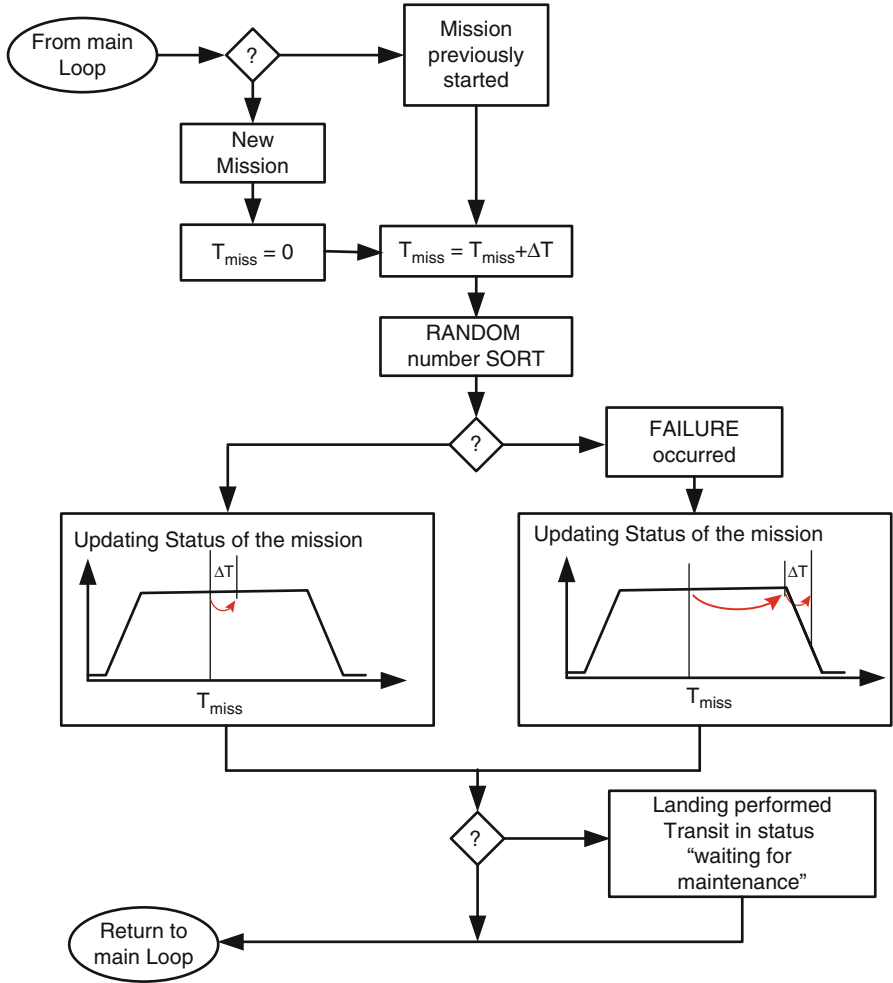


Fig. 107.29 Flowchart for “perform mission” task

- From status “under maintenance by maintenance team ‘y’ ” to a status “waiting for maintenance” if the maintenance activity has not been completed while the status of “maintenance team ‘y’ ” has become “off-hour”

In Fig. 107.29, the logic of the routine “perform mission” is shown as an explanation.

Back to Fig. 107.27, it is shown that the time will increase until a determinate value, for instance the accomplishment of a certain operational cycle or the end of a calendar year, if important effects of season become considerable; at this point, the results are disposable, but they are not immediately utilizable as they are incidental; that means deriving from the “random” numbers generated during the simulation. The problem can be solved by repeating several times the simulation

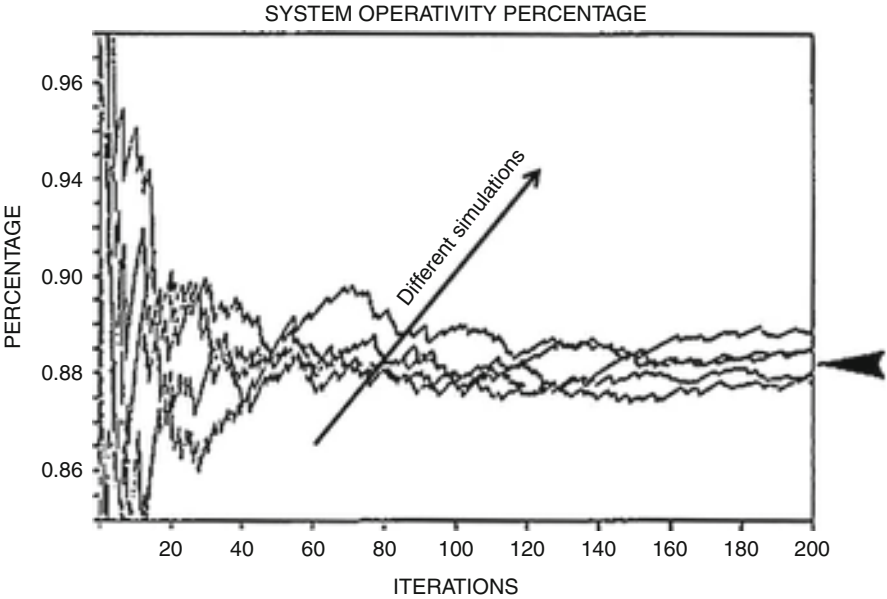


Fig. 107.30 Monte Carlo methodology output

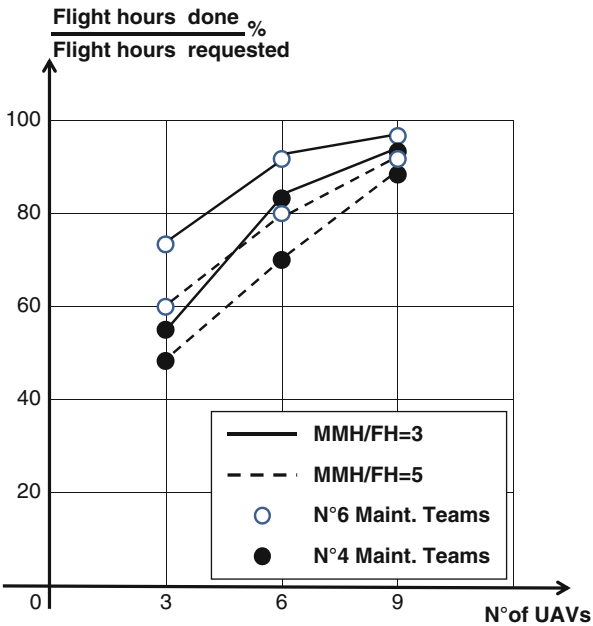


Fig. 107.31 Simulation results – example

and considering the average of results obtained from the several simulations. If the average value among several simulations is considered, this value varies in relation to the number of effected simulations; for the first simulations, high oscillations of such an average result will be noticed, and these oscillations will gradually reduce with the increase of the number of simulations done, until the result stabilizes.

In Fig. 107.30, a typical “output” of an application of Monte Carlo methodology is shown; several lines are observed, every one of which represents the evolution of the average result of various simulations; this result trends to a stabilized value with the increase of the number of simulations done. The different lines refer to different “runs” of the Monte Carlo methodology; they are different only because of the different generation of random numbers; it is interesting to observe that they tend to stabilize not on a single value but on a very close one, offering the possibility of a result (stabilized results average) with a relating “confidence level.”

As previously said, the Monte Carlo simulation instrument can be extremely useful to define a complex system, as a UAS-based operative system certainly is. A well-realized and flexible model can obviously simulate the results obtainable by the system with its different configurations; for example, the number of UAVs and their performance and logistic characteristics could be changed; different distributions of the ground bases and/or different deployment of UAS and logistic resources on the same bases could be considered, as well as the utilization of different numbers of maintenance teams, with different work shifts and different operational and/or maintenance procedures; finally, different probabilities of events that require system unscheduled activities could be simulated.

The useful results obtained for every simulated solution are the expression of efficacy and efficiency of the same solution; possible results could be: number or flight hours/number of hours required (index of System efficacy), of UAVs utilization, percentage of maintenance teams utilization and so on. An example of results obtainable with a Monte Carlo simulation model applied to a system based on UAS is shown in Fig. 107.31.

References

- Army Regulation 700-127 Integrated Logistics Support, 27 September 2007
- R. Austin, *Unmanned Aircraft System: UAVS Design, Development and Deployment* (Wiley, Chichester, 2010)
- B.S. Blanchard, *Logistics Engineering and Management*, 6th edn. (Prentice Hall, Englewood Cliffs, 2003)
- B.S. Blanchard, *Design and manage to life cycle cost* (M/A Press, Michigan, 1978)
- S. Chiesa, *Affidabilità, Sicurezza e Manutenzione nel progetto dei sistemi* (CLUT, Torino, 2007)
- S. Chiesa, P. Gianotti, Operational availability of airborne surveillance systems, in *N.A.T.O. 37th DRG SEMINAR on NON-LETHAL WEAPONS, c/o C.I.S.A.M.*, San Piero a Grado (Pisa), 1996
- S. Chiesa, K. Kostandinidis, Simulation for aircraft fleet efficacy evaluation, in *SOLE Third International Logistics Congress*, Firenze, 1987
- S. Chiesa, L. Olivero, Simulazione operativa di una flotta di aeromobili anti-incendio, in *XXXIII Convegno Internazionale delle Comunicazioni*, Genova, 1985

- S. Chiesa, S. Corpino, A. Chiesa, Advanced monitoring system based on unmanned aerial vehicles: fleet and integrated logistic system sizing by Monte-Carlo simulation, in *ICAS 27th International Congress of the Aeronautical Sciences*, Nice, 2010
- B. Dudley, Reliability prediction, START sheet. RAC J. (1997). <http://rac.alionscience.com/pdf/pred.pdf>
- G.S. Fishman, *Monte Carlo Concepts, Algorithms and Applications*. Springer Series in Operations Research (Springer, New York, 1996)
- S.B. Johnson, Introduction to system health engineering and management in aerospace, in *1st Integrated Systems Health Engineering and Management Forum*, Napa, 2005
- J.V. Jones, *Integrated Logistic Support Handbook*. Mc Graw – Hill Logistic Series, 3rd edn. (McGraw-Hill/Sole Logistics Press, New York, 2006)
- K. Keller, D. Wiegand, K. Swearingen, C. Reisig, S. Black, A. Gillis, M. Vandernoot, *An Architecture to Implement Integrated Vehicle Health Management Systems*, IEEE 0-7803-7094-5/01 (IEEE, New York, 2001)
- Lt Gen D. Deptula, The way ahead: remotely piloted aircraft in the United States air force (2009), http://www.daytonregion.com/pdf/UAV_Roundtable_5.pdf
- W. Xiaoyang, Aircraft fuel system prognostics and health management. MSc thesis, Cranfield University School of Engineering, 2012
- MIL-HDBK 1814 Integrated Diagnostics, U.S. Department of Defense
- MIL-HDBK 217 Reliability Prediction for Electronic Components, U.S. Department of Defense
- MIL-HDBK 217F N2 Reliability Prediction, U.S. Department of Defense
- MIL-STD 471 Maintainability, U.S. Department of Defense
- MIL-STD 472 Maintainability Prediction, U.S. Department of Defense
- MIL-STD 2165 Testability, U.S. Department of Defense
- MIL-STD-1629A Procedures For Performing A Failure Mode, Effects And Criticality Analysis, U.S. Department of Defense
- MIL-STD-2173 Reliability centered maintenance requirements, U.S. Department of Defense (superseded by NAVAIR 00-25-403)
- MIL-STD-13881B LSA, U.S. Department of Defense
- MIL-STD-13882B LSAR, U.S. Department of Defense
- MTain Inc., Level of Repair Analysis LORA (2011), <http://www.mtain.com/pdf/MTain%20Level%20of%20Repair%20Analysis%20Notes.pdf>
- Unmanned Aerial Vehicle Reliability Study UAVs Design, U.S. Department of Defense, 2004

Cengiz Karaağaç, Ahmet G. Pakfiliz, Fulvia Quagliotti, and
Nafiz Alemdaroglu

Contents

| | | |
|---------|--|------|
| 108.1 | Introduction..... | 2602 |
| 108.2 | Logistics Principles | 2604 |
| 108.3 | Logistics Life-Cycle Management | 2605 |
| 108.3.1 | Integrated Logistics Support (ILS)..... | 2606 |
| 108.3.2 | Performance-Based Logistics (PBL)..... | 2613 |
| 108.4 | Logistics During Materiel Solution Analysis Phase | 2615 |
| 108.5 | Logistics During Technology Development Phase | 2616 |
| 108.6 | Logistics During Engineering and Manufacturing Development and Demonstration Phase..... | 2617 |
| 108.7 | Logistics During Production and Deployment Phase..... | 2617 |
| 108.8 | Logistics at Sustainment (Operations and Support) | 2618 |
| 108.8.1 | Supply | 2618 |
| 108.8.2 | Maintenance..... | 2619 |
| 108.8.3 | Configuration Management | 2627 |
| 108.9 | Logistics at Disposal Operations Phase | 2630 |
| 108.9.1 | Disposition Methodologies | 2631 |
| 108.9.2 | Life-Cycle Cost Analysis | 2632 |

C. Karaağaç (✉)

Systems Department, STM A.Ş., Ankara, Turkey

e-mail: ckaraagac@stm.com.tr

A.G. Pakfiliz

3rd Air Supply and Maintenance Center, Turkish Air Force, Ankara, Turkey

e-mail: agpakfiliz@yahoo.com.tr

F. Quagliotti

Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy

e-mail: fulvia.quagliotti@polito.it

N. Alemdaroglu

Aerospace Engineering Department, Middle East Technical University, Ankara, Turkey

e-mail: nafiz@metu.edu.tr

108.10 Future UAV Logistics Trends..... 2634

108.10.1 Airborne Logistics Support..... 2634

108.10.2 Power Refueling by Laser 2634

References 2634

| | |
|---|--|
| Abstract | |
| Although an Unmanned Aerial Vehicle (UAV) does not require a crew onboard, it is a kind of flying platform, which requires nearly the same logistics support similar to the most manned aircraft. UAV system is a system of the systems. Thus, UAV logistics support should cover the required support for all sub-systems of the UAV system including unmanned aircraft and, ground control station. | |
| UAV logistics support system depends on many different parameters including the type of the UAV system, its operational requirements, and its operational environment. It should be a system with a reduced logistic footprint, enabling a fast deployment and high mobility. | |
| Because of urgent operational requirement for the UAVs, traditional UAV design procedures weren't followed, and first prototypes were deployed for operations without enough consideration for logistics. So, mishap rate and sustainment cost of the initial UAVs was very high because of insufficient logistic consideration during design phase. Immature UAV systems have required many modernizations and upgrades. | |
| UAV logistics support system depends on many different parameters including the type of the UAV system, its operational requirements, and its operational environment. For a small hand-launched UAV system, relatively little logistics support is required, while larger UAV systems mostly need more logistics support. | |
| UAV logistics support should be considered during every phases of the life cycle. Logistics support efforts should be proactive to predict the possible problems before they become serious issues. | |
| The goal of this chapter is to provide the reader with general knowledge on UAV Logistics for Life-Cycle Management with the logistics challenges faced by the UAV pioneers and the possible future UAV logistics trends which improve UAV employment. | |

108.1 Introduction

Although a UAV does not require a crew onboard, it is a kind of flying platform, which requires nearly the same logistics support similar to the most manned aircraft. Improving the overall efficiency and readiness of UAV systems for all purposes, including military, civil, and commercial employments, is critical. A well-established and operated logistics system serves to increase the efficiency and readiness of the UAV system while decreasing the acquisition and sustainment costs for customers.

UAV system is a system of the systems. The employment the UAV system requires the contribution of each subsystem. One small failure or problem for a subsystem of the UAV system may impede the overall success of the operation at a certain level or completely stops the system. Thus, UAV logistics support should cover the required support for all subsystems of the UAV system including unmanned aircraft and ground control station.

UAV logistics support system depends on many different parameters including the type of the UAV system, its operational requirements, and its operational environment. It should be a system with a reduced logistic footprint, enabling a fast deployment and high mobility.

In a traditional acquisition, a system would be designed to meet certain requirements, and then a prototype would be built and tested before production of the operational vehicle. But, under some circumstances, the need to field a functional system may be greater than the need for a system that meets all requirements. According to a RAND Report [1] this has been the case for some current UAVs, whose prototype vehicles were quickly pressed into actual service, even as the overall production process was being accelerated; test, evaluation, and real-world operations were taking place concurrently. USAF UAS Flight Plan [2] states that the lack of any substantive logistics planning during acquisition has resulted in large contractor logistics support (CLS) expenditures, postproduction engineering studies, and modifications that could have been mitigated with a more rigorous approach.

Lessons learned during operational employment of the UAVs have been applied to later versions of such UAVs. So, mishap rate and sustainment cost of the later version UAVs have been decreased to reasonable numbers. Although there has been some important progress, this problem still exists for the newly developed UAVs in many countries. On the other hand, retrofits and upgrades created many different versions/blocks of the same UAV while creating different logistics support packages (mostly spares) for the same fleet of the UAV.

UAV operations resemble those of manned aircraft in many ways. The similarities include aerial platforms, professional personnel, use of airspace, requirement for aviation logistics support, and training. Although a UAV does not require a crew onboard, it is piloted or directed by a pilot or an operator from a ground or airborne control station. Thus, a UAV is a kind of flying platform, which requires nearly the same logistics support similar to the most manned aircraft.

For logisticians, basic support for UAV systems is the same as for manned aircraft. However, there are some fundamental differences between manned and unmanned systems that affect the UAV logistics support system. Here are some important characteristics that are unique for UAV systems:

- Critical importance of ground systems and payloads.
- Complicated system availability tracking because of the role many different subsystems.
- Fault isolation is more complex, requiring robust onboard diagnostics.

These unique characteristics and associated implications require a different approach than that of manned platforms in some areas of life-cycle management,

especially when engaged in requirements generation, systems engineering, product support planning, and management [2].

For a small hand-launched UAV system, relatively little logistics support is required, while larger UAV systems mostly need more logistics support. Like manned aircraft, UAV systems require dedicated logistical support, which includes the equipment to deploy, transport, launch, enable communications, and sustain the UAV system according to U.S. Army UAS Roadmap [3].

Since UAVs have only been employed by armed forces with quite a few examples of civilian and experimental UAV usages so far, UAV logistics support section is mainly based on military sources. But, UAV logistics support procedures, which are explained in this chapter, would be adapted to civil and commercial UAV employments that are expected to expend in mid-term future.

108.2 Logistics Principles

UAV systems require a logistics approach similar to manned aircraft. According to U.S. Marine Corps MCWP 3-21.2 Aviation Logistics document [4], the logistics planners and the entities responsible for the acquisition should consider the seven principles of the logistics:

- *Responsiveness*: Providing the right support at the right time and at the right place. UAV system could be deployed to anywhere easily in a relatively short time and operated efficiently with its logistics support system. The logistics system should provide necessary support in reasonable time. It should also respond to unpredictable failures.
- *Flexibility*: Adapting logistics support to changing conditions. A UAV system may be used for different purposes in any environment around the world. UAV logistics system should adopt itself to the new condition without a decrease in the logistics support and system operation efficiency.
- *Attainability*: The ability to acquire the minimum essential logistics support to begin operations. UAV logistic support package should not be a burden or impediment for the efficient employment of the UAV system. If the UAV logistics footprint is increased because of high level of spares and/or redundant tools/personnel, the mobility of the UAV system decreases while increasing its sustainment cost.
- *Survivability*: Ensuring the functional effectiveness of the logistics infrastructure in spite of degradation and damage. UAV systems will have to monitor their own health by using built-in tests (BIT) and will either have to notify the ground station of any failures or, in the case where data communications are not possible, will have to be preprogrammed with decision criteria on whether or not to continue with the mission [5].
- *Sustainability*: Ensuring adequate logistics support for the duration of the operation. UAV logistics support system should provide the necessary support while enabling required level of UAV system availability and reliability for operations with reasonable acquisition and sustainment costs.

- *Economy*: Effective employment of logistics support assets. UAV support equipment and maintenance procedures should be similar to those of other UAV types and manned aircraft in order to increase interchangeability and commonality while decreasing the acquisition and sustainment costs. It will be beneficial if UAVs use similar consumables and spares to that of manned aircraft.
- *Simplicity*: Avoiding unnecessary complexity in preparing, planning, and conducting logistics operations. Although UAV systems have many different complex systems and new sophisticated technologies, their logistic support should be as simple as possible.

108.3 Logistics Life-Cycle Management

There are many different acquisition process models; most of them are very similar. The United Kingdom Ministry of Defense’s acquisition process has six phases: concept, assessment, demonstration, manufacture, in-service, and disposal/termination. The U.S. Department of Defense’s acquisition process for a requirement has also six phases. The U.S. Department of Defense’s acquisition process [6] with the relevant procedures will be used as a basis in the following parts of the chapter of UAV logistics support with also considering the United Kingdom Ministry of Defense’s acquisition process [7].

The U.S. Department of Defense’s acquisition process for a requirement starts with the Materiel Development Decision. It consists of an identification of a capability gap, a description of related risks, and a recommendation of whether or not to enter the acquisition process or use a nonmateriel solution. Materiel Development Decision is followed by life-cycle phases.

Figure 108.1 depicts the life cycle phases for an advanced UAV system on a time scale and provides a general understanding how the phases take place. The durations

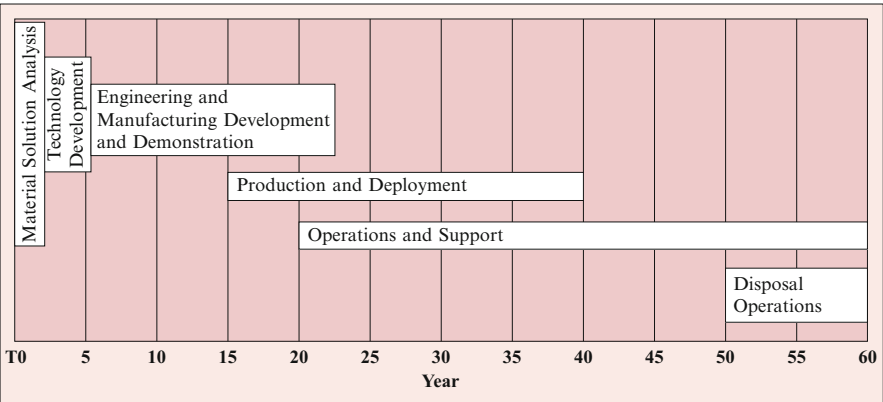


Fig. 108.1 Life-cycle phases

of the total life cycle and the phases depend on the type of the UAV system and the urgency level of the UAV system requirement. For an urgent operational requirement, UAV system development and production may be done in a couple of years, and of course, this simple UAV system could be in the inventory for a relatively short period of time. On the other hand, a complicated UAV system, which is similar to advanced manned aircraft, requires the same life-cycle approach like most manned aircraft.

Life-cycle sustainment planning shall be considered during Materiel Solution Analysis and shall mature throughout Technology Development. A Life-Cycle Sustainment Plan shall be prepared during Technology Development Phase. The planning shall be flexible and performance oriented, reflect an evolutionary approach, and accommodate modifications, upgrades, and reprourement. At every phases of the life-cycle support, logistics support efforts should be proactive to predict the possible problems before they become serious issues.

Logistics Life-Cycle Management System may be adopted for all types of acquisitions. For example, a user, such as a small Army with a requirement of a couple of UAVs, does not need to follow the first three phases of the life cycle. So, logistics support strategy/package would be prepared during Production and Deployment Phase.

108.3.1 Integrated Logistics Support (ILS)

Integrated logistics support (ILS) is an integrated and iterative process for developing materiel and a support strategy that optimizes functional support, leverages existing resources, and guides the system engineering process to lower life-cycle cost and decrease the logistics footprint (demand for logistics), making the system easier to support [6].

From the earliest stages of the systems development, the acquisition strategy and life-cycle sustainment plan will ensure that the requirements for each of the elements of ILS are properly planned, resourced, and implemented. These actions will enable the system to achieve the operational readiness levels required by the customers at the time of fielding and throughout the life cycle.

ILS is a comprehensive discipline that is applicable to all acquisition activity through life. However, the cost-effective application of ILS requires that there is a balance between benefit and cost. Research shows that over the whole life cycle of a product, the cost of acquisition is small compared to the cost of support, both financially and in unavailability of assets during operations. Reliability and maintainability have large implications on the overall cost of ownership. Thus, investment during development or production in these areas will be saved many times over the whole life of the product [8].

Although there are many descriptions for UAV classes, the NATO UAV classification described in the relevant NATO Strategic Concept [9] is the most widely accepted one. UK Joint doctrine Note 2/11 [10] acknowledges that many platforms

may well share characteristics across classes. The type of the UAV system has a critical importance on the logistics support system including ILS:

- *Class I:* These are typically hand-launched, self-contained, portable systems. Payloads are generally fixed electro-optical/infrared (EO/IR), and the system has a negligible logistics footprint.
- *Class II:* These UAVs are typically medium-sized, often catapult-launched, mobile systems. They do not usually require an improved runway surface. The payload may include a sensor ball with EO/IR, a laser range finding or designation capability, synthetic aperture radar and ground moving target Indicator (SAR/GMTI), and signal intelligence (SIGINT) pod. It has a small logistics footprint.
- *Class III:* These are typically the largest and most complex UAVs, operating at high altitude with, typically, the greatest range, endurance, and transit speeds of all UAVs. Payloads may include sensor ball(s) with EO/IR, multi-role radars, lasers, SAR, communications relay, SIGINT, automatic identification system (AIS), and weapons. Most Class III UAVs will require improved areas for launch and recovery and may be piloted from outside the joint operations area via a satellite control link. The logistics footprint may approach that of manned aircraft of similar size.

108.3.1.1 Supportability Analysis

Supportability analysis is the principal tool of ILS. It is the primary means by which the objectives of ILS are achieved and its activities consist of a series of analytical tasks which [7]:

- Influences the design of the product to take account of logistic support considerations
- Identifies support issues, readiness requirements, and cost drivers as early as possible in the product life cycle
- Defines logistic support resource requirements for the life of the product
- Creates the cost-effective physical logistic solution
- Creates the necessary data for logistic support and project decision-making, through life
- Creates the necessary information for logistic support and project decision-making.

108.3.1.2 ILS Elements

The ten ILS elements listed in Fig. 108.2 are described in the following part with using Defense Acquisition University information [6] and U.S. Army Regulation 700–127 Integrated Logistics Support [11].

108.3.1.3 Design Influence

Design influence is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality,

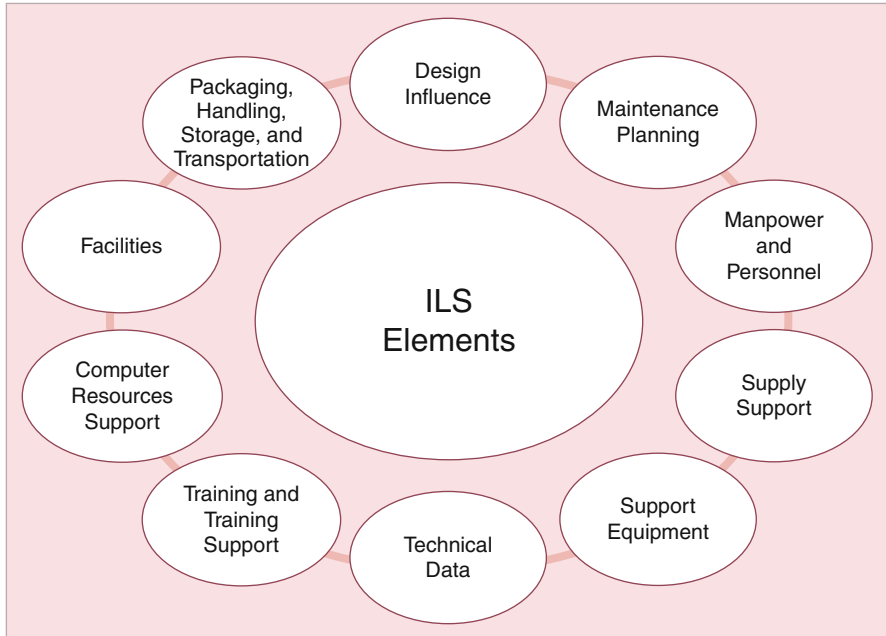


Fig. 108.2 ILS elements

cost, schedule, and user requirements. Logistics design influence acknowledges the inherent characteristic of design in recurrent logistics fundamentals. It seeks to imprint these logistics fundamentals into the end-item design as part of the system configuration and performance parameters.

108.3.1.4 Maintenance Planning

Maintenance planning is the process conducted to evolve and establish maintenance concepts and support requirements for the life of the system. It includes the following:

- Describe the maintenance concept for the system including all levels of maintenance. Identify trade-offs to be performed and maintenance considerations peculiar to the system.
- Identify maintenance tasks required to sustain the end item at a defined level of readiness, including all critical and high tasks.
- Describe the general overall support concepts. Identify proposed or actual skills; tools; test, measurement, and diagnostic equipment (TMDE), support equipment, and so on, to be available at each level of maintenance. Include analysis of possible design for discard of components and repair parts.
- Indicate strengths and weaknesses of each support alternative and the effect of the support concept on the system design, system readiness objectives (SRO),

acquisition and operations & sustainment (O&S) costs, and on affected ILS elements.

- Summarize known or planned interservice support, host nation support (HNS), interim contractor support (ICS) or life-cycle software support (LCCS), and contractor warranties. Identify proposed solutions to potential problems that may result during transition to organic support.
- Include information about planned organic depot maintenance.
- For systems being acquired for multiservice use, address the feasibility and desirability of centralized repair and supply support by a single service, the predominant user in a geographical area, or the one with centralized support capability.

108.3.1.5 Manpower and Personnel

It is the process of identifying and acquiring military and civilian personnel with the skills and grades required to operate and support a materiel system over its lifetime at peacetime and wartime rates.

The overall objective of ensuring that during equipment definition and procurement, full account is taken of the capabilities and limitations of the military and civilian personnel required to operate and maintain the equipment or facility in-service.

108.3.1.6 Supply Support

Supply support includes all the management actions, procedures, and techniques used to determine requirements to acquire, catalog, receive, store, transfer, issue, and dispose of secondary items. This means having the right spares, repair parts, and supplies available in the right quantities, at the right place, at the right time and at the right price.

The objective of supply support is to identify, plan, resource, and implement management actions to acquire repair parts, spares, and all classes of supply to ensure the best equipment/capability is available to support the warfighter or maintainer when it is needed at the lowest possible life-cycle cost.

The process includes provisioning for initial support, as well as acquiring, distributing, and replenishing inventories. This encompasses provisioning for initial support and all end-to-end replenishment supply support and supply pipeline plans and activities. Supply support must be distribution based rather than inventory based and proactive rather than reactive.

During all phases of Life-Cycle Management System, reliability improvement developments should be continuously considered, which may result in a decrease in malfunctions leading to a decrease in spare part quantities.

108.3.1.7 Support Equipment

Support equipment includes all the management actions, procedures, and techniques used to determine requirements for and to acquire the fixed and mobile equipment needed to support the operations and maintenance of a system.

The objective of support equipment is to identify, plan, resource, and implement management actions to acquire and support the equipment (mobile or fixed) required to sustain the operation and maintenance of the system to ensure that the system is available when it is needed at the lowest life-cycle cost. Support equipment consists of all equipment (mobile or fixed) required to support the operation and maintenance of a system. This includes, but is not limited to, ground handling and maintenance equipment, trucks, air conditioners, generators, tools, metrology and calibration equipment, and manual and automatic test equipment. During the acquisition of systems, program managers are expected to decrease the proliferation of support equipment into the inventory by minimizing the development of new support equipment and giving more attention to the use of existing government or commercial equipment.

108.3.1.8 Technical Data

Technical data is all scientific or technical information, equipment publications and technical drawings associated with the system, and its operation, maintenance, and support. Technical data for all support equipment are also included under this ILS element. Although computer programs and related software are not considered technical data, any documentation about computer programs, software support, and output reports is considered technical data.

Technical data are used to provide the necessary information to manufacture and support the system after deployment. The Technical Data Package (TDP) may include engineering drawing and specifications, process descriptions, and documents that define physical dimensions, material composition, performance characteristics, manufacture, assembly, and acceptance test procedures.

Technical manuals information may be presented, according to prior agreement between the contractor and the customer, in any form or characteristic, including hard printed copy, audio and visual displays, electronic imbedded media, disks, other electronic devices, or other media. They normally include operational and maintenance instructions, parts list, and related technical information or procedures exclusive of administrative procedures [12].

The manuals will be drafted in the Engineering and Manufacturing Development and Demonstration Phase and amplified in the Production and Deployment Phase. ICAO Annex 6 [13] require that the operations manual shall be amended or revised as is necessary to ensure that the information contained therein is kept up to date. All such amendments or revisions shall be issued to all personnel that are required to use this manual.

108.3.1.9 Training and Training Support

Training and training support consists of the processes, procedures, and techniques used to identify requirements for and to acquire programs of instruction, training facilities, and training systems/devices needed to train/qualify military and civilian personnel to operate and maintain a system proficiently. This includes institutional training, on-the-job training, new equipment training, sustainment training, and individual/crew training.

The goal of training and training support is to plan, resource, and implement a cohesive integrated strategy to train military and civilian personnel to maximize the effectiveness of the doctrine, manpower, and personnel, to fight, operate, and maintain the equipment throughout the life cycle.

As part of the strategy, plan, resource, and implement management actions to identify, develop, and acquire training aids devices simulators and simulations to maximize the effectiveness of the manpower and personnel to fight, operate, and sustain equipment at the lowest total ownership cost.

Training is the learning process by which personnel individually or collectively acquires or enhances predetermined job-relevant knowledge, skills, and abilities by developing their cognitive, physical, sensory, and team dynamic abilities. The “training/instructional system” integrates training concepts and strategies and elements of logistic support to satisfy personnel performance levels required to operate, maintain, and support the systems. It includes the “tools” used to provide learning experiences such as computer-based interactive courseware, simulators, and actual equipment (including embedded training capabilities on actual equipment), job performance aids, and interactive electronic technical manuals.

All UAV systems should be developed with the maximum use of personalized learning management, simulation-enabled computer-based training and virtual instruction. Training for support equipment should also be a part of the training package.

108.3.1.10 Computer Resources Support

Computer resources support is all the management actions, procedures, and techniques used in determining requirements and acquiring hardware, software, facilities, documentation, personnel, and manpower required to develop and sustain computer resources for operation and maintenance of a system, including planning and support requirements for post-deployment software.

Computer resources support element evaluates the operational and support capabilities of the various computer resource systems. The evaluations will center on the overall operational and support concepts of computer resources, the effectiveness of computer resources, quality and accuracy of data, and man/machine interface. The computer resources support evaluation will include evaluation of the built-in test (BIT) system, all computer resources that interface with the test article, all off-equipment computer resources the maintainer may interface with to complete a task, and all test article systems involving computer resources [14].

As the primary end item, support equipment, and training devices increase in complexity, more and more software is being used. The expense associated with the design and maintenance of software programs is so high that one cannot afford not to manage this process effectively. It is standard practice to establish some form of computer resource working group to accomplish the necessary planning and management of computer resources support. Computer programs and software are often part of the technical data that defines the current and future configuration baseline of the system necessary to develop safe and effective procedures for operation and maintenance of the system. Software technical data comes in many

forms to include, but not limited to, specifications, flow/logic diagrams, computer software configuration item definitions, test descriptions, operating environments, user/maintainer manuals, and computer code.

108.3.1.11 Facilities

Facilities consist of all the management actions, procedures, and techniques used to determine requirements for and to acquire the permanent and semi-permanent real property assets needed to support operation, maintenance, and storage of a system and its support equipment.

The objective of facilities is to identify, plan, resource, and acquire facilities that will allow storage, maintenance, and training to maximize the effectiveness of a systems operation and logistic support system at the lowest life-cycle cost and enable responsive support to the warfighter by identifying and preparing plans for the acquisition of facilities.

Facilities consist of both semipermanent and permanent real property assets for the support of a system. Studies that define location, types of facilities, facility improvements, space needs, security requirements, environmental requirements, and equipment are also considered in facilities. This element includes new and modified facilities for supply storage, equipment storage, ammunition storage, maintenance, etc.

Although nearly all of the UAV systems are mobile systems, most of them require some infrastructure such as hangars, aprons, taxiways, runways, buildings, communication lines, and electricity. The facilities for support equipment should also be a part of the facility requirement. Moreover, facilities for the storage of hazardous material such as lithium batteries required for UAV operations must be planned.

108.3.1.12 Packaging, Handling, Storage, and Transportation (PHST)

The PHST concept describes any anticipated special considerations for the packaging, handling, storage, and transportability required for deployment and sustainment of the system as well as the process for determining these requirements. The operational environment conditions (temperature, humidity, etc.) and potential for use of hazardous materials must be considered.

Transportation of a UAV system will range widely from the one or two man-carried backpacks of a small UAV to a strategic UAV system with a requirement of strategic airlift. Since the strategic airlift platforms are very rare at the moment, the air transportation of the UAV system requires special consideration. Transportation of the UAV system from main base to the operational deployment location should be planned carefully.

A typical medium altitude long-endurance (MALE) UAV system like ANKA, Heron, MQ-1B Predator, or MQ-9 Reaper consists of 4 UAV platforms, one or two ground control station, one or two ground data terminal, some support equipment, and an important amount of spares. These systems and equipment should be packed properly for the air transportation. Strategic airlift is required for the deployment of such UAV system. Although the required airlift capacity depends on the type of the

UAV system, typically two to three An-124, IL-76, or C-17 sorties, and six to seven C-130 or C-160 sorties are sufficient.

108.3.2 Performance-Based Logistics (PBL)

B.D. Coryell [15] states that PBL means different things to different people, but principally it is defined as establishing a contract or agreement with a logistics provider for a specified level of performance for an item at a system, subsystem, or component level. This level of performance can be achieved by a contractor, the government, or a combination of both.

According to Defense Acquisition University document of Performance-Based Logistics: A Program Manager's Product Support Guide [16], PBL is the purchase of support as an integrated, affordable, performance package designed to optimize system readiness and meet performance goals for a weapons system through long-term support arrangements with clear lines of authority and responsibility. Simply put, performance-based strategies buy outcomes, not products or services.

The application of PBL can provide numerous benefits. The most visible benefits of PBL are in the operational portion of the program life. PBL should reduce logistics costs and footprint, increase system reliability, and mitigate obsolescence [15]:

- *Reducing logistics cost and footprint:* PBL will get the best capability return on the funds available in the Defence Forces to support weapons systems. In other words, incentive-oriented agreements should reduce product support costs through reduced infrastructure, reduced obsolescence, and reduced provisioning and data requirements. PBL will provide more visibility and control over operations and support costs which will ultimately improve the ability to manage program resources. With this visibility and control, PBL should create a reduction in the overall demand for logistic support and reduce the logistics footprint.
- *Increasing system reliability:* Incentive oriented agreements will improve weapon system availability and provide warfighters with increased operational readiness. The provider is provided incentives to ensure that set levels of system support performance are achieved. If the contract is correctly incentivized, the items, components, or subsystems that configure the weapon system will be designed to fail less. The PBL supplier then has the financial incentive to continuously improve performance because it has a bottom-line impact. This in turn will improve customer satisfaction. PBL will result in confidence in the warfighter that the system will provide the required performance capabilities when needed in combat.
- *Mitigating obsolescence:* PBL provides a powerful tool for mitigating obsolescence and making continuous modernization a reality for legacy weapon systems. PBL clearly fulfills the need for continuous modernization and obsolescence mitigation. Because it provides incentives for private industry to continually improve reliability and the performance of the managed system. In this manner, private industry conducts research and development and acquisition activities

in-stride with performing their contracted logistics support contract. Consistent with evolutionary acquisition practices and the spiraling of technology as it matures, the PBL contractor can leverage research and development efforts for spirals into legacy component system reliability.

The essence of PBL is buying performance, instead of the traditional approach of buying individual parts or repair actions. It is important to note that, although the fundamental concept of buying performance outcomes is common to each PBL arrangement, the PBL strategy for any specific program or commodity must be tailored to the operational and support requirements of the end item.

A key component of any PBL implementation is the establishment of metrics. Since the purpose of PBL is “buying performance,” what constitutes performance must be defined in a manner in which the achievement of performance can be tracked, measured, and assessed [16]. U.S. DoD Momerandum [17] lists the following criteria to be used for PBL performance objectives:

- *Operational availability*: The percent of time that a system is available for a mission or the ability to sustain operations tempo.
- *Operational reliability*: The measure of a system in meeting mission success objectives (percent of objectives met, by system). Depending on the system, a mission objective could be a sortie, tour, launch, destination reached, or other service- and system-specific metric.
- *Cost per unit usage*: The total operating cost divided by the appropriate unit of measurement for a given system. Depending on the system, the measurement unit could be flight hour, steaming hour, launch, mile driven, or other service- and system-specific metric.
- *Logistics footprint*: The government/contractor size or “presence” of deployed logistics support required to deploy, sustain, and move a system. Measurable elements include inventory/equipment, personnel, facilities, transportation assets, and real estate.
- *Logistics response time*: The period of time from logistics demand signal sent to satisfaction of that logistics demand. “Logistics demand” refers to systems, components, or resources (including labor) required for system logistics support.

There is no “one-size-fits-all” approach to PBL. Similarly, there is no template regarding sources of support in PBL strategies. It comprises a combination of public (organic) and private (commercial) support sources. Finding the right mix of support sources is based on best value determinations of inherent capabilities and compliance with statutes and policy. This process will determine the optimum PBL support strategy within the product support spectrum, which can range from primarily organic support to a total system support package provided by a commercial original equipment manufacturer [16].

According to University of Tennessee and Supply Chain Visions’ White Paper [18], PBL may be implemented at four levels:

- *Level 1 – Component*: The lowest level of PBL implementation is at the system component level, such as aircraft tires. At this level the “performance” purchased is the consistent and timely delivery of needed components. The scope of support

responsibility for the contractor is generally narrow – focusing primarily on supply chain activities.

- *Level 2 – Major subsystem:* At level 2 the scope of support and corresponding performance outcomes broadens, and the customer/operator-contractor relationship begins to become more collaborative. The focus is not just on delivery speed, but on broader improved material availability, which necessitates enlarging the scope of support to include not only supply chain activities but also encompassing repair processes, engineering and technical support, configuration management, and even minor modifications and process improvements. One of the best examples of a major subsystem PBL is the GE F404 Aircraft Engine.
- *Level 3 – Platform availability:* At this level the scope of support is the buying warfighter performance. The warfighter doesn't think in terms of employing components or subsystems; their focus is on the availability and readiness of weapon system platforms – an aircraft, a ship, or a tank – the tools of combat capability. Consequently, this level of PBL transfers even more responsibility for management of support activities to the contractor. One example of a level 3 PBL is the Lockheed Martin and U.S. Air Force Total System Support Partnership (TSSP) program involving the F-117 Nighthawk.
- *Level 4 – Mission:* While level 3 PBLs optimize weapon system availability for the warfighter, what the warfighter ultimately needs in combat is not only for the system to be available to perform its mission but to successfully complete the mission. One of the best examples is the Army Shadow Tactical Unmanned Aerial Vehicle program.

108.4 Logistics During Materiel Solution Analysis Phase

The purpose of Materiel Solution Analysis Phase is to assess potential materiel solutions. During Materiel Solution Analysis Phase, ILS issues, supportability deficiencies, and opportunities for improvements and efficiencies are evaluated.

Prior to the Materiel Development Decision, supportability objectives must be clearly defined. System supportability objectives provide the basis upon which product support capabilities are evaluated. They include design, technical support data, and maintenance procedures to facilitate detection, isolation, and timely repair and/or replacement of system anomalies.

The goal is to ensure product support capabilities, requirements, and risks are considered early and throughout the acquisition continuum, in order to minimize support costs, provide end users with required resources, and attain materiel availability objectives.

To facilitate an evaluation of product support capabilities, from concept through fielding, required system reliability, planned maintainability, and supportability methods, practices, and processes must be clearly defined and integrated with the systems engineering process. Additionally, the concept of operations must be

defined to provide the basis for assessing system requirements and overall product support capabilities, as well as the initial definition of the system maintenance and support concept for sustainment.

Life-cycle sustainment planning and execution should seamlessly span a system's entire life cycle, from Materiel Solution Analysis to disposal. It translates force provider capability and performance requirements into tailored product support to achieve specified and evolving life-cycle product support availability, reliability, and affordability parameters.

108.5 Logistics During Technology Development Phase

The purpose of Technology Development Phase is it to reduce technology risk and to determine the appropriate set of technologies to be integrated into the full system. It is an iterative process designed to assess the viability of technologies while simultaneously refining user requirements. The Technology Development Phase begins when a materiel solution has been endorsed in the end of the Materiel Solution Analysis Phase.

Logistics support goals for the Technology Development Phase are to refine sustainment objectives and requirements, demonstrate system feasibility and mature technology, and design in sustainment and establish Product Support Package requirements.

The objective of the Technology Development Phase is to reduce technology risk. From a sustainment standpoint, this means designing critical supportability aspects into the system and developing the system support package.

A sustainment concept should be prepared in the Materiel Solution Analysis Phase. At the beginning of the Technology Development Phase, the data gathered while developing the sustainment concept should be used to establish sustainment requirements. The sustainment concept should also feed the design of the product support package. The Product Support Package has two major components:

- The Maintenance Plan and Requirements describe the management approach for developing and implementing the maintenance plan.
- Product Support Package Elements include an overview of technical documentation (paper and electronic), Test/Support & Calibration Equipment, Manpower & Training/Computer-Based Training, supply support, packaging, handling, storage and transportation, facilities, and computer resources support.

The program manager should develop a product support strategy for life-cycle sustainment and continuous improvement of product affordability, reliability, and supportability while sustaining readiness. The support strategy is a major part of the acquisition strategy which includes design, technical support data, and maintenance procedures to facilitate detection, isolation, and timely repair and/or replacement of system anomalies. This includes factors such as diagnostics, prognostics, real-time maintenance data collection, and human system integration considerations.

108.6 Logistics During Engineering and Manufacturing Development and Demonstration Phase

The purpose of Engineering and Manufacturing Development and Demonstration Phase is to develop a system or an increment of capability, complete full system integration, develop an affordable and executable manufacturing process, ensure operational supportability with particular attention to minimizing the logistics footprint, implement human systems integration, design for producibility, ensure affordability, and demonstrate system integration, interoperability, safety, and utility.

During Engineering and Manufacturing Development and Demonstration Phase, Product Support Plan is developed. Product (or logistics) Support Plan is a document that outlines how product support and sustainment of a weapon system will be managed over its life cycle. Product support is the package of support functions necessary to maintain the readiness and operational capability of weapon systems, subsystems, and support systems. It encompasses all critical functions related to weapon system readiness, including materiel management, distribution, technical data management, maintenance, training, cataloging, configuration management, engineering support, repair parts management, failure reporting and analyses, and reliability growth.

Logistics support model is to be defined at System Development and Demonstration Phase. The source of the support may be organic or commercial, but the product support plan's primary focus is to optimize customer support and achieve maximum availability at the lowest total ownership cost.

UAV platform may incorporate a modular design for flexible payload configuration. The configurable pods and payload bays can house a wide array of sensors and weapons. Fault isolation within UAV system must mostly be accomplished by using built-in test (BIT).

The design of the UAV systems should consider the climatic extremes ranging from high temperatures in arid regions, through frigid temperatures with associated ice and snow, to tropical climates with high relative humidity and excessive rainfall.

108.7 Logistics During Production and Deployment Phase

The purpose of Production and Deployment Phase is to achieve an operational capability that satisfies the mission need. Operational test and evaluation shall determine the effectiveness and suitability of the system. During the Production and Deployment phase, the system should achieve operational capability that satisfies mission needs. It has two main stages: Low-Rate Initial Production and Full-Rate Production and Deployment.

Low-Rate Initial Production (LRIP) is intended to result in completion of manufacturing development in order to ensure adequate and efficient manufacturing capability to produce the minimum quantity necessary to provide production-representative articles for testing and evaluation.

Full-Rate Production and Deployment should be based on the results of testing initial production articles and refining manufacturing processes and support activities which indicate the product is operationally capable, lethal and survivable, reliable, supportable, and producible within cost, schedule, and quality targets.

During Production and Deployment Phase, all documentation is finalized, necessary training of personnel for maintenance and operations are accomplished, and Product Support Package/Performance-Based Logistics (PBL) is implemented.

108.8 Logistics at Sustainment (Operations and Support)

Logistics plays a crucial role at sustainment phase, which takes place between acquisition to disposal phase. In logistics point of view, sustainment is the longest part of the ILS. This period lasts 10–30 years up to requirements, economy, and technological trends. During this period UAV systems are required a perfectly coordinated and regularly manipulated logistic system.

The purpose of logistics at sustainment is to execute a support program that meets materiel readiness and operational support performance requirements and sustains the system in the most cost-effective manner over its total life cycle.

During sustainment, logisticians execute a support program that meets materiel readiness (availability) and operational support performance requirements and sustains the system in the most cost-effective manner over its total life cycle. When the system has reached the end of its useful life, it shall be disposed in an appropriate manner.

System modifications are initiated whenever necessary to improve performance and reduce ownership costs. The users conduct continuing reviews of sustainment strategies, utilizing comparisons of performance expectation as defined in performance agreements against actual performance measures. Sustainment strategies are revised, corrected, and improved as necessary to meet performance requirements.

108.8.1 Supply

Supply is an essential part of logistics for all kind of systems. In maintenance point of view, its necessity comes from the requirement of material flowing, because maintenance process can't occur without it. In aviation supply plays a crucial role, and also UAV systems pursue this rule. This section of logistics deals with acquisition, packaging, shipping spares, and meeting them with technicians.

To supply all the equipment for an Unmanned Air System (UAS) rapidly is required to establish a reliable and robust supply mechanism. Operational requirements, especially expected flying hours, declare the time toleration for spare packages. In order to manage products and ensure high levels of equipment availability with an online asset management tool which tracks complete UAS flying hours, life expired and obsolete parts are replaced in a timely manner that fits in the mission cycle. Emergency logistical support is also maintained through the level of

spares management and rapid distribution system. The supply materials may be consumables or replaceable components. Reg Austin defines them as follows [19];

- Consumables: Depending upon the size of the system and the quantities required, lubricants, cleaning materials, batteries, and fuel may be carried within the Control Station (CS) especially if the CS vehicle uses the same type of fuel as the aircraft. Otherwise, for larger systems or for safety reasons, fuel will be carried in a separate vehicle or browser.
- Replaceable components: These kinds of components are listed in the Maintenance Manuals, and the replenishment of these to the UAV system's operating base is required from the logistics support organizations.

To supply all of the required equipment rapidly, a robust and well-defined chain must be established. This chain is generally called as supply chain. A supply chain is a complex logistics system in which raw materials are converted into finished products and then distributed to the final users. It includes suppliers, manufacturing centers, warehouses, distribution centers, and retail outlets. At this point reflection of the supply chain on the supply process of UAVs is inspected. Not only the regular periodic maintenances but also the unplanned repairs require spare materials. Establishing a supply-flow method is a crucial target for logistic planner. For this purpose an effective and optimal supply chain must be planned before sustainment and applied during sustainment. In planning phase the following factors are taken into account, and also this list may increase or decrease according to the requirements:

- Flying hours: The expected flying hours of UAV in a year, also this includes consecutive flying hours
- Number of the UAVs and ground assets: The number of UAVs and ground assets achieved during procurement and probable loss and additive assurance amount during sustainment
- Percentage of availability during life cycle: The amount of available UAVs and ground assets in a definite period of time during life cycle
- Probable service life: Expected life cycle of UAVs and whole related assets (ground assets, avionics, payloads, etc.)
- Obsolescence program: To put probable expired and obsolete parts on a time table

By using the aforementioned requirements, an analysis is conducted, and an application plan is prepared. This plan must support all the maintenance levels. Additionally this plan should meet the readiness requirements of UAV system. After preparing the plan, it is applied to practical life. A supply chain is established, and related stakeholders are connected to the chain network. An example of UAV system supply chain is shown in Fig. 108.3.

108.8.2 Maintenance

Maintenance is most critical and time-consuming part of logistic activities during the life cycle of all kind of systems. Within the airline industry, it has been estimated

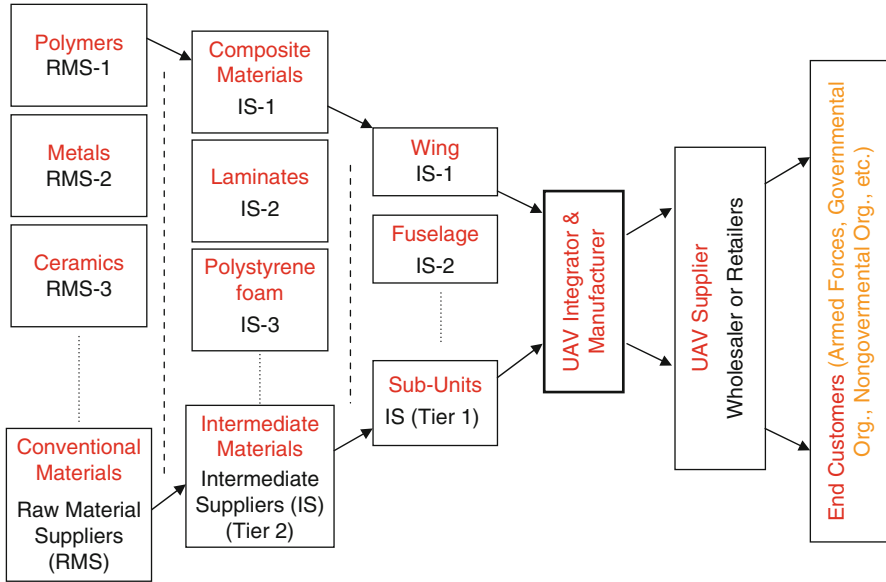


Fig. 108.3 A supply chain example for UAV system

that for every hour of flight, 12 man-hours of maintenance occur. Maintenance was defined as any activity performed on the ground before or after flight to ensure the successful and safe operation of an aerial vehicle. Maintenance generally includes the following:

- Assembly
- Fuelling
- Preflight inspections
- Repairs
- Regular maintenances
- Software updates

Maintenance activities may involve the vehicle as well as equipment such as the UAV ground control station. According to the maintenance point of view, activities can be classified as either corrective or preventative. Corrective maintenance involves the repair or replacement of systems that have experienced wear or damage. In many cases, corrective maintenance is nonroutine and is performed in response to an operational event such as a hard landing or a system failure. Nonroutine tasks are more likely to require fault diagnosis, problem-solving, and special skills. Preventative maintenance tasks are performed before a problem occurs and may involve tasks such as inspections, lubrication, or the replacement of components at predetermined intervals. Preventative maintenance tasks are typically routine and tend to require a more limited range of skills and knowledge than corrective maintenance tasks [20].

108.8.2.1 Maintenance Plan for UAVs

In general perception a maintenance plan describes the requirements and tasks to be accomplished for achieving, restoring, or maintaining the operational capability of the UAVs. A maintenance plan is a concise summary of maintenance requirements that is performed on the system. It is prepared at the beginning of the project with the other project plans. A maintenance plan should preserve the inherent design levels of the UAV subsystems reliability with minimum expenditure of maintenance and support resources.

A maintenance plan establishes and defines the line-replaceable units (LRUs) categories in groups of repairable LRUs and consumable LRUs that required maintaining UAVs. For each LRU, the maintenance plan identifies the maintenance level authorized to perform the required maintenance tasks. The maintenance plan also provides Source, Maintenance, and Recoverability (SMR) codes, which summaries the maintenance concept of each LRU.

For preparing a maintenance plan, some preliminary estimated information is used. This information reflects the general requirements of UAVs. These requirements include, but are not limited to, the following:

- Flying hours: The expected flying hours of UAV in a year, also this includes consecutive flying hours.
- Reliability: The probability of realization of the required function in identified case and in between a certain time.
- MTBF (mean time between failures): Average time duration of a system or equipment not working or malfunctioning in a certain interval.
- MTBM (mean time between maintenance): Average time between all maintenance actions (protective and corrective).
- MCMT (mean corrective maintenance time): Mean corrective maintenance time is composed of some elements. It is mathematically modeled as follows:

$$MCMT = T_p + T_{fi} + T_d + T_i + T_r + T_a + T_{co} + T_{st}$$

T_p = Mean readiness time

T_{fi} = Trouble shooting time

T_d = Mean disassembling time

T_i = Mean replacing time

T_r = Mean assembling time

T_a = Mean corrective time

T_{co} = Mean checkout time

T_{st} = Mean starting time

- Mean preventive maintenance time – MPT: The preventive maintenance contains the procedures to keep the system in a certain performance level and the functions such as periodic inspections, services, and calibration. MPT is the average of time duration passed during preventive maintenance.
- Logistic delay time – LDT: Total duration of required time to receive the repair equipment, spare parts, or maintenance test equipment.
- Maintenance labor hours per month – MLH/month: The total cost of monthly maintenance labor man-hour.
- Probable expired and obsolete parts on a time table.

After determining the life-cycle requirements of UAVs, a maintenance plan is established by using the information of requirements and the experiences of contractor and customer. Maintenance plan shapes the maintenance structure during sustainment.

108.8.2.2 Maintenance Levels

Maintenance level structure is a strategic decision which depends on the budget of project, verdict of the contractor, and the facilities of customer. Also defining boundaries and abilities of these levels is another decision area. Decision process for maintenance levels starts in the planning phase. Maintenance levels and borders of them, the responsibilities of the project stakeholders, are determined according to the requirements and defined in the maintenance plan.

The organizational structure of maintenance will be designed to produce appropriate sortie rates in order to meet requirements. Both customer and contractor personnel will operate the system and provide technical expertise and maintenance capability down to the subcomponent level. Because of the requirement to maintain UAVs at the same level of safety and availability as manned aircraft, a similar maintenance concept and execution like manned aircraft is required, especially for the UAVs larger than small UAVs.

In order to establish a reliable and robust maintenance system for UAVs, probable generic operational employment scenarios can be used during the planning phase. These scenarios are composed of, but not limited to, the following:

- First, create a required UAV employment scenario
- Define periodic maintenance requirements such as operational-level maintenance, depot-level maintenance
- Define possible disorders such as mishaps and emergencies which create logistic requirements
- Show all operational steps and logistic footprints on a sketch

As mentioned above and in the maintenance plan subsection, planning is the dominant part of shaping the maintenance level. At this point it is proper to explain the maintenance levels. In general maintenance concept has three maintenance levels. These levels are operational-level (O-Level), intermediate-level (I-Level), and depot-level (D-Level) maintenances. The relation of levels is shown in Fig. 108.4.

In the below figure the relationship and the capabilities of the maintenance levels are outlined. Three-level maintenance activities can be summarized as follows [21];

- Operational-level maintenance: This level consists of the on-equipment tasks necessary for day-to-day operation, including inspection and servicing and remove-and-replace operations for failed components
- Intermediate-level maintenance: This level consists of off-equipment repair capabilities possessed by operating units and in-theater sustainment organizations. These capabilities can be quite extensive and include remove-and-replace operations for subcomponents of line-replaceable units (LRU), local manufacture, and other repair capabilities

- **Depot-level maintenance:** This level consists of all repairs beyond the capabilities of the operating units, including rebuild, overhaul, and extensive modification of equipment platforms, systems, and subsystems

Generally the responsibility borders between maintenance levels are drawn by using maintenance allocation chart (MAC). MAC reflects the responsibility allocation for the performance support tasks for each LRU and assigns the tasks to the proper maintenance activity in accordance with the maintenance concept. MAC structure generally includes Logistic Control Number (LCN), Description (item nomenclature), and Source Maintenance Recovery (SMR) codes.

The details and aims of the maintenance levels and their capabilities are detailed in the following subsections.

Operational-Level (O-Level) Maintenance

This is the first step of the maintenance levels, so it has the least maintenance capability. The maintenance activities at the O-Level are performed on the UAV systems and subsystems in the flight line and in the hangar. O-Level maintenance for UAV system consists of those maintenance tasks intended to keep the system operational and prevent deterioration.

O-Level maintenance is performed by operating units on a day-to-day basis in support of their operations. The O-Level maintenance mission is to maintain assigned UAV equipment in a full mission-capable status while continuing to improve the local maintenance process. While O-Level maintenance may be done by I-Level or D-Level activities, O-Level maintenance is usually accomplished by maintenance personnel assigned to operation area. Generally, O-Level maintenance can be grouped under “inspections, servicing, and handling” categories.

The main problem in O-Level is fuselage defects of UAVs. Most small UAVs and some parts of big UAVs are produced by composite materials. During their employment, the air vehicles can be smashed, cracked, chipped, or dented. Generally O-Level technicians are not trained to repair composites.

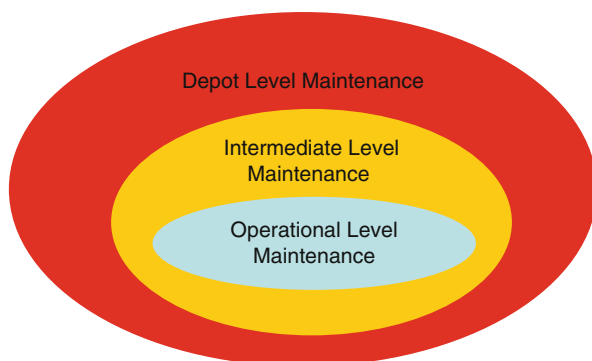


Fig. 108.4 Relationship of maintenance levels

O-Level maintenance includes, but is not limited to, the following activities:

- UAV system servicing
- Failure detection and isolation using the UAV System built-in test (BIT)
- Replacement of faulty LRUs
- Limited cabling and structural repair on UAV System and subsystems
- Preventive maintenance checks and services (UAV System serviceability checks, cleaning, etc.)
- Periodic inspection or replacement to comply with scheduled maintenance requirements, corrosion prevention, detection, and removal
- Inspection after abnormal events
- Functional tests
- Loading software to UAV System LRU
- Simple adjustments/calibrations
- Packing for transportation
- Discarding defective LRUs
- Shipping defective LRUs to intermediate level

Intermediate-Level (I-Level) Maintenance

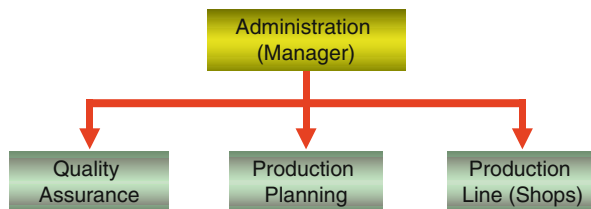
I-Level maintenance has more capabilities than O-Level maintenance. I-Level personnel focus on only repair and testing elements and sub-elements of UAVs, rather than operational activities. This level is an intermediate level but not in the middle; its abilities are closed to D-Level abilities. I-Level personnel are not only composed of technicians but also engineers and administrative. Generally I-Level maintenance organization is made up of maintenance managers, staff divisions, and production divisions.

I-Level maintenance mission is to enhance and sustain the combat readiness and mission capability of supported activities by providing quality and timely material support at the nearest location with the lowest practical resource expenditure.

The maintenance activities that are performed at I-Level involve maintenance procedures, which require specialized skills and resources for inspection, testing, refurbishment, or repair. I-Level maintenance includes, but is not limited to, the following activities:

- Detection and isolation, by using test equipment for selected items, the faulty SRU (shop replaceable unit) in the faulty LRU (line-replaceable unit) which was sent from the O-Level
- Test of LRU, by using test equipment, whose SRU has been replaced
- Calibration of designated equipment
- Processing aircraft components from stricken aircraft
- Providing technical assistance to supported units
- Performance of on-aircraft maintenance when required
- Age exploration of aircraft and equipment under reliability-centered maintenance (RCM)
- Loading software to LRU
- Intermediate-Level LRU periodic maintenance
- Off-system limited cabling and structural repair

Fig. 108.5 Organizational chart for I-Level maintenance



I-Level maintenance has more capabilities than O-Level maintenance. This capability increase is caused by both personnel and equipment. In I-Level maintenance the technicians are well trained about the systems; there are few engineers taken system course. Also this maintenance level covers advanced test equipments, Automatic Test Equipments (ATEs), I-Level technical orders (TOs), and shops. The shops are electrical, electronics, mechanical, and composite shops. In addition to these capabilities, a forward distribution depot consists in I-Level maintenance structure. I-Level has considerable capability, but this capability is limited when compared with the D-Level maintenance capabilities. Notwithstanding I-Level has a chance for obtaining new abilities, but not as much as D-Level.

In order to establish I-Level maintenance, a segmentation structure is required. This structure is not required in O-Level maintenance, because it is fulfilled with only a group. Every sub-function of O-Level can be realized with one or few people. But the sub-functions of I-Level require more personnel. Basic organization structure for I-Level maintenance is shown in Fig. 108.5.

Depot-Level (D-Level) Maintenance

Maintenance actions that cannot be accomplished at O or I-Level are conducted in D-Level maintenance. D-level maintenance has the capability to do anything necessary to repair failed equipment or to make periodic D-Level maintenances. D-level maintenance facilities have the wider range of tools, test equipment, and knowledgeable maintenance personnel. Fabrication of structural parts, major overhauls, refurbishment, and rebuilding of equipment can be done at D-Level.

D-level maintenance is performed at aviation industrial establishments to ensure continued flying integrity of airframes and flight systems during subsequent operational service periods. D-level maintenance is also performed on material requiring major overhaul or rebuilding of parts, assemblies, subassemblies, and end items. D-level maintenance supports O-level and I-level maintenance by providing engineering assistance and performing maintenance beyond their capabilities.

The maintenance activities that are performed at D-Level maintenance of UAVs involve maintenance procedures, which require specialized skills and resources for inspection, testing, refurbishment, or repair. D-Level maintenance of UAVs generally includes the following activities:

- Standard D-level maintenance of UAVs and ground assets
- Rework and repair of engines, fuselage, and components
- Faulty SRU repair

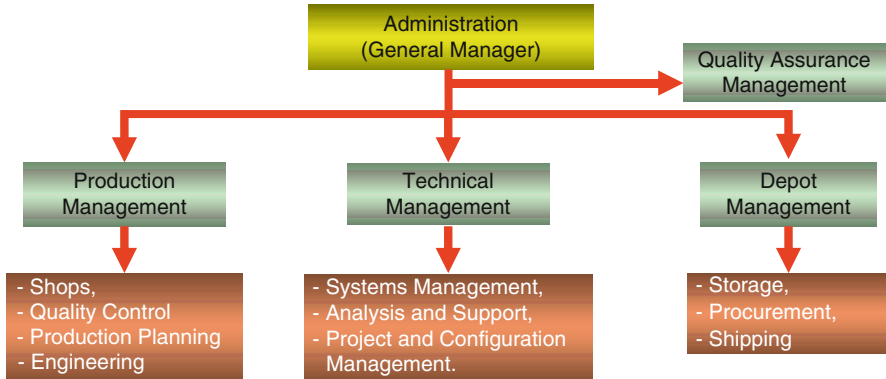


Fig. 108.6 Organizational chart for D-Level maintenance

- Extensive structural control, repair, and modifications
- Calibration exceeding O- and I-Levels
- Harness repair exceeding O- and I-Levels
- Configuration control, authorization, and classification
- Update maintenance plans and documents when required
- Technical management
- Depot-level periodic maintenance and inspection
- Participate in design change process to improve maintenance

Organization structure of D-Level maintenance is relatively larger than that of I-Level. As mentioned above D-Level includes manufacturing parts, modifying, testing, inspecting, sampling, and reclamation. In order to achieve these functions, D-Level has a comprehensive organizational structure. Every sub-function of D-Level requires an additional segmentation structure. A general organization structure for D-Level maintenance of UAVs is shown in Fig. 108.6.

Also it is beneficial to deal with the kinds of shops in D-Level maintenance of UAV systems. The shops cover whole facilities of UAV system. This means that they cover not only UAVs but also ground assets. For this purpose the shops and their capabilities are listed as follows:

- Structural (composite) shop: Standard structural repair processes and required kits, composite material processes (repair, replace, modify, etc.), wet layups technology, hinge replacement, panel replacement, integral fuel tank sealing, finish and touch-up repairs, and structural checks
- Mechanical shop: Standard mechanical repair processes and required kits, maintenance and repair processes of UAV motors and propellers, maintain processes of motor-and motor and related parts, replacement of faulty mechanical LRU
- Electronic shop: Standard electronic repair processes and required kits, fault detection and isolation, removal and replacement capability for faulty LRUs,

repair and maintenance processes for avionics, motor electronics (Full Authority Digital Engine (or Electronics) Control – FADEC), payloads and ground assets, and fault detection and isolation for data link

- Electrical shop: Standard electrical repair processes and required kits, processes for defining condition of electrical assemblies, fault detection and isolation for UAVs electricity, and system generator and power supplies

108.8.3 Configuration Management

In this part configuration management (CM) process for UAV systems is identified and described. CM is a field of management that focuses on establishing and maintaining consistency of a system or product's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life [22]. Before entering into more details about CM, it will be beneficial to give some related definitions. The definitions are given in terms of UAV.

- Configuration: Physical and functional characteristics of a UAV system that is defined in the technical documents and reached form after using.
- Configuration verification: Conformance control of a UAV system according to configuration documents.
- Configuration item (CI): CI refers to the fundamental structural unit of a configuration management system.
- Configuration identification: Selection of configuration item (CI), defining required configuration document for every CI, giving ID numbers for technical documents and CIs.
- Configuration baseline: A baseline is a group of configuration items (products, deliverables) developed during a specific phase of the development process that has been formally accepted. Once the baseline is established, changes to the items can only be done through a formal change process. There are five baselines typically used in system development: functional, allocated, design, product, and operational. The functional baseline may contain an initial investigation baseline, a feasibility study baseline, and a requirements definition baseline.
- Configuration control (CC): Endorsing proposed engineering changes to UAV systems systematically and executing them fully coordinated and documented.
- Configuration control board (CCB): The technical and administrative board that makes decisions regarding whether or not engineering change proposals (ECPs) to an endorsed configuration document should be implemented.
- Configuration management plan: A managerial discipline applied to a UAV system for controlling and tracing functional and physical specifications during whole life cycle.

After giving the above definitions, it will be beneficial to depict the basic elements of configuration management. Elements of configuration management are shown in Fig. 108.7.

UAV systems have both air and ground assets. Also they have many electronic hardware parts, complicated and different kinds of software. As many other

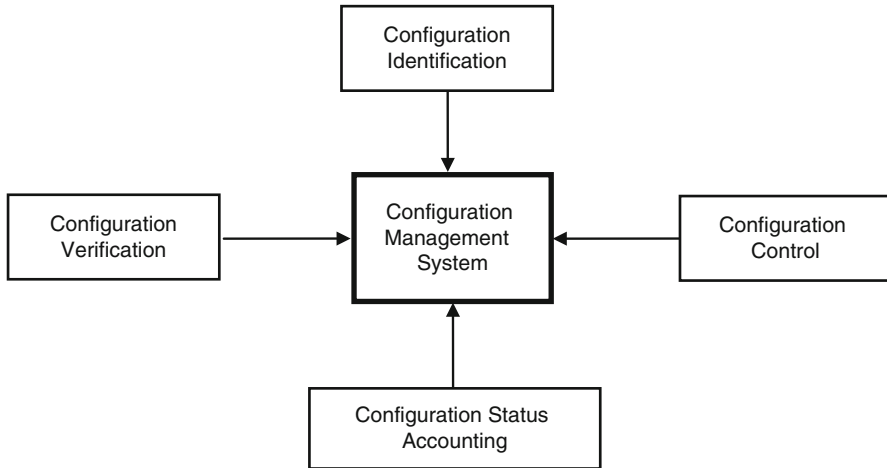


Fig. 108.7 Elements of configuration management system

complicated systems, configuration management is an essential assignment for an organization. Otherwise managing and maintaining a system during its life cycle is almost impossible. In fact configuration management system for UAVs is not different than any other complicated system. For this purpose the elements of configuration management system are given in general. In the following part the elements of configuration management system are given separately. By the way the elements will be elaborated in the following headings:

108.8.3.1 Configuration Identification

Configuration identification is the process of identifying the attributes that define every aspect of a configuration item. In order to fulfill configuration identification process, the following operation should be performed successively:

- Selection of configuration item must be realized
- A system must be established for disseminating configuration documents
- Selection of the documents used for developing configuration baseline must be done
- The interfaces must be identified and documented
- Functional, allocated, and product baselines must be composed
- Configuration item, configuration definition, series, and band numbers must be identified
- A correlation among branding, labeling, configuration documents, and the other data must be established

108.8.3.2 Configuration Control

Configuration control ensures that proposed engineering changes to configuration items are fully coordinated and documented. Configuration control is tailored during each life-cycle phase of a configuration item as follows:

- **Acquisition Phase:** During this phase configuration control will be applied to a new item's operational and functional requirement; during this phase configuration control will be applied to all documents which establish the functional baseline documentation. This phase represents the acquisition of a new device and as such establishes the product baseline documentation.
- **Operational Support Phase:** During this phase configuration control will be applied to any revisions to the established functional and product baseline documentation of an existing item. This phase represents the modification/update of existing devices.

An engineering change proposal is the standard method for proposing changes to a configuration item. Engineering change proposal is used to initiate engineering change proposals. There are two categories of engineering change proposals:

- **Solicited.** Solicited ECPs are submitted by the contractor in response to a written request made by the government. At NAWCTSD, it occurs as a result of a solicited ECP on a weapon system platform (WS/ECP) from the major claimant which, in turn, could affect the training device/system.
- **Unsolicited.** Unsolicited ECPs are those submitted without a formal written request. Such proposals are not encouraged and will be rejected unless they satisfy one or more of the following criteria:
 - Correct deficiencies.
 - Make significant reduction in manufacturing, operational, or logistics support costs.
 - Prevent slippage in production schedules.
 - Are a value engineering proposal.

Also ECPs are classified in two groups as Class I and Class II. These ECP classes can be explained as follows:

- **Class I engineering change proposals:** There are two types of Class I engineering change proposals.
 - **Preliminary engineering change proposal.** A preliminary engineering change proposal is used to determine if a formal engineering change proposal is justified. It is not reviewed by a change control board and cannot be used to authorize a change to a configuration item.
 - **Formal engineering change proposal.** A formally submitted engineering change proposal which has been engineered, documented, and priced in sufficient detail to support approval.
- **Class II engineering change proposals:** The ECPs that falling outside Class I ECPs are Class II ECPs. Generally unless otherwise specified by contract, Class II ECPs are not reviewed and approved. Only the contractor has the authority to approve the classification and implementation of the change as Class II.

108.8.3.3 Configuration Status Accounting (CSA)

Objective of CSA report is to provide visibility into the current status of configuration items whether part of a baseline or not. CSA reports also depict the changes made to base-lined configuration items. CSA is the process of creating and organizing the knowledge base necessary for the performance of configuration management. In addition to facilitating configuration management, the purpose of CSA is to provide a highly reliable source of configuration information to support all program/project activities including program management, systems engineering, manufacturing, software development and maintenance, logistic support, modification, and maintenance. CSA includes the following:

- A record of the approved configuration documentation and identification numbers
- The status of proposed changes, deviations, and waivers to the configuration
- The implementation status of approved changes
- The configuration of all units of the configuration item in the operational inventory
- Discrepancies from functional and physical configuration audits

108.8.3.4 Configuration Verification

The objective of verification is to detect and manage all exceptions to configuration policies, processes, and procedures. The configuration verification process includes the following:

- Configuration verification of the initial configuration of a CI and the incorporation of approved engineering changes, to assure that the CI meets its required performance and documented configuration requirements
- Configuration audit of configuration verification records and physical product to validate that a development program has achieved its performance requirements and configuration documentation or the system/CI being audited is consistent with the product meeting the requirements

108.9 Logistics at Disposal Operations Phase

The purpose to dispose a system from inventory is to redistribute, transfer, donate, sell, abandon, or destroy the system. In other words the disposal phase is the process for eliminating the system. At the end of the system's life cycle, the program manager must perform the actions defined in the disposal plan to properly remove the system from the inventory. At the end of its useful life, a system shall be disposed in accordance with all legal and regulatory requirements and policy relating to safety (including explosives safety), security, and the environment.

Before going further on this subject, let's look at the problem from another direction. If the problem is to define the end of life cycle of a system, the definition will predominantly depend on economical, political, and maybe forensic issues. The technical issues may not be as dominant as the other factors. Actually there

is no definitive end to a system. Systems normally evolve or transition to the next generation because of changing requirements or improvements in technology.

The disposal activities ensure the orderly termination of the system and preserve the vital information about the system so that some or all of the information may be reactivated in the future, if necessary. Particular emphasis is given to proper preservation of the data processed by the system so that the data is effectively migrated to another system or archived in accordance with applicable records management regulations and policies for potential future access. This point of view is interesting and may be taken into account in some cases. But in this study it is not the case. In the following explanations disposal subject proceeds without regarding information transfer between generations, and it is accepted that with some assumptions end-time definition of a system exists.

In this part disposition options of UAV system and the secondary market alternatives are examined. At the end of the sustainment period, a decision process is taken into account. This process interrogates the discarding method of a system from inventory. This method may be either return the UAV to vendor, to the landfill, or to the secondary market [23].

Actually disposal of a system is an inventory management problem and depends on economic concerns and cost-effective aims. The reasons of disposal or transfer to a secondary market may be listed as follows:

- UAV system does not meet the customers requirements anymore
- To repair the malfunction or defect of a UAV system isn't cost-effective
- The obsolete percentage of material requirement list on the system is relatively high and equivalent components are not available
- Maintaining a UAV system in operation is not cost-effective because the end of systems life cycle is reached

Especially for UAV systems malfunctions and defects are more encountered than piloted aircraft systems. This feature affects not only the maintenance procedures but also the disposal procedures. For that reason the flying part of UAV systems are generally disposed before their life cycle. On the other hand life cycle of ground assets of UAV systems is limited with obsolescence of subsystems. Disposing before life-cycle trend for flying part of UAV systems is more valid for small UAVs than larger ones. In fact disposing large UAVs before their life cycle is not so cost-effective. Nonetheless, repairing and replacing most parts of large UAVs and increasing the periodic maintenance frequency of them are applied methods for lasting the life cycle of UAVs longer.

108.9.1 Disposition Methodologies

Disposition of a system means remove the system from inventory of an organization. The organization may be military, governmental, or nongovernmental. Whole kinds of organizations have an inventory, and inventory removal methods are similar by some means or other. Disposition methods and rules of UAV systems are similar to any other airborne system; the only difference is additional ground assets. Disposition of a UAV system is an economical, political, and managerial decision,

and for some cases it can be a strategic decision. For removing a UAV system from inventory, there are basically three methods. These methods are listed as follows:

- *Method 1:* Remove the UAV system and whole its subsystems from inventory for good, and do not operate any reverse logistics processes:
- *Method 2:* Remove the UAV system from the inventory, but don't dispose the whole subsystems. Some subsystems are recycled to the system, in order to reuse in somewhere else in the organization.
- *Method 3:* Remove the UAV system from inventory by selling a secondary market.

All the methods have some advantages and some disadvantages in some ways. Methodology selection process is conducted by assessing many merits into account. Some of these merits are listed as follows:

- The condition of UAV system, in operation or failure
- Technology level of UAV system and its relative condition with the state-of-the-art technology
- Cost efficiency of maintenance
- Reusing case of UAV subsystems
- Operational requirement for the system
- Predetermined life cycle of the UAV system

108.9.2 Life-Cycle Cost Analysis

Disposal decision for a system is given at the beginning of the project with life-cycle cost (LCC) analysis. For this purpose LCC analysis is mentioned under disposal phase. Generally it is carried out in the planning part of a project, but it can be performed in all phases of life cycle. Also LCC analysis can be performed in during sustainment phase. By the way corrective calculations due to usage can be added into LCC analysis. If the LCC analysis is made at the beginning phase of a project, then the inputs of sustainment phase experiences may not be covered. As mentioned before the accident rate of UAV systems is higher than that of piloted aircraft. The source of accidents can be categorized as human, material, or environmental factors. Human causal factors relate to human errors. Material causes being the result of equipment failure and damage as a result of design flaws, component, or system failure such that the system becomes inoperable. Environmental factors include noise, illumination, and weather conditions [24]. If the LCC analysis is carried out in the beginning of the project, the undesirable conditions of UAV system during operations and usage may lead the organization to make the LCC analysis during sustainment phase again.

Before explaining LCC analysis, giving the definition for LCC will be beneficial. Fundamentally LCC is cumulative cost of a product over its life cycle. There are a lot of different definitions for LCC. One short and to the point of them is defined as, "LCC is the total cost of ownership of machinery and equipment, including its cost of acquisition, operation, maintenance, conversion, and/or decommission" [25].

LCC are summations of cost estimates from inception to disposal for both equipment and projects as determined by an analytical study and estimate of

total costs experienced in annual time increments during the project life with consideration for the time value of money. The objective of LCC analysis is to choose the most cost-effective approach from all possible alternatives to achieve the lowest long-term cost of ownership. LCC is an economic model over the project life span. Usually operation, maintenance, and disposal costs exceed all other first costs related with procurement. The best balance among cost elements is achieved when the total LCC is minimized [26].

LCC helps change provincial perspectives for business issues with emphasis on enhancing economic competitiveness by working for the lowest long-term cost of ownership which is not an easy answer to obtain. Consider these typical problems and conflicts observed in most organizations:

- Project engineering wants to minimize capital costs as the only criteria
- Maintenance engineering wants to minimize repair hours as the only criteria
- Production wants to maximize uptime hours as the only criteria
- Reliability engineering wants to avoid failures as the only criteria
- Accounting wants to maximize project net present value as the only criteria
- Shareholders want to increase stockholder wealth as the only criteria [27]

LCC analysis is a collective term comprising many kinds of analysis. These analyses cover reliability-availability-maintainability analysis, economic analysis, and risk analysis. A main objective of the LCC analysis is to quantify the total cost of ownership of a product throughout its full life cycle, which includes research and development, construction, operation and maintenance, and disposal. The predicted LCC is useful information for decision-making in purchasing a product, in optimizing design, in scheduling maintenance, or in planning revamping. LCC analysis may be applied for:

- Evaluation and comparison of alternative design
- Assessment of economic viability of projects/products
- Identification of cost drivers and cost-effective improvements
- Evaluation and comparison of alternative strategies for product use, operation, test, inspection, maintenance, etc.
- Evaluation and comparison of different approaches for replacement, rehabilitation/life extension, or disposal of aging facilities
- Optimal allocation of available funds to activities in a process for product development/improvement
- Assessment of product assurance criteria through verification tests and their trade-offs – long-term financial planning [28]

An LCC analysis procedure has some essential subprocesses. To be mentioned these subprocesses will be helpful for understanding LCC analysis. These subprocesses may be summarized as six basic processes as follows:

- Problems definition
- Cost elements definition
- System modeling
- Data collection
- Cost profile development
- Evaluation [29]

108.10 Future UAV Logistics Trends

108.10.1 Airborne Logistics Support

UAVs will fill in the physical gap between satellites and the manned aircraft in the near future. Some sort of UAVs will access to the stratosphere and stay aloft for weeks, months, or even years without landing. They would accomplish the missions of the satellites such as TV broadcast, and radio communication with more reasonable costs. These UAVs should be designed to tolerate the special environmental conditions in the stratosphere and have the ability to serve for a long period of time without any important intervals requiring any logistics support on the ground. Thus, such UAVs required staying aloft for a long period of time; airborne logistics support will be needed.

Airborne logistics support will be done manned or unmanned aircraft which will have necessary equipment to diagnose the malfunctions and to do periodic tests. They also have the capability to carry enough supply materials such as fuel, oil, and lubricants. How the maintenance will be done will depend on the type of the maintenance requirement. If periodic maintenance such as supplying fuel or doing periodic tests is required, it would be done automatically by the technicians on the ground. But for the malfunctions requiring the replacement of parts, the technicians should be on the airborne logistics support vehicle with required spares.

108.10.2 Power Refueling by Laser

Apart from some solar-powered high-altitude UAVs currently under development, UAVs have to land to have their batteries recharged or fuel cells filled. There are currently some experiments to develop economically viable technology to use lasers to remotely power motors and recharge batteries while UAVs are flying. The concept under development works by turning electricity into laser light which is beamed to a remote receiver on the UAV. This converts the light back into electricity where it can be used to drive a motor or power a battery or payload. If this laser technology matures enough, some type of UAVs can fly and do mission much longer than today.

References

1. RAND Project Air Force (PAF), Balancing rapid acquisition of Unmanned Aerial Vehicles with support considerations, 2005, http://www.rand.org/pubs/research_briefs/2005/RAND_RB176.pdf. Accessed 11 Feb 2012
2. U.S. Air Force, *Unmanned Aircraft Systems Flight Plan 2009–2047*, Headquarters, United States Air Force, Washington D.C., 18 May 2009
3. U.S. Army UAS Center of Excellence, U.S. Army Roadmap for UAS 2010–2035, The Development, Concepts and Doctrine Centre, Fort Rucker, Alabama
4. U.S. Marine Corps, MCWP 3-21.2 Aviation Logistics, Marine Corps Combat Development Command, PCN: 14300010200, 21 Oct 2002

5. L.A. Ingham, T. Jones, A. Maneschijn, Considerations for UAV design and operation in South African airspace. *Aeronautical Journal* **110**, 695–701 (2006)
6. Defense Acquisition University, Life Cycle Logistics, <https://acc.dau.mil/CommunityBrowser.aspx?id=17616&lang=en-US>. Accessed 11 Feb 2012
7. UK MoD, Defence Standard 00-600 Issue 1: Integrated Logistic Support Requirements for MOD Projects, UK Defence Standardization, 23 Apr 2010
8. UK MoD, JSP 886 The Defence Logistics Support Chain Manual, Volume 7 Part 1, Version 2.4, 27 Nov 2013
9. NATO Joint Air Power Competence Centre, Strategic Concept of Employment for Unmanned Aircraft Systems in NATO, 4 Jan 2010
10. UK MoD, Joint Doctrine Note 2/11 the UK Approach to Unmanned Aircraft Systems, UK MoD The Development, Concepts and Doctrine Centre, SWINDON, Wiltshire, 30 Mar 2011
11. U.S. Army, Regulation 700–127 Integrated Logistics Support, Headquarters, Department of the Army Washington, DC, 17 July 2008
12. U.S. DoD, MIL-STD-40051 Technical Manual Preparation, USAMC, Redstone Arsenal, AL, 31 July 1996
13. ICAO, Annex 6 Operation of Aircraft, ICAO Headquarters, Montreal, Canada, July 2001
14. NATO, *RTO-AG-300-V20 Logistics Test and Evaluation in Flight Test* (NATO, Neuilly-sur-Seine, 2001)
15. B.D. Coryell, *Performance-Based Logistics, Contractor Logistics Support, and Stryker* (U.S. Army Command and General Staff College, Fort Belvoir, 2007)
16. Defense Acquisition University, *Performance Based Logistics: A Program Manager's Product Support Guide* (Defense Acquisition University Press, Fort Belvoir, 2005)
17. U.S. DoD OUSD (AT&L) Memorandum, Performance Based Logistics: Purchasing Using Performance Based Criteria, U.S. DoD The Under Secretary of Defence, Washington, DC, 16 Aug 2004
18. K. Vitasek, J. Cothran, S. Geary, S. Rutner, *Performance-Based Logistics: the Changing Landscape in Support Contracting*, University of Tennessee and Supply Chain Visions' White Paper, June 2006
19. Reg Austin, *Unmanned Aircraft Systems: UAVS Design, Development and Deployment* (John Wiley & Sons Ltd, West Sussex, United Kingdom, 2010)
20. A. Hobbs, S.R. Herwitz, Human Challenges in the Maintenance of Unmanned Aircraft Systems, FAA and NASA Report (2006)
21. J.S. Gansler, W. Lucyshyn, L.H. Harrington, A.C. Corl, *The Current State of Performance Based Logistics and Public-Private Partnerships for Depot-Level Maintenance: Operating Models, Outcomes, and Issues*, The Center for Public Policy and Private Enterprise, Oct 2010
22. MIL-HDBK-61A, Configuration Management Guidance, U.S. DoD Under Secretary of Defense (Acquisition, Technology, and Logistics), Washington, DC (2001)
23. D.S. Rogers, R.S. Tibben-Lembke, *Going Backwards: Reverse Logistics Trends and Practices* (University of Nevada/Center for Logistics Management, Reno, 1998)
24. C.J. Hodson, Civil Airworthiness for a UAV Control Station, The University of York, Department of Computer Science, North Yorkshire, United Kingdom (2008)
25. Society of Automotive Engineers (SAE), *Reliability and Maintainability Guideline for Manufacturing Machinery and Equipment* (Society of Automotive Engineers (SAE), Warrendale, 1999)
26. R.R. Landers, *Product Assurance Dictionary* (Marlton Press, Marlton, 1996)
27. H.P. Barringer, *A Life Cycle Cost Summary* (2003)
28. Y. Kawauchi, M. Rausand, *Life Cycle Cost (LCC) Analysis in Oil And Chemical Process Industries* (Department of Production and Quality Engineering/Norwegian University of Science and Technology, Trondheim, 1999)
29. IEC60300-3-3, Life cycle costing, International Electrotechnical Commission, Geneva, Switzerland (2004)

Section XX

UAV Applications

Aníbal Ollero

Kimon P. Valavanis and George J. Vachtsevanos

UAV Applications offers a “taste” of possible civil and public domain applications in which single or multiple UAVs may be used. As the roadmap of the UAS integration into the global airspace is formulated amidst current obstacles and skepticism, it is only logical to consider that the spectrum of possible applications will be much wider in the years to come. At the present stage, UAVs have either been considered for or have seen (very) limited utilization in a wide range of applications that include but are not limited to power line inspection; pipeline inspection; ship inspection; mine inspection; dam inspection; anomaly detection/prevention; early fire detection and forest protection; hazard monitoring; traffic monitoring; environmental monitoring; search and rescue operations; emergency response; border patrol; harbor patrol; police surveillance; aerial photography; SWAT support; imaging and mapping; intelligence, surveillance, and reconnaissance (ISR); chemical spraying; crop dusting; night vision; and entertainment industry and filming.

However, in addition to enabling and mature technologies, UAV utilization will also be affected by cost considerations, certification restrictions (which may or may not be application specific), and supply and demand.

► **Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring** by Kanis-tras, Martins, Rutherford, and Valavanis focuses on surveying UAV-based systems for traffic monitoring and management. Although there is voluminous research on the subject, UAVs are proven to be a viable and less time-consuming alternative to

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

real-time traffic monitoring and management as they provide the dynamic “eye-in-the-sky” solution to the problem.

► **Measurement and Exploration in Volcanic Environments** by Melita, Longo, Muscato, and Giudice presents a challenging application for UAVs in volcanic environments, as they may be used to collect typical aerial measurements and data. The main forms of measurements concern the collection of visual/thermal images and gas analysis and sampling. An overview of the major worldwide projects carried out in the last few years by several research organizations is summarized and discussed.

► **Cooperative Unmanned Aerial Systems for Fire Detection, Monitoring, and Extinguishing** by Merino, Martınez-de Dios, and Ollero deals with the application of cooperative UAVs to forest fires. It presents a decision and control architecture for multi-UAS teams that are equipped with diverse sensors (e.g., infrared and visual cameras) and details the perception techniques (fire segmentation, geo-localization, and data fusion) that are used toward fire extinguishing.

► **Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States** by Ambrosia and Zajkowski describes some of the UAS lessons learned by both the U.S. Forest Service (USFS) and the National Aeronautics and Space Administration (NASA) during their more than 10-year efforts on identifying appropriate UAS wildfire observation capabilities. The agencies, along with the UAS community, have demonstrated the use of both low-altitude, short-endurance (LASE) and medium-altitude, long-endurance (MALE) platforms for wildfire sensor intelligence gathering. Those experiences are identified along with an examination of the future direction of UAS for supporting the wildfire management community.

► **Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation** by Kondak, Ollero, Maza, Krieger, Albu-Schaeffer, Schwarzbach, and Laiacker discusses the novel and very challenging applications of using UAVs for aerial manipulation and load transportation. A specific application of using autonomous helicopters equipped with robotic arms for aerial manipulation is detailed, along with field experiments demonstrating joint load transportation with multiple autonomous helicopters.

► **Unmanned Aircraft Systems for Maritime Operations** by Borges de Sousa, McGuillivary, Bento, Morgado, Matos, Gomes Bencatel, and de Oliveira discusses current practices and major UAS trends for maritime operations, along with projections of future UAS maritime applications. Various aspects of UAS program management, including organization, operator responsibilities, program operation, and overall program costs are reviewed, followed by maritime aircraft missions and required capabilities. Technological trends are highlighted focusing on developments relevant to maritime UAS operations. Trends including miniaturization of sensors and computer systems, high energy density of power sources, and increased subsystem standardization and modularity will have important effects in the future, including increased system autonomy via new command and control frameworks.

► **Autonomous Remote Sensing of Invasive Species from Robotic Aircraft** by Goktogan and Sukkarieh presents an ensemble system consisting of a helicopter,

a fixed-wing UAV, and their supporting infrastructure that is used in a number of environmental research experiments focused on autonomous remote sensing, detection, classification, and management of invasive species, weeds, in Australia. In these experiments, three distinct families of weeds on three different terrains are investigated. In the first group of experiments, a helicopter UAV equipped with a high-resolution imaging payload is flown over difficult to reach water channels and wetlands for detection of aquatic weeds. The second set of experiments is performed in relatively flat rangelands to map woody infestations. The third set of experiments is focused on the airborne detection of wheel cacti on remote mountainous terrain using fixed-wing aircrafts. Successful results of these experiments suggest that robotic aircrafts in a properly designed UAS can play an important role in environmental robotic science.

► **Cyber-Physical Systems Enabled by Small Unmanned Aerial Vehicles** by Coopmans, Stark, Jensen, Chen, and McKee provides several examples of unmanned aircraft sensing-enabled cyber-physical system (CPS) scenarios, enabling adaptive management and effective control of complex physical systems such as water distribution based on measurement of soil moisture and crop evapotranspiration, radio tag-based tracking of fish, alternative energy harvesting, mapping of invasive plant species, and airborne plume (pollution) tracking. This chapter serves as a motivator for potential wide-spreading UAS applications where UAS are used as mobile sensors and/or mobile actuators in large-scale, closed-loop CPS.

Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring

110

Konstantinos Kanistras, Goncalo Martins, Matthew J. Rutherford,
and Kimon P. Valavanis

Contents

| | | |
|----------|--|------|
| 110.1 | Introduction | 2644 |
| 110.2 | Existing Surveys in Traffic Monitoring | 2646 |
| 110.3 | Current Research Work | 2646 |
| 110.3.1 | Airborne Road Traffic Monitoring with Radar | 2646 |
| 110.3.2 | Real Time Road Detection by Learning from One Example | 2648 |
| 110.3.3 | Statistical Profile Generation for Traffic Monitoring Using Real-Time UAV Based Video Data | 2649 |
| 110.3.4 | Vehicle Detection from Aerial Imagery | 2650 |
| 110.3.5 | An Improved Object Tracking Method in UAV Videos | 2652 |
| 110.3.6 | Visual MTI for UAV Systems | 2653 |
| 110.3.7 | Real-Time Video Relay for UAV Traffic Surveillance Systems Through Available Communication Networks | 2653 |
| 110.3.8 | Roadway Traffic Monitoring from an Unmanned Aerial Vehicle | 2654 |
| 110.3.9 | Detecting and Counting Vehicles from Small Low-Cost UAV Images | 2655 |
| 110.3.10 | A Vision Algorithm for Dynamic Detection of Moving Vehicles with a UAV | 2655 |
| 110.3.11 | Supporting Traffic Flow Management with High-Definition Imagery | 2656 |
| 110.3.12 | Robust Data Alignment Based on Information Theory and Its Applications in Road-Following Situation | 2657 |
| 110.3.13 | From Images to Traffic Behavior: A UAV Tracking and Monitoring Application | 2657 |

K. Kanistras (✉) • G. Martins

Department of Electrical and Computer Engineering, University of Denver, Denver, CO, USA
e-mail: Konstantinos.Kanistras@du.edu; Goncalo.Martins@du.edu

M.J. Rutherford

Department of Computer Science, University of Denver, Denver, CO, USA
e-mail: mjr@cs.du.edu; Matthew.Rutherford@du.edu

K.P. Valavanis

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel
Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver,
CO, USA
e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

110.3.14 The Use of Small Unmanned Aircraft by Washington State
Department of Transportation 2658

110.3.15 Not Yet Implemented on UAVs 2660

110.4 Conclusion 2664

References 2665

Abstract

The focus of this chapter is on surveying UAV-based systems for traffic monitoring and management. Although there has been voluminous research on the subject, unmanned aerial vehicles (UAVs) are proven to be a viable and less time-consuming alternative to real-time traffic monitoring and management, providing the solution to the problem.

110.1 Introduction

The continuous increase of vehicles on roadways forces transportation managers to look for more effective ways to reduce the traffic congestion problem. Because of the increasing traffic volume, there is a demand for state-of-the-art intelligent traffic information for real-time traffic monitoring and management. This requires collection of accurate information, for not only the state of the traffic and roadway conditions but also real-time information in case of emergencies like car accidents that lead to traffic congestion and/or rerouting (Puri 2005; Puri et al. 2007).

Research is being conducted on this topic and several methods proposed to investigate what is the most effective way of transmitting and analyzing traffic data acquired from (UAVs) (Bethke et al. 2007; Kim 2005; Chen et al. 2007). Research is also focused on the type of data that should be collected and the information that should be extracted to design traffic simulation models and evaluate traffic networks (Puri et al. 2007; Coifman et al. 2006; McCormack 2008; Heintz et al. 2007). UAVs are preferred over the traditional technologies (inductive loop detectors, video, radar, ultrasound technologies, etc.) because of their mobility and because the cost of the operation is considerably lower than manned systems. The ability to hover, and fly above highways, allows the UAV to provide information such as vehicle trajectories and routing information which is not possible to be provided from detectors that are positioned in certain areas along the highway. Visual data are collected in real time by the unmanned systems, and they are used to generate traffic-related statistical profiles, serving as inputs to traffic simulation models (Puri 2008) as Fig. 110.1 shows.

This survey presents current research in the area of UAVs for traffic monitoring and management. A search on related research using UAVs for traffic monitoring and algorithms that could be implemented on UAVs for that purpose since 2005 was performed. Journal and conference papers related to the subject have been considered. The databases and keywords used to complete the survey are given in Table 110.1.

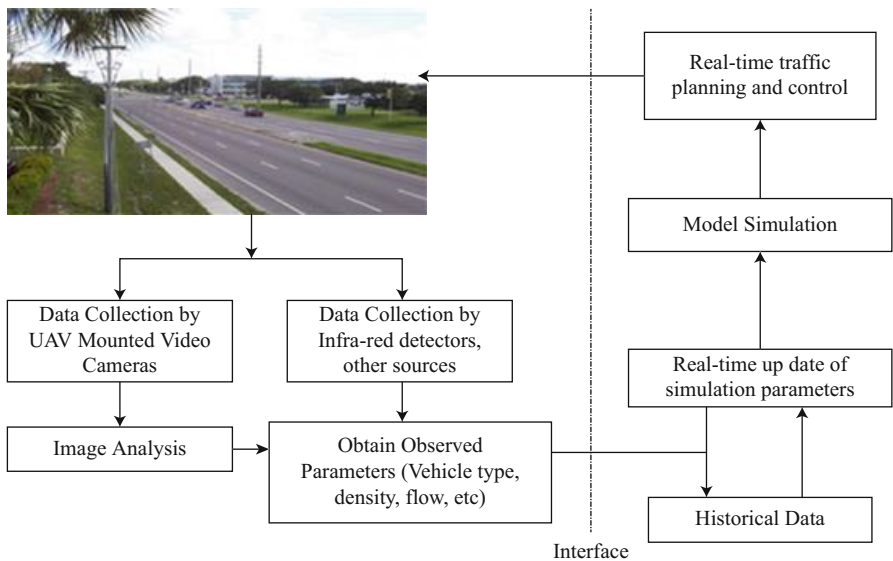


Fig. 110.1 Traffic monitoring: framework for incorporating real-time data in simulation models

Table 110.1 Search process

| | |
|----------------|---|
| Databases | IEEE Xplore Web of Science Engineering Village Engineering Index ENGnetBASE ScienceDirect Google Scholar |
| Keywords | UAV and traffic monitoring algorithms UAV and traffic monitoring UAV traffic surveillance Road tracking UAV and road tracking Unmanned systems and traffic monitoring Survey and UAVs and tracking monitoring |
| Date of search | August 2012 |

The rest of the chapter is organized as follows: Sect. 110.2 discusses comprehensive surveys in traffic surveillance and monitoring, while Sect. 110.3 summarizes current research that is being done in the area since 2007 when the last survey related to traffic monitoring was conducted. They are classified in three subsections mentioning universities and research centers with projects on UAVs that could be used for traffic monitoring implementing some tests up to 2007 (Puri 2005; Ro et al. 2007), universities that implemented tests with UAVs for traffic monitoring from

2007 up to today, and others claiming that those systems or algorithms could be implemented on UAVs for the same purpose. Section 110.4 concludes the chapter.

110.2 Existing Surveys in Traffic Monitoring

After surveying the literature, two comprehensive surveys in traffic monitoring were found. In Puri (2005) the author provides a technical report of research related to the application of UAVs for traffic management. The advantages of using UAV platforms over the traditional technologies of traffic sensing are pointed out. A wide list of UAV platforms used for traffic monitoring is also proposed.

Table 110.2, copied from Puri (2005), summarizes all research done up to 2005 at several universities around the world in the area of traffic surveillance with UAV platforms. It presents, per university, the performed research, the platforms that were used for the experiments, and the objectives of each research project.

In Ro et al. (2007), applications of small UAVs for urban highway traffic monitoring related to a specific project were conducted at Western Michigan University. In this report, FAA regulations and standards for UAVs are described in detail. Also, the classification of airspace and entry requirements are summarized and illustrated. Finally, a list of suitable network protocols, which are standards of networking, is also provided.

110.3 Current Research Work

Real time is a difficult task for a UAV to accomplish not only due to the high level of complexity of the aerial images due to the need for full 3D control. Following a road implies that the UAV should be able to *see* it first; therefore there is no doubt that rich information could be provided by vision and radar sensors. However, processing that information is difficult especially in real time (Kim 2005). Ongoing research is focused on the type of sensors used onboard and on the way that data are processed (onboard or off board) for traffic monitoring.

110.3.1 Airborne Road Traffic Monitoring with Radar

This paper Bethke et al. (2007) introduces a possible solution to traffic monitoring with radars and sensors on unmanned systems. According to the authors, travel time is an important measure for characterizing traffic flow. Only high-flying platforms offer the possibility of wide area coverage of infrastructure as well as of the road traffic situation without the need of additional equipment. SAR (Synthetic Aperture Radar) processors provide two-dimensional radar images from the earth surface and are usually suitable for monitoring stationary scenes. GMTI (Ground Moving Target Indication), which is the detection of moving targets against a stationary background, gives the ability to calculate major parameters for estimating the traffic

Table 110.2 Research team (up to 2005), UAV prototype, research objectives, and application stage (Puri 2005)

| Team | System (UAV) | Objective |
|--|---|---|
| University of Florida | Aerosonde | Use of UAV with video for data collection for traffic surveillance |
| Florida DOT Tallahassee Commercial Airport | (Fixed wing) Payload 5 lb | Define communication links using current FDOT microwave systems Timely information on transportation networks – both rural and urban |
| RWIS Research Team | Sony XC 555 video camera | |
| Linköping University, Sweden | Scandicraft Apid MK 3 (rotary wing) Payload 20 kg | Develop technologies for deployment of fully autonomous UAV Integrate autonomy with active vision system |
| | Yamaha RMAX (rotary) Payload 30 kg | Identifying complex patterns of behavior (vehicle overtaking, etc.) |
| Ohio State University | MLB BAT3 | Learning, discovering, and developing potential benefits of UAV applications to transportation surveillance |
| Ohio DOT NCRST | (Fixed Wing) Payload 5 lb | Collecting information on freeway conditions, intersection movements, network paths, and parking lot monitoring |
| Georgia Tech Research Institute Georgia DOT | Customized drone (rotary wing) | Development of generic VTOL, advanced controllers |
| Federal Highway Administration's Priority Technology Program | | Fault-tolerant and autonomous operation algorithms Achieve dynamic performance and flight control command generation |
| University of California, Berkeley Office of Naval Research | Sig. Rascal Radio controlled Airplane (fixed wing) | Intelligent guidance systems for UAV Strategies of path-planning |
| AINS Program | | Augment GPS with machine vision |
| LAAS | MARVIN (rotary) | Design and implementation of a distributed control system for cooperative detection and monitoring using heterogeneous UAVs |
| CNRS | Karma Blimp (fixed) | |
| Real-time Systems and Robotics | Remotely piloted helicopter (rotary) | Control architecture and technique of real-time control |
| ADAI | | Integrating distributed sensing techniques with real-time imaging |
| CVL HELIV | | |

(continued)

Table 110.2 (Continued)

| Team | System (UAV) | Objective |
|---|--|--|
| USDOT's RSPA NASA Bridgewater State College University of Massachusetts MLB Company | MLB BAT 3 (fixed wing) Payload 5 lb | Autonomous UAV to collect and interpret real-time traffic data Gather multimodal data using road-following capabilities |
| NCRST Virginia DOT | ADAS system Payload 20 lb | Traffic surveillance using ADAS |
| ARL, Stanford University | Customized UAV (rotary) Differential carrier-phase GPS | Demonstrate practicality of using unmanned robot helicopters Demonstrate feasibility of using carrier-phase GPS as sensor |
| Carnegie Mellon University | Customized UVA (rotary) | Develop a vision-based robot helicopter which can operate autonomously to carry out well-structured mission goals |
| Eidgenossische Technische Hochschule (ETH) Zurich | Unmanned helicopter, fixed wing, airships | Flight control, integrated navigation, mission planning |
| Universitat Stuttgart Aerobot | (Airship) | Investigate strategies for guidance, navigation, and control of UAV and MAV |

flow with radars. At the German Aerospace Center (DLR), an operational, compact SAR system for high-flying airborne platforms is planned to be developed over the medium term, and it will be used for traffic management.

Advantages – Using radars on high-flying platforms allows not only the capture of data in wide areas regardless of the weather conditions but also the detection of vehicles independently of their instrumentation onboard (Bethke et al. 2007). The resolution of SAR images is on the order of 1–2 m (Bethke et al. 2007).

Disadvantages – Techniques to reduce the amount of data that must be processed and transmitted are not yet developed. Also techniques to detect vehicles with precision are under investigation but not yet implemented.

110.3.2 Real Time Road Detection by Learning from One Example

At the University of California at Berkeley, an algorithm for real-time road detection is presented. Additionally, an algorithm capable of detecting any type of road (or corridor/quasi-linear structure) is proposed (Kim 2005). According to the author, this is a new approach to detect roads in real time by processing a single image of a target road. On that image the road structure is represented as a one-dimensional intensity profile, and the detection is done by matching each horizontal scan line of

a target image. Once the road positions for each line are found, a fitting algorithm can be used to find the final road position.

The Random Sample Consensus algorithm is applied for robust fitting which has been used for computer vision problems (Fischler and Bolles 1981). The purpose is to calculate vanishing points from pairs of line segments and rectify the image in order to detect the road boundaries and then define the road structure. The results show that the algorithm successfully learns the road structure from their first frames and applies that to the rest of the video. No visible error is detected in any single frame except when the road structures are drastically changed due to an entrance ramp.

Advantages – Real-time computation for road detection in urban environments is an important application. The real-time constraint related with the search algorithm can be overcome by reducing the vertical sampling rate. The same algorithm could be applied to a non-straight road with the assumption that it is partially straight. Also, the algorithm could be applied to other applications and not necessarily to road detection.

Disadvantages – It is not adequate to apply algorithms to detect lane markings unless the UAV flies at a low altitude to make the lane markings visible. The position errors of the hypotheses do not follow a Gaussian distribution; thus least-square line fitting gives inaccurate fitting results (Kim 2005). Real-time adaptation for a changing road structure and curved road detection is hard to accomplish.

110.3.3 Statistical Profile Generation for Traffic Monitoring Using Real-Time UAV Based Video Data

The eye-in-the-sky project is proposed from the University of South Florida, where small unmanned helicopters were used to monitor traffic, evaluate and assess traffic patterns, and provide accurate vehicle counts. The purpose of this research is to develop a system, which will optimize the traffic network, to improve the performance and flow of traffic at both normal and peak hours. Each helicopter is equipped with custom-made vision systems for onboard and on-the-ground data processing and pan-tilt cameras for dynamic tracking and car-following. Data are collected and traffic parameters such as occupancy, capacity flow, and density are calculated and modified based on those data. Traffic parameters are generated and fed to a simulation model which will be used in the future for modeling traffic conditions. Also the ratio of the vehicles that turn left or right or continue through the intersection is necessary for traffic management, and this could be calculated by tracking every individual vehicle.

Density is calculated using a Virtual Detection Frame (VDF) as shown in Fig. 110.2. This method uses a single still image at one time and calculates the vehicles present inside the VDF (Puri 2008). A method to calculate the delay, which is an important component of traffic behavior, due to the fact that minimum delay in a network implies a free flow network, is also mentioned in the paper. The simulation model gave promising results.

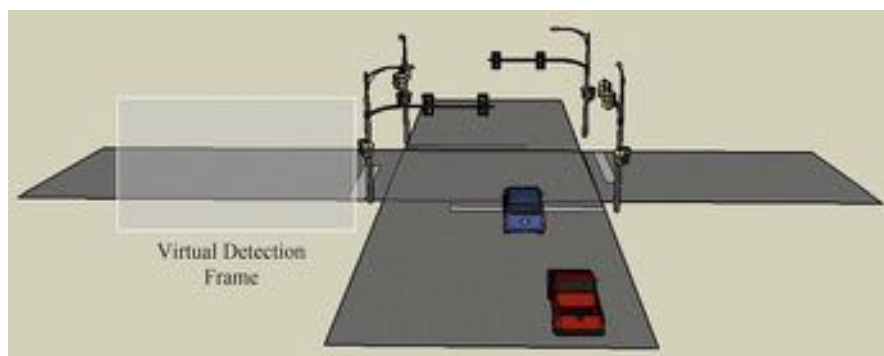


Fig. 110.2 Virtual Detection Frame (Puri 2008)

The proposed and utilized conceptual framework is shown in Fig. 110.3. The real-world traffic network block depicts the physical traffic network. The bold outline blocks depict the utilization of unmanned helicopters and the onboard/on-the-ground image processing system. Data is collected in real time by cameras mounted on the UAV.

Virtual detectors are implemented as colored boxes in the image processing algorithm as Fig. 110.4 depicts. Each lane has its own corresponding box, which is responsible for detecting every vehicle passing through it. The boxes can be either automatically placed after the system recognizes the road environment or the experimenter can manually place the box to the desired detection region (Puri 2008).

Advantages – A simulating model based on real-time, dynamic, and traffic network data improves the prediction of the traffic behavior in real time. The vision system, which has onboard and on-the-ground processing capabilities, contains algorithms which give the ability to adapt to different environmental setups automatically.

Disadvantages – The helicopter's instability leads to a vehicle count error. All data are collected from a fixed point. Also data were not collected while the helicopter was moving (Puri et al. 2007).

110.3.4 Vehicle Detection from Aerial Imagery

This paper (Gleason et al. 2011) is focused on feature extraction methods and classification techniques for automatic vehicle detection by UAVs on traffic control and other applications. The proposed method has two stages wherein in the first stage the algorithm is *looking* for man-made objects, which in this case are vehicles, and rejects most of the background. This stage has four steps:

- (i) Feature detection: Image features based on edges detection. Harris corner detection is used.

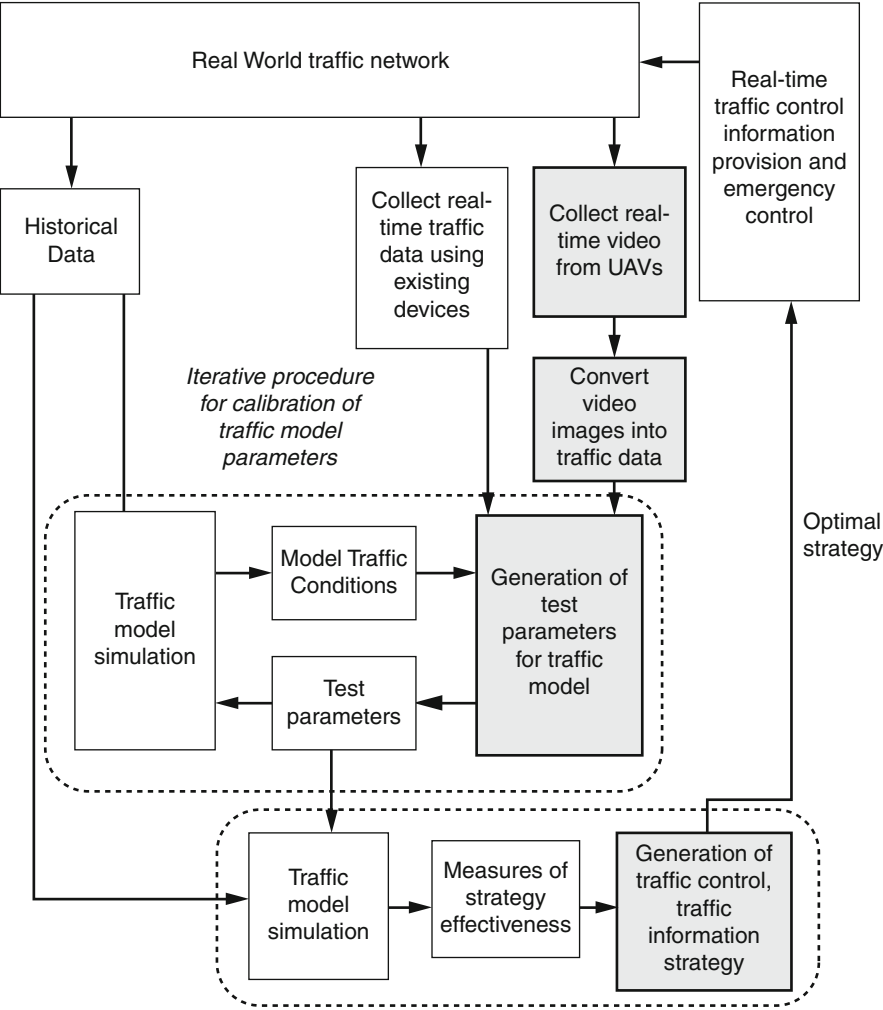


Fig. 110.3 Conceptual framework of real-time data collection and traffic simulation control (Puri 2008)

- (ii) Feature density estimation: The algorithm determines areas with feature density higher than a fixed threshold.
- (iii) Target clustering: The overlapping responses are grouped together and the overlapping detection is rejected.
- (iv) Color-based detection refinement: Color information is used to reduce the false alarms.

The target classification is implemented during the second stage, where further reduction of the false alarm rate is done. Two feature extraction methods (Histogram of Oriented Gradients, Histogram of Gabor Coefficients) and three

Fig. 110.4 *Colored boxes* show implementation of virtual detectors in the image processing technique (Puri 2008)



different classification techniques (k-nearest neighbors (k-NN) (Cover and Hart 1967), random forests (RF) (Breiman 2001), and support vector machines (SVM) (Hearst et al. 1998)) are used and explained in detail in this paper.

Advantages – A reasonably accurate estimate of the overall accuracy of the program was achieved by using the threshold of the fast detection algorithm. The performance of Gabor features under different lighting conditions was good, due to its ability to retain nearly all of the vehicles in the set while successfully reducing the false positives by half.

Disadvantages – The rotation invariant HoG did not perform well due to the reflective water that was in the background.

110.3.5 An Improved Object Tracking Method in UAV Videos

In Fang et al. (2011) Mean Shift tracking algorithms are studied because, according to the authors, those algorithms have problems tracking objects either whose size is changing or which are moving fast. The authors claim that the proposed filtering-based Mean Shift algorithm solves both problems. To overcome the tracking problem, the building of object models according to known target appearance and possible areas that may appear in it is proposed. During the matching process which is important for tracking purposes, it is suggested that the object's bounding box should be fixed so the probabilistic density function value can be calculated afterwards. When the algorithm may converge into a region of background if it is applied on UAVs because the objects are small and move rapidly. To reduce the mistakes in the Mean Shift algorithm, particle filtering is suggested. According to the authors, the along with particle filtering can solve the problems caused by fast-moving objects in UAV videos.

Advantages – Optimizing the suggested algorithm and combining Mean Shift and particle filtering to adapt to changing object size. Also, small objects are rarely lost which makes this algorithm valuable in UAV videos to track small fast-moving objects like vehicles in highways.

Disadvantages – Only when the regions are known, the Mean Shift can track the objects on time (Fang et al. 2011).

110.3.6 Visual MTI for UAV Systems

In Evans and Turkbeyler (2007) focus is given on the problem of detecting moving objects on the ground with video cameras that are mounted on unmanned air vehicles. The main approach is to compare pairs of image frames that are captured a short time apart. This paper provides an approach which is not based on image registration but on the analysis of the motion of the features with respect to a static scene model. The feature analysis is divided in three stages. First, a pair of image frames is used to locate the fixed objects. A method based on random consensus RANSAC is used to calculate a matrix with the largest number of matched features. Then, during the second stage, the objective is to obtain an improved stationary and moving classification. A temporal analysis is done on results from many frame pairs in order to identify the set of features that are judged to be moving or stationary with high confidence. In the last stage, a process for further improvement of the performance is used for the remaining that are caused by the missing targets. The algorithm is evaluated on frames of a motorway roundabout and a car park of a supermarket.

Advantages – The algorithm performs well on the datasets from the supermarket even with complex 3D structures. That scene is challenging because pedestrians are walking between the parked cars and the algorithm is still able to detect the cars.

Disadvantage – False alarms and missing targets remain even after stage 2; thus some clearly moving objects are not detected. That is because the algorithm is blind to motion in some directions. The geometrical interpretation of the scene is not well constrained, and that causes problems even with a smooth traffic flow on the motorway. Finally, additionally problem will be caused if the camera is not fixed onboard.

110.3.7 Real-Time Video Relay for UAV Traffic Surveillance Systems Through Available Communication Networks

At Western Michigan University, a project for real-time video relay for UAV traffic surveillance is conducted (Chen et al. 2007). According to the authors, the challenging part of the project is that the ground control station is located near a highway and the surveillance videos collected need to be relayed to the office at the Michigan Department of Transportation in a timely manner.

The task of relaying is focused on how to deliver the data from the ground station to the control office, and for that purpose the public Internet was used. The ground

control laptop informed the end user computer of a dynamic IP address which is generated each time that a connection occurred, and the end user computers could open a media player to retrieve the video stream. Those videos in their original state are too big to deliver within available network bandwidth; thus compression is needed. The Windows Media Encoder is used to convert and compress media content into a traversal network media format. During the field experiments, data were collected and a comparison between two transmission schemes in different image resolutions in real-time distributions was conducted.

Advantages – More than one user computer can receive the video stream simultaneously with acceptable data rate. Experimental results showed that the proposed method can deliver quality videos to the control office in real time.

Disadvantage – The bandwidth of the cellular network influences the image of the video stream. During the field experiments, the video stream was frozen or buffered while the end users were watching the video. The results showed that the transmission with no server system scheme could only provide a video stream in continuous and smooth image in a bit rate of 45 Kbps and also showed that any rate below that may not meet the requirement of the UAV project.

110.3.8 Roadway Traffic Monitoring from an Unmanned Aerial Vehicle

Another project to investigate and calculate traffic parameters was conducted at the University of Ohio State University using the BAT III UAV technology (Coifman et al. 2006). The unmanned system is carrying a payload of two cameras at a speed of 50 km/h while transmitting video images to the ground station in real time. The purpose of the experiment is to measure the level of service, estimate the average annual daily travel, examine the operation of the intersection and the flow, and measure parking lot utilization.

Two methods of estimating roadway density and flow are developed for this project. The first method uses still frames, and the second method obtains information from a series of frames. The first method is straightforward to apply, but using individual frames is problematic because each frame represents a very small sample size (Coifman et al. 2006). GIS is used to measure the length of the roadway accurately in order to calculate the hourly volume. However, to calculate the speed from a single still frame, it is not possible, but it can be estimated either from the posted speed limit or from the video stream. In addition to the length and the speed, the number of passenger cars and trucks is needed to calculate the density according to the authors. By using the second method, the UAV flies over the traffic and the sensor observes conditions over a longer time and distance.

Advantages – The platoon matching method is proposed, where network OD flow is been calculated in unobserved regions. Since time and distance are known, it is only important to differentiate between entering and leaving the region, without tracking the vehicles the whole time. All the applications were demonstrated from data that were collected from less than 2 h of flight time.

Disadvantages – The field of view of the UAV is not wide and the fluctuations in the traffic cause large variations in density measured from a single still frame. Also the field of view continuously changes, and the sample size is small in order to calculate density and flow from individual frames. The data analysis and reduction were made manually and not automated.

110.3.9 Detecting and Counting Vehicles from Small Low-Cost UAV Images

From the School of Geosciences and Geomatics, East China Institute of Technology, comes another paper for detecting and tracking vehicles with UAV systems (Cheng et al. 2009). The main goal of this project is to develop techniques for generating high-resolution video flow from UAV videos. The UAV is carrying a video camera, GPS receiver, and INS sensor.

After the data are collected, an algorithm is implemented for detecting and counting from video data. The idea is to track the pixel from consecutive video frames. The algorithm was divided into four steps:

- (i) Interesting point feature extraction
- (ii) Conjugate point prediction
- (iii) Implementing cross correlation
- (iv) Verification of matched results

All four steps are explained in detail. The results show that this algorithm is effective and efficient because the vehicle can be detected tracked and accounted from a UAV system.

Advantages – With that method the results showed that the vehicle can be tracked, and from that the accounting vehicle from the video can be obtained. According to the author, the algorithm used the inherent characteristics of UAV video system of parallax to predict the corresponding points while it deletes the mismatching points.

Disadvantage – The average parallax vector is calculated from the tie points of the video stereo frames, and in some cases it cannot be calculated because there are no tie points. However, the algorithm works well and the vehicles can be tracked to reduce the statistical errors is not mentioned.

110.3.10 A Vision Algorithm for Dynamic Detection of Moving Vehicles with a UAV

Work has been conducted at the University of Picardie, to develop a UAV able to autonomously perform surveillance of an area in order to be able to detect vehicles for traffic monitoring (Kaaniche et al. 2005). A UAV was built for that project which was able to fly over a predefined area and collect. The collected data are communicated through two wireless modems where one was in charge of collecting the data and the other one of sending video information to the ground station. The algorithm was based on Dempster-Shafer theory (Shafer 1976). The *common fate*

law was used in two sets, one for representing the background and the other one for the moving objects.

All the steps of the detection algorithm are explained in detail, and focus is given to the matching operation between two images. The operation of matching between corners was followed because according to the author Kaaniche et al. (2005), it is not only less complicated than edge matching but also experimental tests have shown that corner rate reputability is more relevant than edges. To perform the matching process, for each detected corner a set of invariants is computed and then the information is stored in a vector; thus the process is then based on a vector comparison. Also, by following that operation, false matching effects are reduced. Finally the verification algorithm is discussed and the proposed algorithm is based on Dempster-Shafer theory (Shafer 1976) for vehicle recognition.

Advantages – The system is able to treat sequences with different resolutions and sequences that were taken from different altitudes. False matching effects are reduced by using the corner matching operation. Finally, the algorithm is fast and can be used for real-time applications.

Disadvantage – Matching errors are observed due to strong similarities between corners in two images. Although the system is able to detect several vehicles in the same frame, errors are observed because tree shadow edges were close to the vehicle shadow edges.

110.3.11 Supporting Traffic Flow Management with High-Definition Imagery

A feasibility study on the quality of the traffic data that could be achieved by operating an unmanned helicopter was conducted from the research group of the University of Arizona in cooperation with the Ohio State University (Toth et al. 2003).

A 2-min flight is performed in order to extract traffic flow data, and 21 images are taken. The vehicle classification is divided into three main categories: passenger cars, trucks, and multipurpose vehicles. Two subcategories are also presented, along and against, regarding the direction of the vehicles relative to the UAV direction. A small group of vehicles is tracked and velocity and tracked distance are estimated. Simulations are conducted to estimate the standard deviations of error sources as image coordinates, focal length, principal point, perspective center position, and attitude angles. The matching scheme is based on using existing road geometry information and a location from the estimated velocity of the vehicles being tracked.

Advantages – That method proves that high-definition airborne imagery should be used for obtaining traffic flow data. Also, vehicles could be tracked and traffic parameters could be calculated.

Disadvantages – Due to camera limitations on the 6-s image acquisition, an adequate sampling of vehicles position was not accomplished. Thus, only a small group of vehicles was able to be tracked for a long distance.

110.3.12 Robust Data Alignment Based on Information Theory and Its Applications in Road-Following Situation

A new approach to data fusion has been proposed from a research group at Ohio State University, focused on target recognition, surveillance, and tracking research areas in transportation (Jwa et al. 2006). The objective of that project is the implementation of a robust data alignment method without knowing the explicit pairwise correspondence between the two feature sets. An algorithm that will be able to use a shrinking error-bound or reducing the degree of uncertainty will be able to recover the best pairwise match of according to the author (Jwa et al. 2006).

The data alignment problem and two methods to solve that problem are explained in this paper. The proposed method targets a low computational cost; thus, the cost criteria and the search strategy are explained in detail. The proposed method is considered as a correspondence-less method according to the author, and it is compared with a correspondence-based method which is the least-square (LS) method. The results showed that both images are almost the same, but the performance evaluation showed that the proposed method is more robust than LS method because the average pixel error was ten times less (Jwa et al. 2006).

Advantages – The RDA method can successfully recover the transformation even under difficult conditions where noisy data are collected. By using a combined search strategy, the cost can be minimized successfully, and since the random search can avoid the local minima, that gives the ability to improve its numerical stability for a global minimum.

Disadvantages – The performance of the method depends on the characteristics of the UAV's dynamics and sensors.

110.3.13 From Images to Traffic Behavior: A UAV Tracking and Monitoring Application

Another project from the department of Computer and Information Science of the Linköping University in Sweden has to do with the implementation of a system to achieve high-level situational awareness about traffic situations in an urban area (Heintz et al. 2007). The system was tested both in simulation and on data collected during test flight, where a Yamaha RMAX helicopter powered by a 21-horsepower 2-stroke engine and endurance up to 1 h is used.

Image processing is used to calculate world coordinates of vehicles tracked in video sequences with the help of GIS and thermal cameras. Due to the fact that two cameras are onboard and the distance to an object appears different in color and infrared images, the distance to a specific object needs to be determined. The distance could be calculated given the camera specifications, the helicopter pose, and the ground elevation. DyKnow, which is a software framework that provides a set of functionality for contextually accessing, storing, creating, and processing traffic recognition algorithms, was used in order to provide the software support needed to the project (Bethke et al. 2006).

The project consists of two parts: the traffic monitoring application to monitor activities in an intersection and observing with a UAV a road segment and collecting information about the behavior of the vehicles passing by. In the first case, the UAV stays close to an intersection and monitors the cars going through. Each car is tracked for the whole path in order to create a stream of observations. In the second part, the focus was on recognizing overtakes. To do that, a stream of qualitative spatial relations between pairs of cars, such as behind and beside, are computed and used as input by the chronicle recognition system DyKnow.

Advantages – A beneficial approach for solving the problem of object recognition is proposed. The object linkage structures allows one to track a new moving object if the link to the previous is lost. Also, it could replace a lost object with a simulated object and predict the development of the object based on its history. Combined input sequences of color and thermal images are used to construct and maintain objective structures and recognize the traffic behavior of the tracked vehicles in real time.

Disadvantages – The accuracy of the computation of the true location of the car is influenced by the error of the UAV position and the springs in the camera platform suspension. The maximum error is estimated below 4–5 m for distances to the object of 80 m.

110.3.14 The Use of Small Unmanned Aircraft by Washington State Department of Transportation

A study was conducted at the University of Washington in cooperation with the U.S. Department of Transportation on June 2008 (McCormack 2008). Two types of UAVs were used for a study focused on evaluating the use of a UAV as an avalanche control tool on mountain slopes above state highways. In the first test, a fixed wing MLB BAT UAV was used, and the flight was designed to test the capability of the video camera onboard. The camera was needed to be able to view the roadway, to operate off a highway, and to survey the surrounding terrain. All three tasks were successful despite the bad weather conditions. The aircraft had operational limitations related to difficult terrain and weather. On top of that, the aircraft required a 100-foot-long flat stretch of roadway for a miniature airstrip which will limit the use of these aircraft in urban areas.

The second test was deployed by a rotary wing UAV, the RMAX, made by Yamaha. An autopilot application was demonstrated in which the aircraft followed a road with predetermined waypoints. This test was designed to simulate a survey before the start of snow clearing operations on the road, but it was also a successful test of the UAV's ability to fly along a road centerline to record traffic or weather conditions. The ability of the aircraft to hover provided a stable platform on which camera use was effective. The results show that both aircraft systems present considerable potential for aerial roadway surveillance and avalanche control. This paper describes the implementation of the experiment without focusing on the algorithms that are used on the platforms; thus, it is not included in Table 110.4.

Table 110.3 Summary of current research work covered regarding traffic surveillance with UAV platforms

| Team | System(UAV) | Objective |
|--|---|--|
| German Aerospace Center, DLR | Dornier DO-228 (Manned) | |
| Bethke et al. (2007) | Alternatively HALE (high-altitude long-endurance)-UAV is proposed | The development of an operational compact SAR system for high-flying airborne platforms |
| Department of Computer and Information Science | The WITAS Yamaha RMAX autonomous helicopter | The implementation of a system to achieve high-level situational awareness about traffic conditions in an urban area |
| Linköping University, Sweden (Heintz et al. 2007) | | |
| University of California at Berkeley (Kim 2005) | MLB BAT (fixed wing) | The implementation of an algorithm capable of detecting any types of corridors |
| Department of Electrical and Computer Engineering Western Michigan University (Chen et al. 2007) | (Not mentioned) | The implementation of a data link that connects the camera onboard an UAV to the monitoring terminals in the office of the Michigan Department of Transportation |
| Department of Civil and Environmental Engineering and Geodetic Science, Columbus (Toth et al. 2003) | (Not mentioned) | Study on the quality of the traffic data that could be achieved by operating an unmanned helicopter |
| Air Force Research Laboratory and Ohio State University (Jwa et al. 2006) | (Not mentioned) | A new approach to data fusion focused on target recognition, surveillance, and tracking research areas in transportation |
| Department of CSE, Unmanned Systems Lab University of South Florida (Puri et al. 2007) | Custom-made unmanned helicopter | Collect real-time data to monitor traffic, evaluate and assess traffic patterns, and provide accurate vehicle counts |
| Washington State Transportation Center (TRAC), University of Washington, Georgia Technical University (McCormack 2008) | MLB BAT (fixed wing) and RMAX helicopter | Two types of UAVs were used on a study focused on evaluating the use of a UAV as an avalanche control tool on mountain slopes above state highways |
| University of Nevada and NASA Ames Research Center (Gleason et al. 2011) | (Not mentioned) | Features extraction methods and classification techniques for automatic vehicle detection by UAVs on traffic control and other applications |
| Institute of Command Automation, PLA University of Science and Technology, China (Fang et al. 2011) | (Not mentioned) | Mean Shift tracking algorithms |

(continued)

Table 110.3 (Continued)

| Team | System(UAV) | Objective |
|---|---|--|
| | | The proposed algorithms build the object model according to known target appearance and possible areas that those may appear in |
| Roke Manor Research Romsey, Hampshire (Evans and Turkbeyler 2007) | (Not mentioned) | Detection of moving objects on the ground with video cameras that are mounted on UAVs |
| School of Geosciences and Geomatics, East China Institute of Technology (Cheng et al. 2009) | Custom-made small UAV | Development of techniques for generating high-resolution video flow from UAV videos |
| University of Picardie (Kaaniche et al. 2005) | UAV based on a model plane using thermal engine with a 1.5- m wingspan | Development of a UAV able to realize autonomously the surveillance of an area in order to be able to detect vehicles for traffic monitoring |
| Washington State Transportation Center (TRAC) University of Washington (McCormack 2008) | MLB BAT and Yamaha RMAX | The exploration of the general capabilities of UAVs and the evaluation of those platforms as an avalanche control tool on mountain slopes above state highways and capturing aerial images suitable for traffic and data collection surveillance |

Table 110.3 summarizes all the current research described in this paper. The information is presented similarly to Table 110.2. Table 110.4 summarizes the work done in this area by highlighting the advantages and disadvantages of each method used to solve the problem of traffic monitoring.

110.3.15 Not Yet Implemented on UAVs

In Bethke et al. (2006) the authors claim that the TRAMRAD (Traffic Monitoring with Radar) project aims to solve traffic surveillance problems by radar remote sensing techniques for wide-scale monitoring. Road traffic MTI (Moving Target Indication)/SAR techniques and methods were used on that project. According to the authors, standard SAR processing in general is not enough to detect and focus moving targets on the ground; thus other techniques that are described in the paper were considered. Ideas for along-track velocity were proposed, but it was shown that clutter suppression and vehicle acceleration are major challenges for a current estimation of the motion status of road vehicles.

In another study, by performing direct georeferencing and the use of GPS/IMU data, according to the authors Rosenbaum et al. (2008), near-real-time traffic data extraction can be implemented. Road extraction and two types of vehicle detection

Table 110.4 Advantages and disadvantages of the methods used on traffic monitoring

| Title | Advantages | Disadvantages |
|---|--|--|
| Airborne Road Traffic Monitoring with Radar (Bethke et al. 2007) | Capture of data in wide areas regardless of the weather conditions | Amount of data to process and transmit is large |
| | Resolution of SAR images is of order of 1–2 m | Does not detect vehicles with precision |
| Realtime Road Detection by Learning from One Example (Kim 2005) | Algorithm can be applied to a non-straight road with the assumption that it is partially straight | The UAV needs to fly at low altitude in order to make the lane markings visible |
| | Algorithm can be applied to other applications and not necessarily to road detection | The position errors of the hypothesis do not follow a Gaussian distribution; thus least-square line fitting gives inaccurate fitting results Real-time adaptation for a changing road structure and curved road detection is hard to accomplish |
| Statistical Profile Generation for Traffic Monitoring Using Real-Time UAV-Based Video Data (Puri et al. 2007) | The proposed algorithms can adapt on different environmental setups automatically | The helicopter's instability leads to a vehicle count error |
| | | All data is collected from a fixed point Data was not collected while the helicopter was in a moving state |
| Vehicle Detection from Aerial Imagery (Gleason et al. 2011) | The threshold of the fast detection algorithm gave a reasonably accurate estimate of the overall accuracy of the program | The rotation-invariant HOG did not perform well due to the reflective water that was in the background |
| | The performance of Gabor features under different lighting conditions was good, reducing the false positives by half | |
| An improved object tracking method in UAV videos (Fang et al. 2011) | The combination of Mean Shift and particle filtering allows the change of object size | Mean Shift can only track objects on time when their regions are known |
| | Small objects are rarely lost Valuable to track fast small moving objects like vehicles in highways | |

(continued)

Table 110.4 (Continued)

| Title | Advantages | Disadvantages |
|--|--|--|
| Visual MTI for UAV Systems (Evans and Turkbeyler 2007) | The algorithm performs well on the datasets from the supermarket even with d complex 3D structures | False alarms and missing targets remain even after stage 2; thus some clearly moving objects are not detected |
| | The algorithm is still able to detect the cars even with pedestrians walking in the scene | The geometrical interpretation of the scene is not well constrained and that causes problems even with a smooth traffic flow on the motorway |
| Real-Time Video Relay for UAV Traffic Surveillance Systems Through Available Communication Networks (Chen et al. 2007) | More than one user computer can receive the video stream simultaneously with acceptable data rate | The bandwidth of the cellular network influences the image of the video stream |
| | Experimental results showed that the proposed method can deliver quality videos to the control office in real time | During field experiments the video stream was frozen or buffered while the end users were watching the video |
| | | The results showed that the transmission with no server system scheme could only provide a video stream in continuous and smooth image in a bit rate of 45 kbps and also showed that any rate below that may not meet the requirement of the UAV project |
| Roadway traffic monitoring from an unmanned aerial vehicle (Coifman et al. 2006) | Application of platoon matching method where the network OD flow is been calculated in unobserved regions | The field of view of the UAV is not wide and the fluctuations in the traffic cause large variations in density measured from a single still frame |
| | It is not necessary to track the vehicles the whole time | The field of view continuously changes, and the sample size is small in order to calculate density and flow from individual frames The data analysis and reduction was made manually and not automated |
| Detecting and counting vehicles from small low-cost UAV images (Cheng et al. 2009) | Vehicle can be tracked | The average parallax vector is calculated from the tie points of the video stereo frames, and in some cases it cannot be calculated because there are no tie points |

(continued)

Table 110.4 (Continued)

| Title | Advantages | Disadvantages |
|--|--|--|
| | The algorithm used the inherent characteristics of UAV video system of parallax to predict the corresponding points while it deletes the mismatching points | A method to reduce the statistical errors is not mentioned |
| A Vision Algorithm for Dynamic Detection of Moving Vehicles with a UAV (Kaaniche et al. 2005) | The system is able to treat sequences with different resolutions and sequences that were taken from different altitudes | Matching errors were observed due to strong similarities between corners in two images |
| | False matching effects are reduced by using the corner matching operation | Although the system is able to detect several vehicles in the same frame, errors are observed because tree shadow edges were close to vehicle shadow edges |
| | The algorithm is fast and can be used for real-time applications | |
| Supporting traffic flow management with high-definition imagery (Toth et al. 2003) | That method proves that high-definition airborne imagery should be used for obtaining traffic flow data | Due to camera limitations, the 6-s image acquisition and adequate sampling of vehicle position were not accomplished |
| | Vehicles could be tracked and traffic parameters could be calculated | Only a small group of vehicles was able to be tracked for a long distance |
| Robust Data Alignment Based on Information Theory and Its Applications in Road Following Situation (Jwa et al. 2006) | The RDA method can successfully recover the transformation even under difficult conditions where noisy data are collected By using a combined search strategy, the cost can be minimized successfully, and since the random search can avoid the local minima, that gives the ability to improve its numerical stability for a global minimum | The performance of the method depends on the characteristics of the UAV's dynamics and sensors |

(continued)

Table 110.4 (Continued)

| Title | Advantages | Disadvantages |
|--|--|--|
| From Images to Traffic Behavior – A UAV Tracking and Monitoring Application (Heintz et al. 2007) | The object linkage structures allow to track a new moving object if the link to the previous is lost | The accuracy of the computation of the true location of the car is influenced by the error of the UAV position and the springs in the camera platform suspension |
| | Predicts the development of the object based on its history | The maximum error is estimated below 4–5 m |
| | Combined input sequences of color and thermal images are used to construct and maintain objective structures and recognize the traffic behavior of the tracked vehicles in real time | |

(edge extraction, normalized cross correlation) are the steps that were followed on the traffic processor. Due to the powerful cameras (3-head camera system) that were used in order to cover wide areas, the amount of data collected was large. The proposed system is capable of accomplishing traffic data extraction on 1 km² area within seconds. It also gives the ability to add a radar sensor and provide data in case of bad weather, where remote sensing techniques based on optical sensors fail. The data collection and experiments were not implemented on UAVs although the authors state that the system could be mounted on UAVs also. Those papers are not included in Tables 110.3 and 110.4 because those techniques are not yet implemented on UAV platforms.

110.4 Conclusion

This chapter presents a survey of current research activities going on in universities and research centers around the globe in the area of UAVs used for traffic monitoring. As a conclusion, it may be stated that UAVs can be very useful for traffic monitoring. However, issues with UAV deployment for civil applications must be addressed. Due to the fact that the approval of the Federal Aviation Administration (FAA), Federal Communications Commission (FCC), and other regulatory agencies for flying the UAVs in civil airspace keeps on delaying, universities have not yet implemented a lot of real-life scenarios.

Maneuverability and wireless network communication are two of the key points that make unmanned air vehicles more useful than the other methods currently used. The first one makes them capable of tracking vehicles on the ground, while the

second one allows the transmission and reception of instructions and image and video information to a ground base. Research is focused not only on the types of the sensors that could be used onboard (radar, vision, hybrid) but also on the type of processing (onboard, off board) of the data in vision sensors.

This research is partially supported by U.S. National Science Foundation (NSF) Grant CNS-1229236. A short version of this paper has appeared at the proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS' 13), May 2013, Atlanta, GA.

References

- K.-H. Bethke, S. Baumgartner, M. Gabele, D. Hounaman, E. Kemptner, D. Klement, G. Krieger, R. Erxleben, Air- and spaceborne monitoring of road traffic using SAR moving target indication project TRAMRAD. *ISPRS J. Photogramm. Remote Sens.* **61**, 243–259 (2006). Published by Elsevier B.V.
- K.-H. Bethke, S. Baumgartner, M. Gabele, Airborne road traffic monitoring with radar, in *Proceedings of the World Congress on Intelligent Transport Systems (ITS)*, Neijing, 2007, p. 16
- L. Breiman, Random forests. *Mach. Learn.* **45**, 5–32 (2001)
- Y.M. Chen, L. Dong, J.-S. Oh, Real-time video relay for UAV traffic surveillance systems through available communication networks, in *IEEE Wireless Communications and Networking Conference, WCNC*, Hong Kong, 2007, pp. 2608–2612
- P. Cheng, G. Zhou, Z. Zheng, Detecting and counting vehicles from small low-cost UAV images, in *ASPRS 2009 Annual Conference*, Baltimore, 2009
- B. Coifman, M. McCord, R. Mishalani, M. Iswalt, Y. Ji, Roadway traffic monitoring from an unmanned aerial vehicle, in *IEEE Proceedings of Intelligent Transportation System*, Toronto, vol. 153, No. 1, Mar 2006
- T. Cover, P. Hart, Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**(1), 21–27 (1967)
- R.J. Evans, E. Turkbeyler, Visual MTI for UAV systems, in *Proceedings of the 4th EMRS DTC Conference*, Edinburgh, July 2007
- P. Fang, J. Lu, Y. Tian, Z. Miao, An improved object tracking method in UAV videos. *Procedia Eng.* **15**(0), 634–638 (2011)
- M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**, 381–395 (1981)
- J. Gleason, A.V. Nefian, X. Bouysounousse, T. Fong, G. Bebis, Vehicle detection from aerial imagery, in *IEEE Intelligent Conference on Robotics and Automation*, Shanghai, 2011
- M. Hearst, S. Dumais, E. Osman, J. Platt, B. Shalkopf, Support vector machines. *IEEE Intell. Syst. Appl.* **13**(4), 28 (1998)
- F. Heintz, P. Rudol, P. Doherty, From images to traffic behavior – a UAV tracking and monitoring application, in *Proceedings of the 10th IEEE International Conference on Information Fusion*, Quebec, 2007
- S. Jwa, Z. Tang, U. Ozguner, Robust data alignment based on information theory and its application in road following situation, in *IEEE Intelligent Transportation Systems Conference*, Toronto, 2006
- K. Kaaniche, B. Champion, C. Pegard, P. Vasseur, A vision algorithm for dynamic detection of moving vehicles with a UAV, in *International Conference on Robotics and Automation*, Barcelona, 2005
- Z.W. Kim, Real-time road detection by learning from one example, in *Proceedings of the 7th IEEE Workshops on Application of Computer Vision, WACV/MOTIONS*, Breckenridge, vol. 1, 2005, pp. 455–460

- E.D. McCormack, The use of small unmanned aircraft by the Washington state department of transportation, Research Report Agreement T4118, Task 04, Unmanned Aerial Vehicles, Seattle, 2008
- A. Puri, A survey of unmanned aerial vehicles (UAV) for traffic surveillance, Department of Computer Science and Engineering, University of South Florida, 2005
- A. Puri, Statistical profile generation of real-time UAV based traffic data. Ph.D. thesis, University of South Florida, 2008
- A. Puri, K.P. Valavanis, M. Kontitsis, Statistical profile generation for traffic monitoring using real-time UAV based video data, in *Proceedings of the IEEE Mediterranean Conference on Control and Automation*, Athens, 2007
- K. Ro, J.-S. Oh, L. Dong, Lessons learned: application of small UAV for urban highway traffic monitoring, in *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, 2007
- D. Rosenbaum, F. Kurz, U. Thomas, S. Surir, P. Reinartz, Towards automatic near real-time traffic monitoring with an airborne wide angle camera system, in *European Conference of Transport Research Institutes (ECTRI)*, 2008
- G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976)
- C. Toth, D.A. Grejner-Brzezinska, C. Merry, Supporting traffic flow management with high-definition imagery, in *Proceedings of Joint ISPRS Workshop on High Resolution Mapping from Space*, Hannover, 2003

Measurement and Exploration in Volcanic Environments

111

Carmelo Donato Melita, Domenico Longo, Giovanni Muscato,
and Gaetano Giudice

Contents

| | | |
|----------|---|------|
| 111.1 | Introduction | 2668 |
| 111.2 | Typical Aerial Measurements in a Volcanic Scenario | 2670 |
| 111.2.1 | Visual/Thermal Data | 2670 |
| 111.2.2 | Gas Analysis and Sampling | 2671 |
| 111.3 | Importance of Unmanned Aerial Systems for Measurements in Volcanoes | 2672 |
| 111.4 | Overview of Existing Research Activities and Projects | 2676 |
| 111.4.1 | Observation Flight at Erupting Volcano, Mt. Usu, Japan | 2676 |
| 111.4.2 | Volcano Surveillance by ACR Silver Fox | 2676 |
| 111.4.3 | Unmanned Aerial Vehicle Measurements of Volcanic Carbon Dioxide Fluxes | 2678 |
| 111.4.4 | Utilization of In Situ Airborne MS-Based Instrumentation for the Study of Gaseous Emissions at Active Volcanoes | 2680 |
| 111.4.5 | The Volcan UAS Project | 2681 |
| 111.4.6 | Mt. Erebus Volcano Observatory Project | 2684 |
| 111.4.7 | UAV System Development for the Monitoring and Study of Volcanic and Natural Hazard Events | 2684 |
| 111.4.8 | Low-Altitude Remote Sensing of Volcanoes Using an Unmanned Autonomous Helicopter: An Example of Aeromagnetic Observation at Izu-Oshima Volcano, Japan | 2686 |
| 111.4.9 | An Unmanned Aerial Vehicle Synthetic Aperture Radar System (UAVSAR) | 2688 |
| 111.4.10 | U.S. Geological Survey: "Unmanned Aircraft Systems Project Office" | 2689 |
| 111.5 | Conclusions | 2690 |
| | References | 2690 |

C.D. Melita (✉) • D. Longo • G. Muscato

Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania,
Catania, Italy

e-mail: donato.melita@dieei.unict.it; domenico.longo@dieei.unict.it;
giovanni.muscato@dieei.unict.it; gmuscato@dieei.unict.it

G. Giudice

Istituto Nazionale di Geofisica e Vulcanologia, Palermo, Italy

e-mail: g.giudice@pa.ingv.it

Abstract

Volcanic environments are a challenging application for UAVs. About 10 % of the world's population lives in areas directly threatened by volcanoes, and scientists are continuously asked to perform measurements and also to better forecast volcanic activity. This chapter describes the importance of the study of volcanoes and the typical aerial measures that could be usefully executed. The main forms of measurement concern the collection of visual/thermal images and gas analysis and sampling. The specific advantages offered by the use of UAVs in comparison with manned solutions are discussed in terms of cost, simplicity, and readiness to operate. Then an overview of the major worldwide projects carried out in the last few years by several research organizations is summarized and discussed.

111.1 Introduction

The study of volcanoes is of primary importance due to the huge impact that eruptions could have on several human activities. Volcanic events can influence air-traffic safety, can lead to climate change, can seriously damage infrastructure, and may sometimes result in casualties (Alwyn 2002).

The adoption of robots as a tool for the study of Earth processes, and in particular those of volcanoes, is an open challenge to robotics: the collection and analysis of data executed by a robot in the proximity of a volcano is a difficult task faced by a limited number of research groups (Muscato et al. 2012).

In this context, in the last few years Unmanned Aerial Vehicles (UAVs) have been adopted as a powerful instrument for environmental monitoring (Ramanathan et al. 2007; Holland et al. 2001). However, even if several aerial platforms have been developed for adoption in volcanic scenarios, these studies are mainly at a preliminary stage, and there are still many open problems.

Currently, 1,500 volcanoes on Earth are potentially active, approximately 500 of which have been active during the twentieth century and about 70 are presently erupting. At the beginning of the third millennium, 10 % of the world's population lives in areas directly threatened by volcanoes. About 30,000 people have died from volcanic eruptions in the past 50 years, and damages of billions of euros have been suffered.

In these scenarios, effective prevention, or first response strategy, requires the accurate knowledge of volcanic activities. Significant advances have been made in recent years in the understanding of the behavior of volcanoes and in volcanic eruption forecasting. One of the main reasons is the improvement in the collection of field data with innovative methodologies and sensors. Such data are typically used as input for computer simulations of volcanic activity, to improve forecasts for long-lived volcanic phenomena, such as lava flow eruptions and ash-rain (Aiuppa et al. 2010; Symonds et al. 1994; Stix and Gaonac'h 2000).



Fig. 111.1 Volcanologists measuring volcanic activity at a safe distance

The most valuable data relates to volcanic activity during paroxysmal phases of eruptions, which unfortunately are the periods of greatest risk to human operators, as shown in Fig. 111.1. Thus, the collection of field data with innovative methodologies and sensors plays a key role in current research for improving volcanic models. Up to now, data has been mainly collected by means of field measurements, aerial pictures, and satellite images. This kind of measurement is usually prone to errors and is expensive, especially if quantitative and/or qualitative data is required.

As a consequence, the adoption of autonomous robots seems to be the right choice: all the required instrumentation can be carried directly close to the volcanic scenario to obtain different kinds of measurements (Muscato et al. 2012).

Remotely operated or fully autonomous flying vehicles can support, or replace, the actual methodologies used to perform environmental measurements. Several research projects are investigating the adoption of UAVs with the aim of developing an innovative, easy to manage, low-cost, and readily available survey system to be applied for the frequent monitoring of eruptive phenomena and lava flows.

As demonstrated in the following sections, the adoption of unmanned systems allows several improvements in volcanic monitoring:

- Faster and more accurate response than human-based monitoring, providing the means for the improved collection of data, both from a qualitative and quantitative standpoint

- Safer for human operators as volcanic activity can be monitored at a safe distance
- Cheaper than human-based monitoring, because UAVs do not require the use of expensive manned helicopters and highly trained operators

This is mainly possible thanks to the availability of lightweight and compact sensors that allows the adoption of small and versatile flying platforms; moreover, the present state of the art in the field of onboard stabilization, control, and navigation systems makes these aircrafts capable to be managed also by not-skilled operators.

In this chapter, the measurements of interest needed to understand, model, and predict the behavior of a volcano will be analyzed; then the principal problems related to data collection in a volcanic scenario and how the adoption of UAVs could help overcome these issues will be investigated. Finally, an overview of the main research projects related to the adoption of unmanned flying platforms for the study of volcanoes will be described and reviewed.

111.2 Typical Aerial Measurements in a Volcanic Scenario

In the last few years, the surveillance of active volcanoes has greatly improved according to the general progress of technology. The integration among several kinds of measures, obtained from different sensors, allows the quality of data to be increased. Ground-based sensor networks (Brenguier et al. 2012), aerial photogrammetry (Marsella et al. 2008), gas measures (Aiuppa et al. 2011), and information from satellites (Ganci et al. 2012), as reported in Fig. 111.2, can be used to obtain a system that is more robust to sensor noise and hardware failure, thanks to the redundancy obtained.

Many different types of aerial measurement can be useful for researches of volcanic activity. As will be explained below, UAVs can improve the collection procedure, mainly in visual and thermal data acquisition and in gas analysis and sampling.

111.2.1 Visual/Thermal Data

Pictures and videos within immediate proximity to craters allow the better understanding of the evolution of an eruption. Indeed, on active volcanoes, each new effusive eruption alters the local topography and builds new morphological features. Lava flows can threaten towns, roads, and infrastructure with complete destruction, as occurred, for example, several times for Mt. Etna volcano in the last century and which may probably occur again. Data extracted from eruption monitoring can help forecast, through simulations of the lava flow emplacement, the course and rate of advance of an ongoing flow. These pieces of information are of great potential for enabling mitigation measures, such as plans for people evacuation and the building of barriers for preventing a lava flow from destroying existing infrastructure.

In this scenario, aerial vehicles can be provided with high-resolution video and thermal cameras to take pictures of the volcanic area and of lava flows.

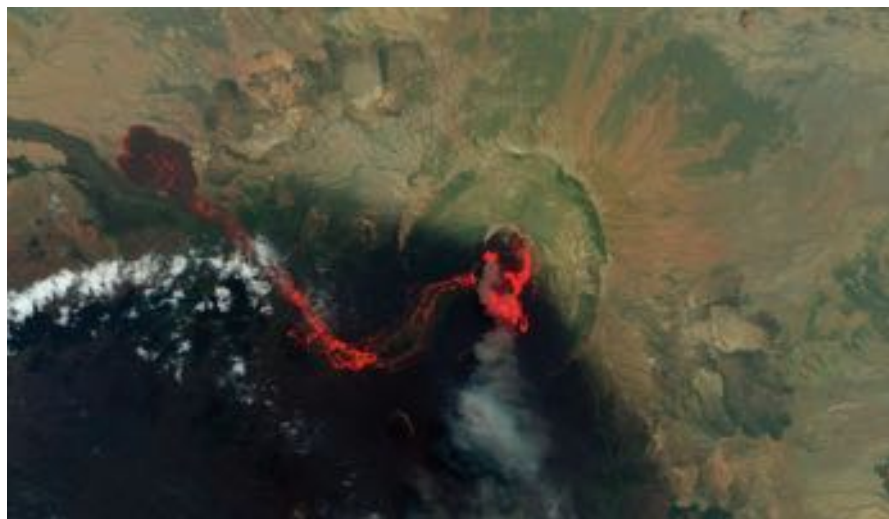


Fig. 111.2 An image of the 2011 eruption of Nabro volcano, Eritrea, taken from an Advanced Land Imager (ALI) on the Earth Observing-1 (EO-1) satellite, showing the volcano in visible and infrared light (Credit: NASA Earth Observatory image by Robert Simmon, using EO-1 ALI data. Caption by Holli Riebeek (<http://visibleearth.nasa.gov/>))

The collected data can be processed by adopting strategies derived from traditional aerial photogrammetry, representing an evolution with respect to the qualitative analysis generally carried out on close-range images, that are compared with the pre-eruption orthophotos and topographic maps. Visible images can be orthorectified for defining the planimetric (2D) expansion of a lava flow, while thermal images can be analyzed to identify the active portions of the lava flow. Images acquired with a suitable overlap can also be processed for measuring 3D coordinates of sparse or lined up points (to evaluate the average lava thickness) or for extracting the orthomosaic or Digital Elevation Model (DEM) of the area being monitored (Coltelli et al. 2007).

111.2.2 Gas Analysis and Sampling

The analysis of the gas ejected from volcanic vents is one of the main indicators of volcanic activity (Shinohara et al. 2008).

Recently, gas analyzers were independently developed by different scientists of the *Istituto Nazionale di Geofisica e Vulcanologia* (INGV, Italian Institute of Geophysics and Volcanology) and the *Geological Survey of Japan* (GSJ) allowing real-time in situ measurements of gas composition in volcanic plumes, a so-called Multi-Gas unit. Further improvements were found by strictly cooperating with other gas researchers (in Japan, USA, UK, France).

A very good correlation between variations in concentration ratios in volcanic main gases (CO_2 , SO_2 , H_2O , etc.) and volcanic activity was found in many cases. More precisely, it is important to measure the fluxes of these gases and especially the CO_2 flux (Aiuppa et al. 2011). This gas in fact is less soluble than the other main components of volcanic plumes, so it begins to exsolve from the magma at higher pressures, which is equivalent to a greater depth. In practice, a change in CO_2 flux is probably one of the earliest signals of a probable change in the volcanic activity. Of course, even the relative flux changes between these plume gases are important to understand the volcanic dynamics. Sometimes, these data were used for forecasting volcanic eruptions or explosions in Italian volcanoes (Aiuppa et al. 2010). The importance of this kind of measurement is becoming relevant for civil protection, to decide the level of risk in areas frequented by tourists that may be very attracted by active volcanoes, from all over the world.

While determining SO_2 flux is relatively easy with remote sensing techniques, CO_2 is difficult to measure remotely; only LIDAR systems are potentially useful, but cost, power requirements, weight, and dimensions are constraining parameters. In every case, all these remote measurements cannot be carried out during cloudy weather.

In fact, the most advanced techniques to obtain plume gas fluxes are based on SO_2 remote sensing flux measurements and on direct in situ monitoring with Multi-Gas units. Combining the ratios of CO_2 and other species with SO_2 , and SO_2 flux, it is possible to derive all the other fluxes.

Airborne measurements are one typical method performed to obtain gas fluxes, moving across the volcanic cloud and measuring gas concentrations and wind velocity, but again costs and safety are less guaranteed in this harsh environment, and usually only diluted parts of the plume are reached, which are not the most useful.

Bad weather conditions and issues of maintenance limit the abundance and continuity of these measurements.

111.3 Importance of Unmanned Aerial Systems for Measurements in Volcanoes

The main goal of using robotic systems for volcanoes is to reduce the level of risk for the volcanologists involved who are working too closely to volcanic vents during eruptive phenomena (Muscato et al. 2012). The need for a robotics solution for operation in hazardous environments has been already focused on and tackled by several research projects (Trevelyan et al. 2008).

However, the adoption of autonomous UAVs in volcanic environments increases the safety of the operators but also improves and simplifies many measurement operations usually performed for the study of volcanoes.

For example, leaving aside the issue of safety, at the beginning of an eruption if aerial shots are urgently needed, a helicopter or a microlight aircraft should

be available. Consequently all the facilities and operators needed should be ready at the same time: a trained pilot and an instrumentation technician (usually a volcanologist). Moreover, the takeoff/landing field is usually not in the proximity of the volcano. This would entail a considerable delay in the intervention. In addition the cost of the purchase or lease and maintenance of an aircraft and of the related facilities should not be underestimated.

On the other hand, the use of small UAVs can represent a considerable improvement in all of the phases of the operations mentioned. First, it is not necessary to have an expensive manned aircraft and the relative facilities available: UAVs can also be easily stored inside some of the numerous observatories of volcanology available in the proximity of volcanoes. Moreover the presence of highly trained personnel is not necessary: the level of autonomy provided by modern UAVs also allows non-skilled operators to use such systems and to easily control the payload (cameras and other sensors and measuring instruments). In addition UAVs can be quickly launched and consequently be able to collect information before any impending crisis, when it is easier to mitigate its effect.

All these advantages mentioned are obtained with a considerable cost reduction. The overall cost of acquiring, operating, and maintaining UAVs is much lower with respect to manned solutions. This last aspect allows the possibility to frequently repeat the surveys, thus enabling scientists to monitor the evolution of the lava flow and to collect, with sufficient temporal resolution, data to be used as input of simulation codes and for studies on flow emplacement. All these operations can be performed with no risk to the operators, since they will always stay at a safe distance from the site.

For example, the adoption of UAVs, with respect to traditional aerial photogrammetry, allows for frequently performing aerial surveys in volcanic areas, which are usually limited by the availability and costs of the traditional platforms (airplane/helicopter).

Moreover, for the same reasons, several researchers in volcanic gas monitoring in the last few years have tried UAVs to obtain plume gas fluxes. The idea is to reach the same results as airborne measurements, integrating UAV systems and Multi-Gas units, offering the possibility to get closer to the vent and to potentially even fly in cloudy or dangerous environments.

The specific environmental and working conditions of volcanoes in which UAVs have to carry out the missions suggest the adoption of an autonomous navigation system for the plane. There are two main problems that do not allow using a classic remote-controlled plane. The first is the usually long distance between a safe place and the volcanic craters. The volcanic scenario is very rugged, and it is not often possible to find a runway in proximity to the crater; so takeoff is usually executed in safe areas from where tele-operations are very difficult due to the distance between the takeoff area and the target. The second problem is related to the loss of the line of sight when the plane flies into the fumes: the gas within the volcanic plume, as can be observed from Fig. 111.3, is usually very dense and does not allow the visual recognition of the vehicle flying inside it.

Fig. 111.3 A picture taken by International Space Station crew members during the eruption of Mt. Etna in 2002 showing the density of the volcanic plume (Credit: astronaut photographs taken on October 30, 2002, provided by the Earth Sciences and Image Analysis Laboratory at Johnson Space Center (<http://visibleearth.nasa.gov/>))



Moreover, an intelligent and autonomous system requires less interaction with the first responders, requiring a minimal amount of operators, fast setup procedures, and no particular expertise.

Regarding the kind of aerial vehicle to be adopted, several considerations must be taken into account with respect to:

- The desired complexity of the system
- The payload to be carried (in terms of cost and weight)
- The logistics available
- The goal of the mission

Kites, balloons, and airships are very simple systems already adopted in meteorology and by volcanologists for studying the composition of volcanic plumes, measuring and modeling ash dispersion. However, their limited maneuverability makes these kinds of system ill-suited for adoption in complex missions or in the presence of bad weather conditions (strong winds, turbulence).

On the other hand, several groups of volcanologists have adopted helicopters for their research studies. Helicopters have a high degree of maneuverability that makes them ideal to be used in different kinds of missions in hostile environments. Moreover, these vehicles do not need a runway for takeoff and landing, can carry heavy payloads, and are suited for data acquisition, thanks to their capacity to hover and their stable flight (Saggiani and Teodorani 2004). However, helicopters are characterized by a mechanical complexity and need continuous maintenance; moreover, highly trained operators are required to fly these vehicles.

These limits have been partially overcome in the last few years thanks to the advent of multicopters, able to carry suitable payloads, equipped with low-cost autopilots, that make these aircrafts easy to be piloted.

Finally, fixed-wing UAVs are used in all those situations where the autonomy of the mission plays an important role. They are usually adopted for medium-/high- altitude flights or for long-endurance missions, which is a typical situation in a volcanic environment where the operators must stay at a safe distance from the activity.



Fig. 111.4 A graphical representation of the interaction between three cooperating UAVs flying in the proximity of a crater

Fixed-wing UAVs can carry heavy payloads and consequently be equipped with several different sensors during the same mission. Moreover, the control of these vehicles is usually simpler with respect to helicopters, thanks to the intrinsic stability of the fixed-wing vehicles adopted for data collection and monitoring. On the other hand, they are not able to hover above a target or to approach close to the terrain to perform detailed measurements on the ground.

A further possibility is to use a fleet of UAVs rather than a single aircraft. A team of UAVs could bring several advantages with respect to a single UAV system (Ollero et al. 2005; Ramanathan et al. 2007):

- It is more efficient, by allowing for a faster time to collect data or the possibility of a greater geographical coverage within a single mission.
- It is more robust to sensor noise and hardware failure, due to the redundancy in data acquisition and hardware.
- It is more powerful because it allows joint actions to be performed, such as coordinated data acquisition and dynamic allocation of surveying tasks.

For example, in the specific case of lava flow monitoring, a fleet of cooperating small UAVs can be used as shown in Fig. 111.4. An “Explorer UAV” could be used to provide a global view of the area to monitor. This airplane will be launched from a safe area and, after a climbing phase, will reach a loiter altitude over the area of interest. A video camera and thermal camera, mounted onboard the UAV, will be used to gather information about the scenario affected by the eruption and to estimate the significant geographical locations of the lava flow; this data will be used to build a global coarse map to be used by two other “Surveyor UAVs,” flying at a lower altitude, in order to decide on the appropriate configuration for collecting several high-resolution pictures of the established area. These aerial images, together with sensor poses and Ground Control Points (GCPs), will be used to reconstruct the geometry of the lava flow. Natural features, to be measured on pre-eruption high-resolution DEMs and orthophotos, and artificial features, to be placed before the survey, can be adopted as GCPs.

111.4 Overview of Existing Research Activities and Projects

In this section a summary of the main activities performed by several research groups will be briefly discussed. The main features of each system and the type of measurements and results are described.

111.4.1 Observation Flight at Erupting Volcano, Mt. Usu, Japan

One of the first reported activities concerning the adoption of UAVs in volcanoes was performed by YAMAHA Motor Co. Ltd. in Japan (Sato 2001). In April 2000 a Yamaha RMAX helicopter was used for the surveillance of the Mt. Usu volcano in the Hokkaido region of Japan. A complete autonomous control system was developed: the onboard payload was based on the Yamaha attitude control system (YACS), a differential GPS, and three digital cameras. The remote control of the RMAX was performed from a ground control station by means of a radio/video link. Another onboard video camera and a still digital camera were used to collect images and videos, performing several surveillance missions. The hazards caused by volcanic sediments and ashes were observed; moreover, thanks to the possibility of flying close to the ground, the images acquired had a better quality with respect to those taken from manned helicopters.

The helicopter was also used in February 2001 after the eruption of Mt. Oyama on Miyake-jima Island, Japan. During the missions, the sensor payload was based on a video camera and a gas detection meter, used to measure the densities of volcanic gas. Moreover, lance-shaped poles were launched from the helicopter to measure the thickness of the volcanic mudslide.

111.4.2 Volcano Surveillance by ACR Silver Fox

In 2004, thanks to a collaboration between the U.S. Geological Survey (USGS) and Advanced Ceramics Research (ACR), acquired by BAE Systems in 2009, a Silver Fox UAV by ACR (Fig. 111.5), equipped with sensor payload, was deployed to monitor the activity of Mt. St. Helens in Vancouver, Washington (Patterson et al. 2005).

The Silver Fox has a weight of 11 kg, a wingspan of 2.4 m, an endurance of up to 8 h, a speed between 35 and 60 knots (65–83 km/h), a nominal mission altitude of 150–365 m AGL, and a service ceiling of 3,657 m AMSL. It could carry a payload of up to 3.5 kg and is equipped with a navigation system that makes the vehicle fully autonomous.

In November 2004, just after the first eruption of October 2004, several flights were executed over Mt. St. Helens, as shown in Fig. 111.6. During the preliminary missions, a Silver Fox was equipped with visible electro-optical and infrared video cameras; a radio/video link was adopted to transmit data collected from the UAV



Fig. 111.5 The Silver Fox UAV by Advanced Ceramics Research – BAE Systems (Copyright © 2012 BAE Systems)



Fig. 111.6 The Silver Fox UAV ready to be launched: Mt. St. Helens is in the background (Copyright © 2012 BAE Systems)

to the ground station, flying at an altitude of 2,600m at a speed of 50 knots (92.5 km/h). Two important events were observed: a rock fall and an explosion, shown in Fig. 111.7, with a subsequent small eruption.

A similar payload was used 1 month later in a mission executed with the aim of collecting thermal data: the data acquired, used to investigate the formation of a new lava dome, were consistent with the information acquired by the USGS using the traditional methodology based on the use of a helicopter.

An attempt to obtain information on the composition of gases was executed by mounting electrochemical gas sensors onboard the UAV. The technology available in 2004 did not allow relevant information to be obtained, because the lightweight sensors available were too slow with respect to sampling time and not sensitive enough.

Fig. 111.7 Image of an explosion taken by the onboard camera of the Silver Fox UAV in November 2004 (Copyright © 2012 BAE Systems)



However, the trials, executed during the activity of Mt. St. Helens in 2004, represented a good demonstration that a flying autonomous vehicle, equipped with the appropriate sensor payload, is a winning methodology in the study of volcanic behavior.

111.4.3 Unmanned Aerial Vehicle Measurements of Volcanic Carbon Dioxide Fluxes

In this project a small remotely piloted helicopter was adopted to carry a system for the sampling and analysis of gas inside volcanic plumes (McGonigle et al. 2008). The UAV, shown in Fig. 111.8, was a Thunder Tiger Raptor 90 helicopter with a 15 cc internal combustion engine and 710 mm standard carbon fiber blades. Without a payload the aircraft weighted 5.2 kg and was able to carry payloads of up to 3 kg: with the onboard instrumentation it reached speeds of 45 knots (83.4 km/h) and had an endurance of up to 16 km.

The methodology developed needed the adoption of two sensor suites to measure SO_2 concentration and the CO_2/SO_2 ratio: this allowed the CO_2 concentration to be indirectly obtained. These measures, as explained in the previous sections, are among the main precursory signals of changes in volcanic activity.

For this reason two payloads were needed: in the first sensor suite, an ultraviolet spectrometer was mounted onboard the helicopter to measure SO_2 flux, while in the second configuration of the payload, an infrared spectrometer, together with an electrochemical sensor, was used to measure the CO_2/SO_2 ratio. An onboard GPS was used to precisely geo-localize the acquired samples.

The helicopter was remotely piloted by a human operator, and the complete mission was executed in two phases. In the first phase the helicopter was equipped with the sensor suite able to measure SO_2 flux. The vehicle was flown below the



Fig. 111.8 The Thunder Tiger Raptor 90 helicopter and the sensor payload used to measure volcanic carbon dioxide fluxes (Photo: ©Rolex Awards/Marc Latzel)

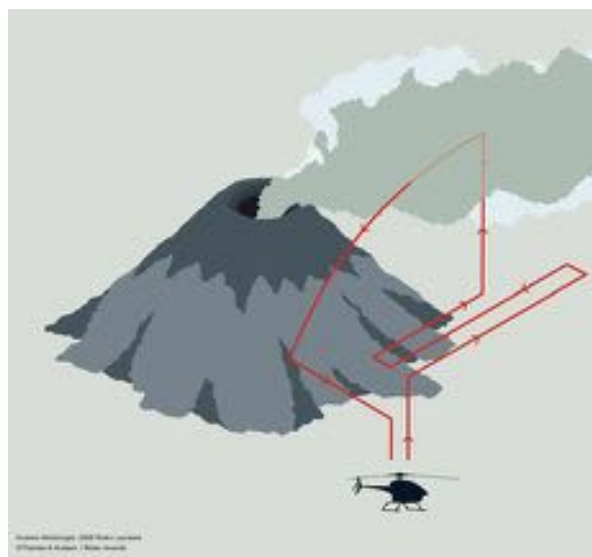


Fig. 111.9 A representation of the mission executed for the measurement of volcanic carbon dioxide fluxes. The two phases allowed the measurement of the SO_2 flux and CO_2/SO_2 ratio (Photo: ©Thames & Hudson/Rolex Awards Reference)

plume (not inside it) and, taking into account the plume transport speed, the SO_2 flux was determined (McGonigle 2007). In the second phase, the second payload was mounted, and the helicopter was flown inside the plume to measure the CO_2/SO_2 ratio, as represented in Fig. 111.9.

Several tests were performed in 2007 in the proximity of the La Fossa crater, Vulcano, Italy, in collaboration with the INGV. The measures collected allowed the scientists to calculate the flow of gases from the volcano, giving proof to the validity of the methodology adopted. However, this strategy presented all the limits related to a human-operated vehicle: the operational range was limited by the line of sight (≈ 400 m) and could not be adopted in the presence of dangerous events in the proximity of the area of interest.

111.4.4 Utilization of In Situ Airborne MS-Based Instrumentation for the Study of Gaseous Emissions at Active Volcanoes

The goal of this research activity is the realization of unmanned airborne-based instrumentation architecture for the acquisition of data to be used for the calibration and validation of satellite remote sensing data (Diaz et al. 2011).

Four NASA centers (Ames Research Center, Goddard Space Flight Center, Jet Propulsion Laboratory, Kennedy Space Center's Hazardous Gas Detection Laboratory), together with the Organization for Tropical Studies (Costa Rica) and the University of Costa Rica, are investigating the potential for small UAVs to be used for the in situ transportation of several kinds of sensors: mass spectrometers, electrochemical sensors, and ultraviolet and infrared cameras will be used to analyze the composition of gases inside volcanic plumes. The involvement of two public institution of Costa Rica represents a key to the success of the project, thanks to the presence in this country of several active volcanoes that can be used to develop and test the entire system.

With respect to the airborne system, a tethered balloon system is used to carry a payload of up to 1 kg to a maximum altitude of 5 km. Moreover, the University of Costa Rica has a WING 100 by Maryland Aerospace Inc., a small UAV with electrical propulsion, shown in Fig. 111.10, that could carry a payload of up to 1 kg and has a maximum flight time of 30 min.

For the transportation of heavier payloads, it is planned that the Sensor Integrated Environmental Remote Research Aircraft (SIERRA) will be adopted, a medium-class unmanned aircraft system actually used by NASA's Ames Research Center for remote sensing and Earth science research. This platform is able to carry up to 45 kg of payload and has been developed for long-endurance missions (up to 11 h) in harsh environments.

Several instruments are under development. The actual sensor package is represented by two systems: the first is a small and light sensor suite to be carried by balloons and has been adapted for the WING 100. It contains GPS, SO₂, and temperature sensors. The second sensor suite, named "ULISSE" (utilization of lightweight in situ sensors and remote sensing to study active volcanic emissions sites), is a portable mass spectrometer system (Diaz et al. 2010) developed in order to measure the concentration of different chemical species inside volcanic plumes.

The system was tested during the eruption of the Turrialba volcano in Costa Rica in January 2010: the data acquired before and after the eruption allowed the

Fig. 111.10 The WING 100 by Maryland Aerospace Inc. used by the University of Costa Rica for the activities related to the study of volcanoes (Courtesy of Dr. Jorge Andres Diaz, University of Costa Rica)



demonstration of the validity of the architecture under development, as well as the importance of the correlation between in situ measurements with remote sensing data obtained from satellites.

During these trials, the actual version of the ULISSE package was tested onboard a manned light aircraft, but a new prototype, to be carried by the SIERRA UAV, is under development.

111.4.5 The Volcan UAS Project

The Service Robots Group of the “Dipartimento di Ingegneria Elettrica, Elettronica e Informatica” (DIEEI) at the University of Catania is involved in several research projects focused on robotics applied to the study of volcanoes (Service Robots Group 2012). The proximity of Mt. Etna, Europe’s largest active volcano, allows the prolific cooperation with the researchers of the INGV: in the last few years several robotic platforms have been developed, taking into account the necessities and the suggestions of the volcanologists (Muscato et al. 2012).

Some of these activities are related to the development of autonomous aerial systems for applications in volcanic environments such as the *Volcan UAS Project*. The target is the realization of a flying system able to autonomously perform aerial surveillance and analyze the composition of gases inside volcanic plumes (Astuti et al. 2008a). The Volcan is a fixed-wing UAV, shown in Fig. 111.11, developed by taking into account the problems related to flight at high altitude in a volcanic scenario: harsh environment, strong winds and turbulence, rapid changes in weather, and presence of corrosive gases.

The mission will be performed at the top of Mt. Etna (3,300 m AMSL), flying over the two main, most active, degassing vents on volcano craters “Voragine” and “North-East.” Takeoff and landing procedures will be executed from “Piano delle Concazze” (37° 45′ 55.59″ N, 15° 00′ 50.35″ E, 2,800 m AMSL), an almost flat area



Fig. 111.11 The Volcan UAV ready to be launched by means of an elastic rope

near the main craters. The UAV will carry the gas-sensing instrumentation inside the plume by following a preplanned trajectory: CO_2 and SO_2 plume concentrations will be measured using onboard infrared and electrochemical cells. Data gathered will be correlated with the positions, given by the onboard GPS, for obtaining an exact map of the gas concentrations. Figure 111.12 shows a typical mission scenario.

In the laboratories of the DIEEI, a complete mission management and navigation system has been developed: it is based on the onboard modules, the communication radio link, and the ground control station.

The onboard avionics is supervised by the developed autopilot module that takes care of both the navigation and the stability of the vehicle. Taking into account the information coming from the onboard instrumentation, the control algorithms, running on an embedded microcontroller, compute the control signals to be sent to the developed actuators interface board.

An Air Data Attitude and Heading Reference System (ADAHRS) has also been developed: it computes, by means of quaternion-based Kalman filtering techniques, the attitude and the position of the plane by integrating measurements from gyros, accelerometers, magnetometers, pressure and temperature sensors, Pitot tube, and GPS.

The communication between the onboard modules is based on CAN bus and CANAerospace protocol; this protocol allows both the increase in robustness and reliability of the system and of obtaining a plug and play architecture. Thanks to



Fig. 111.12 A typical mission of Volcan UAV on the *top* of Mt. Etna. After takeoff from the base station in Piano delle Concazze, the UAV will reach the target plume and will start to collect samples. Then it will return to the base station to be recovered

the developed hardware and software interface modules, based on the adoption of the CANAerospace protocol, the onboard sensor and actuator payload can be easily changed as required (mono and stereo cameras, chemical sensors, infrared and thermal vision systems, data loggers).

A high-rate wireless Ethernet gateway ensures long-range communication between the onboard modules and the ground control station: the latter allows, with easy “click and fly” operations, mission planning, monitoring, and in-flight adjustment on a rugged notebook PC from a safe, remote location.

A Hardware-In-the-Loop architecture, based on the adoption of an X-Plane flight simulator by Laminar Research, has been used in the development phase of the modules involved in the navigation and stability control of the Volcan UAV (Astuti et al. 2008b, 2007). The developed modules have been “closed in the loop” with the flight simulator in order to test and tune the different algorithms of both autopilot and ADAHRS, allowing for a reduction in development time, cost, and risk.

Many on-field tests have been performed at the top of Mt. Etna, confirming the validity and reliability of the Volcan UAS. The completed phases of the Volcan project are:

- Test of avionics
- Test of autonomous navigation system

- Test of launching system
- Test of instrumentation
- High-altitude test of airplane

Extensive measurement campaigns in the Mt. Etna volcanic area are ongoing.

111.4.6 Mt. Erebus Volcano Observatory Project

The Mt. Erebus Volcano Observatory (MEVO [2012](#)) is developing an autonomous quadcopter for unmanned mapping and gas concentration measurements. An initial target will be the volcano's fumarolic ice caves (Curtis and Kyle [2011](#)), to be explored using a simultaneous localization and mapping (SLAM) system based on a 2D laser scanner (Hokuyo URG-04LX) mounted on the quadcopter. A UAV is required for this application because sensitive microbiology and elevated CO₂ levels complicate human entry into these environments.

The UAV will also be adapted for gas studies of Erebus' persistently degassing lava lake. Initial flight tests conducted in January 2012 demonstrated good flight performance, despite an atmospheric pressure of 650 mbar and a temperature of -15°C (Fig. [111.13](#)).

The vehicle was a quadcopter powered by a 2,200 mAh 3 cell lithium polymer battery and 950 kV brushless motors, stabilized by an ArduPilotMega controller with onboard 3-axis gyroscope, 3-axis accelerometer, and magnetometer and barometric pressure sensor for attitude, heading, and altitude control. The quadcopter was able to carry payloads of up to 1 kg with a maximum wind velocity of 15.5 knots (28.8 km/h); in the flight tests on Mt. Erebus, the payload was a GoPro HD video camera. In Fig. [111.14](#) a frame from the video, showing the Lower Erebus Hut and surrounding camp, is provided.

111.4.7 UAV System Development for the Monitoring and Study of Volcanic and Natural Hazard Events

A pioneering flight experiment to investigate the feasibility of using UAVs over volcanoes was carried out by the "Centro Nazionale Terremoti, INGV," Italy (INGV-CNT, National Earthquake Center of the Italian Institute of Geophysics and Volcanology), and the University of Bologna, DIEM Engineering Department, Forlì, Italy, on October 2004 (Saggiani et al. [2004](#)). The target was the Stromboli volcano. Stromboli is an active volcano, the island of which has a low population during the autumn and winter, and it is a good site for UAV experiments on a volcano. The University of Bologna was responsible for the aircraft, a radio-controlled plane named "Butterfly," while INGV was responsible for the payloads, campaign logistics, and civil protection relations. Butterfly, shown in Fig. [111.15](#), was a proof of concept demonstrator able to carry small scientific payloads and cameras; it is a fixed-wing UAS made of composite materials like glass, carbon, and Kevlar fibers in order to obtain a lightweight structure. It has been designed to perform different civil



Fig. 111.13 Mt. Erebus Observatory's volcano science quadcopter, based on the open-source ArduCopter project (Photo: Paige Czoski, courtesy of Dr. Aaron Curtis, Department of Earth and Environmental Science, New Mexico Institute of Mining and Technology, Socorro, New Mexico, USA)



Fig. 111.14 A frame from the onboard video camera, showing the Lower Erebus Hut, garage, and tents where researchers reside (Courtesy of Dr. Aaron Curtis, Department of Earth and Environmental Science, New Mexico Institute of Mining and Technology, Socorro, New Mexico, USA)

Fig. 111.15 The INGV Butterfly aircraft ready to take off (Courtesy of Amici S. CNT-INGV e Saggiani G.M., DIEM, University of Bologna)



missions by means of an interchangeable fuselage. It is characterized by a wing span of 3.5 m, a 50 cc boxer 4-stroke engine, and a cruise speed of 70 km/h. It can carry a payload of 3 kg, with autonomy of 4–5 h. A ground control station allows flight management, also displaying pilot cameras views, payload data, flight instruments panel, and terrain maps. Two flights were successfully realized on October 2004 and recorded by a pilot view camera (Fig. 111.16) demonstrating how promising this technique could be for volcanoes and in risk assessment and management applications.

Based on the experience of the Butterfly UAV, another UAV is under development for natural hazard surveillance missions (Saggiani et al. 2007). It represents an evolution of the previous airplane, not only with respect to the increased dimensions and payload capabilities (it is a fixed-wing plane with a wingspan of 4.6 m and a payload of 8 kg), but also for the introduction of a flight control system: the developed onboard avionics allows the UAV to autonomously perform its mission, carrying a sensor payload (EO/IR camera, gas sensors). Several on-field campaigns have been executed and are still ongoing.

111.4.8 Low-Altitude Remote Sensing of Volcanoes Using an Unmanned Autonomous Helicopter: An Example of Aeromagnetic Observation at Izu-Oshima Volcano, Japan

In 2008, the magnetization intensity mapping of Izu-Oshima volcano, Japan, was obtained by means of an unmanned helicopter carrying a magnetometer (Kaneko et al. 2011a, b). The vehicle was an RMAX-G1 helicopter by YAMAHA Motor Co. Ltd. with a 250 cc two-stroke engine, a main rotor of 3.1 m, a weight of 84 kg (including fuel), a video camera, and a payload of approximately 10 kg.



Fig. 111.16 Two Butterfly flights over volcano Stromboli have been realized in October 2004. Weather conditions were extremely good, and the wind was not very strong. A video was recorded by the pilot sight view camera during the flights. A snapshot clearly shows the volcano plume while approaching the crater in flight (Courtesy of INGV. Copyright © 2012 Google (Google Earth) © 2012 European Space Imaging @ 2012 Cnes/Spot Image)

It could reach a maximum speed of 72 km/h and a maximum altitude of 1,500 m, with an operational range within 5 km.

The helicopter was equipped with an autonomous navigation system, based on the Yamaha attitude control system (Sato 2001) and on a real-time kinematic GPS. Onboard video, acquired by a high-resolution camera and transmitted through a radio link, was broadcast to the ground control station, together with the main data transmitted from the onboard electronics.

Regarding the measurement of magnetic intensity, this was obtained by mounting a lightweight cesium magnetometer onboard the UAV. Particular attention was needed in the installation phase of this sensor, the device being influenced by the magnetic disturbances coming from the body of the helicopter; for this reason the sensor was attached beneath the helicopter by means of a three meter-long rope, as shown in Fig. 111.17.

The flights executed allowed significant magnetic observations to be obtained and for a detailed magnetization intensity map of the area around the craters to be reconstructed: from the information obtained it was possible to detect the shallow level structure beneath the caldera area. These kinds of measurements can be useful during eruptions to detect magma movement, and it is not easy to obtain the same information with a manned helicopter since it is necessary to fly at low altitude close to the terrain.



Fig. 111.17 The YAMAHA Motor Co. Ltd. RMAX-G1 helicopter carrying the cesium magnetometer, attached by means of a rope (Copyright © 2011 Taylor & Francis Group)

111.4.9 An Unmanned Aerial Vehicle Synthetic Aperture Radar System (UAVSAR)

Several NASA projects and research centers are involved in the study of aerial platforms for earth science (NASA ARC 2012; NASA DFRC 2012). One of the most interesting research activities related to the adoption of UAV for the study of volcanoes is the “Unmanned Aerial Vehicle Synthetic Aperture Radar System” (UAVSAR) project, supported by the Jet Propulsion Laboratory and NASA Dryden (NASA JPL 2012; Rosen et al. 2006).

The developed architecture includes a sensor suite that uses a polarimetric L-band synthetic aperture radar (SAR) for acquiring differential interferometric measurements: this type of information is useful for the study of terrain deformations. The sensor suite has been designed to be mounted on a Northrop Grumman autonomously operated Global Hawk aircraft. However, the first tests and demonstrations have been executed on a Gulfstream III, assisted by a Platform Precision Autopilot designed at NASA’s Dryden Flight Research Center.

The system has already been used for the study of the activities of several volcanoes in the last few years: Mt. St. Helens (2008), Aleutian Islands, Alaska (2009), Kilauea, Hawaii (2011).

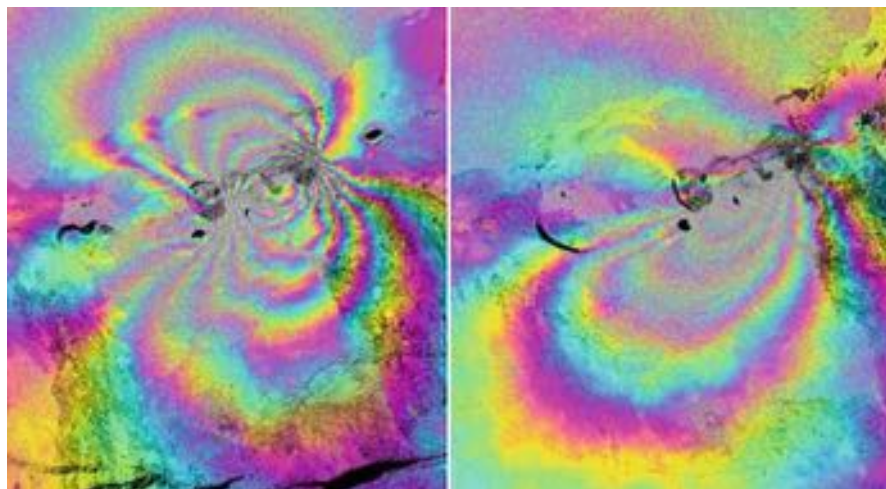


Fig. 111.18 Interferogram images of Kilauea volcano obtained by processing the data collected by the UAVSAR in 2010 and 2011: the different colors indicate the amount of deformation caused by the eruption of March 2011 (Image credit: NASA/JPL-Caltech)

Figure 111.18 shows two interferogram images of the Kilauea volcano obtained by processing the data sets collected in January 2010 and May 2011: these color-enhanced pictures represent the deformations of the volcano caused by the eruption of March 2011.

111.4.10 U.S. Geological Survey: “Unmanned Aircraft Systems Project Office”

The USGS has also planned to adopt UAVs for the study of volcanoes: the “Unmanned Aircraft Systems Project Office” (USGS 2012a) is supporting USGS in the adoption of unmanned flying platforms for Earth-science-related research activities. This project office is actually using three kinds of platform:

- AeroVironment Raven RQ-11A, a small and light hand-launched UAV equipped with an electro-optical (EO) color camera and two infrared (IR) thermal cameras, used for rapid deployment and the acquisition of data
- Honeywell RQ-16A T-Hawk Gasoline Micro Air Vehicle, a Vertical Takeoff and Landing (VTOL) UAS, adopted for real-time EO/IR video acquisition
- General Atomics MQ-9 Predator B, a high-altitude, long-endurance (HALE) UAS equipped with electro-optical, thermal infrared, and synthetic aperture radar (SAR)

The USGS is one of the most important research organizations in the world, and its contribution to the field of UAVs applied to the study of volcanoes is of fundamental importance. In its “Unmanned Aircraft Systems Roadmap 2010–2025”

(USGS 2011), the USGS details the planned action to be executed for the adoption of UASs and their integration in the Volcano Hazards Program (USGS 2012b).

111.5 Conclusions

In this chapter an overview of the importance of UAVs for volcanic measurements has been provided. UAVs can be integrated with the actual measurement methods adopted by volcanologists, allowing relevant cost reductions and more rapid deployment in the field with respect to traditional techniques. Several existing projects have been described. Most of the results are still preliminary because of the harshness of the volcanic environment, which makes it hard to operate, and because of the difficulty of synchronizing a research project plan with the unpredictable event of an eruption.

However, the actual results demonstrate the validity of this choice, and, since most of these projects are still ongoing, significant new results will certainly be obtained in the near future.

References

- A. Aiuppa, M. Burton, T. Caltabiano, G. Giudice, S. Gurrieri, M. Liuzzo, F. Murè, G. Salerno, Unusually large magmatic CO₂ gas emissions prior to a basaltic paroxysm. *Geophys. Res. Lett.* **37**, 1–5 (2010)
- A. Aiuppa, M. Burton, P. Allard, T. Caltabiano, G. Giudice, S. Gurrieri, M. Liuzzo, G. Salerno, First observational evidence for the CO₂-driven origin of Stromboli's major explosions. *Solid Earth* **2**, 135–142 (2011)
- S. Alwyn, *La Catastrophe: The Eruption of Mount Pelee, the Worst Volcanic Disaster of the 20th Century* (Oxford University Press, Oxford/New York, 2002)
- G. Astuti, D. Caltabiano, G. Giudice, D. Longo, C. Melita, G. Muscato, A. Orlando, Hardware in the loop tuning for a volcanic gas sampling UAV, in *Advances in Unmanned Aerial Vehicles State of the Art and the Road to Autonomy*, ed. by K.P. Valavanis (Springer, Dordrecht, 2007), pp. 473–493. ISBN:978-1-4020-6113-4
- G. Astuti, G. Giudice, D. Longo, C.D. Melita, G. Muscato, A. Orlando, An overview of the Volcan project: an UAS for exploration of volcanic environments. *J. Intell. Robot. Syst.* **54**(1) 471–494 (2008a). Springer
- G. Astuti, D. Longo, C.D. Melita, G. Muscato, A. Orlando, HIL tuning of UAV for exploration of risky environments. *Int. J. Adv. Robot. Syst.* **5**(4), 419–424 (2008b). issn:1729-8806
- F. Brenguier, P. Kowalski, T. Staudacher, V. Ferrazzini, F. Lauret, P. Boissier, P. Catherine, A. Lemarchand, C. Pequegnat, O. Meric, C. Pardo, A. Peltier, S. Tait, N.M. Shapiro, M. Campillo, A. Di Muro, First results from the UnderVolc high resolution seismic and GPS network deployed on Piton de la Fournaise. *Volcano Seismol. Res. Lett.* **83**, 97 (2012)
- M. Coltelli, C. Proietti, S. Branca, M. Marsella, D. Andronico, L. Lodato, Analysis of the 2001 lava flow eruption of Mt. Etna from three-dimensional mapping. *J. Geophys. Res.* **112**, F02029 (2007)
- A. Curtis, P. Kyle, Geothermal point sources identified in a fumarolic ice cave on Erebus volcano, Antarctica using fiber optic distributed temperature sensing. *Geophys. Res. Lett.* **38**, L16802 (2011). doi:10.1029/2011GL048272

- J.A. Diaz, D. Pieri, C.R. Arkin, E. Gore, T.P. Griffin, M. Fladeland, G. Bland, C. Soto, Y. Madrigal, D. Castillo, E. Rojas, S. Achí, Utilization of in situ airborne MS-based instrumentation for the study of gaseous emissions at active volcanoes. *Int. J. Mass Spectrom.* **295**(3), 105 (2010)
- J.A. Diaz, D. Pieri, G. Bland, T. Miles, E. Gore, R. Arkin, Utilization of small in-situ airborne platforms, lightweight sensors and remote sensing for volcanic plume analysis, in *34th International Symposium on Remote Sensing of Environment*, Sydney, Apr 2011
- G. Ganci, A.J.L. Harris, C. Del Negro, Y. Guehenneux, A. Cappello, P. Labazuy, S. Calvari, M. Gouhier, A year of lava fountaining at Etna: volumes from SEVIRI. *Geophys. Res. Lett.* **39**, L06305 (2012)
- G.J. Holland, P.J. Webster, J.A. Curry, G. Tyrell, D. Gauntlett, G. Brett, J. Becker, R. Hoag, W. Vaglianti, The Aerosonde robotic aircraft: a new paradigm for environmental observations. *Bull. Am. Meteorol. Soc.* **82**(5), 889–902 (2001)
- T. Kaneko, T. Koyama, A. Yasuda, M. Takeo, T. Yanagisawa, K. Kajiwar, Y. Honda, Low-altitude remote sensing of volcanoes using an unmanned autonomous helicopter: an example of aeromagnetic observation at Izu-Oshima volcano, Japan. *Int. J. Remote Sens.* **32**(5), 1491–1504 (2011a)
- T. Kaneko, T. Ohminato, T. Koyama, M. Takeo, A. Watanabe, T. Shimano, T. Yanagisawa, Y. Aoki, A. Yasuda, Y. Honda, The summit crater of Mt. Shinmoe and the adjacent areas, taken from unmanned autonomous helicopter. *Bull. Volcanol. Soc. Jpn.* **56**, 171–173 (2011b) (in Japanese)
- M. Marsella, M. Coltelli, C. Proietti, S. Branca, R. Monticelli, 2002–2003 lava flow eruption of Stromboli: a contribution to understanding lava discharge mechanisms using periodic digital photogrammetry surveys, in *The Stromboli Volcano: An Integrated Study of the 2002–2003 Eruption*, ed. by S. Calvari et al. Geophysical Monograph Series, vol. 182 (American Geophysical Union, Washington, DC, 2008), pp. 229–246
- A.J.S. McGonigle, Measurement of volcanic SO₂ fluxes with differential optical absorption spectroscopy. *J. Volcanol. Geotherm. Res.* **162**, 111 (2007)
- A.J.S. McGonigle, A. Aiuppa, G. Giudice, G. Tamburello, A.J. Hodson, S. Gurrieri, Unmanned aerial vehicle measurements of volcanic carbon dioxide fluxes. *Geophys. Res. Lett.* **35**, L06303–L06308 (2008)
- Mt. Erebus Volcano Observatory (MEVO) homepage, <http://erebus.nmt.edu>. Accessed 9 May 2012
- G. Muscato, F. Bonaccorso, L. Cantelli, D. Longo, C.D. Melita, Volcanic environments: robots for exploration and measurement. *IEEE Robot. Autom. Mag.* **19**(1), 40 (2012)
- NASA ARC – Ames Research Center, Earth Science Division, <http://geo.arc.nasa.gov/>. Accessed 9 May 2012
- NASA DFRC – Dryden Flight Research Center, <http://www.nasa.gov/centers/dryden/home/index.html>. Accessed 9 May 2012
- NASA JPL – Jet Propulsion Laboratory, Unmanned aerial vehicle synthetic aperture radar system, <http://uavsar.jpl.nasa.gov/index.html>. Accessed 9 May 2012
- A. Ollero, S. Lacroix, L. Merino, J. Gancet, J. Wiklund, V. Remuss, I. Veiga, L.G. Gutierrez, D.X. Viegas, M.A. Gonzalez, A. Mallet, R. Alami, R. Chatila, G. Hommel, F.J. Colmenero, B. Arrue, J. Ferruz, J. Martinez de Dios, F. Caballero, Architecture and perception issues in the COMETS multi-UAV project. *IEEE Robot. Autom. Mag.* **12**(2), 46–57 (2005)
- M.C.L. Patterson, A. Mulligan, J. Douglas, J. Robinson, L. Wardell, J.S. Pallister, Volcano surveillance by ACR Silver Fox, aerospace, paper no. 6954, American Institute of Aeronautics and Astronautics (AIAA) Infotech., Arlington, 26–29 Sept 2005
- V. Ramanathan, M.V. Ramana, G. Roberts, D. Kim, C. Corrigan, C. Chung, D. Winker, Warming trends in Asia amplified by brown cloud solar absorption. *Nature* **448**, 575–579 (2007). Nature Publishing Group
- P.A. Rosen, S. Hensley, K. Wheeler, G. Sadowy, T. Miller, S. Shaffer, R. Muellerschoen, C. Jones, H. Zebker, S. Madsen, UAVSAR: a new NASA airborne SAR system for science and technology research, in *2006 IEEE Conference on Radar*, Verona, 24–27 Apr 2006, p. 8
- G.M. Saggiani, B. Teodorani, Rotary wing UAV potential applications: an analytical study through a matrix method. *Aircr. Eng. Aerosp. Technol.* **76**(1), 6 (2004)

- G.M. Saggiani, F. Persiani, A. Ceruti, M.F. Buongiorno, S. Amici, C. Spinetti, G. Romeo, G.F. Di Stefano, D. Quagliotti, L.M. Loreface, D. Pieri, An UAV system for volcanic activity monitoring and surveillance, in *2004 IEEE Geoscience and Remote Sensing Symposium*, Anchorage, 20–24 Sept 2004
- G.M. Saggiani, F. Persiani, A. Ceruti, P. Tortora, E. Troiani, F. Giulietti, S. Amici, M.F. Buongiorno, G. Distefano, G. Bentini, M. Bianconi, A. Cerutti, A. Nubile, S. Sugliani, M. Chiarini, G. Pennestri, S. Petrini, D. Pieri, UAV system development for the monitoring and study volcanic and natural hazard events, in *2007 American Geophysical Union, Fall Meeting*, San Francisco, 10–14 Dec 2007
- A. Sato, Research, development and civil application of an autonomous, unmanned helicopter, in *Proceedings of 57th American Helicopter Society International Forum*, Washington, DC, 2001
- H. Shinohara, A. Aiuppa, G. Giudice, S. Gurrieri, M. Liuzzo, Variation of $\text{H}_2\text{O}/\text{CO}_2$ and CO_2/SO_2 ratios of volcanic gases discharged by continuous degassing of Mount Etna volcano, Italy. *J. Geophys. Res.* **113**, B09203 (2008)
- J. Stix, H. Gaonac'h, Gas, plume and thermal monitoring, in *Encyclopedia of Volcanoes*, ed. by H. Sigurdsson (Academic, San Diego, 2000), pp. 1141–1164
- R.B. Symonds, W.I. Rose, G.J.S. Bluth, T.M. Gerlach, Volcanic-gas studies: methods, results and applications, in *Volatiles in Magmas*, ed. by M.R. Carroll, J.R. Holloway. Reviews in Mineralogy, vol. 30 (Mineralogical Society of America, Washington, DC, 1994), pp. 1–66
- J.P. Trevelyan, S. Kang, W.R. Hamel, in *Springer Handbook of Robotics*, ed. by B. Siciliano, O. Khatib (Springer, Berlin, 2008), p. 1101
- Università di Catania- Service Robots Group homepage, <http://www.robotic.diees.unict.it>. Accessed 9 May 2012
- USGS – United States Geological Survey National United States Geological Survey, United States geological survey unmanned aircraft systems roadmap 2010–2025 (2011). Available on line http://rmgsc.cr.usgs.gov/outgoing/UAS/UAS_RoadMap_DRAFT. Accessed 9 May 2012
- USGS – United States Geological Survey National Unmanned Aircraft Systems Project Office (2012a), <http://rmgsc.cr.usgs.gov/uas>. Accessed 9 May 2012
- USGS – United States Geological Survey National Volcano Hazards Program (2012b), <http://volcanoes.usgs.gov>. Accessed 9 May 2012

Cooperative Unmanned Aerial Systems for Fire Detection, Monitoring, and Extinguishing

112

Luis Merino, José Ramiro Martínez-de Dios, and Aníbal Ollero

Contents

| | | |
|---------|--|------|
| 112.1 | Introduction | 2694 |
| 112.2 | An Architecture for Multi-UAS Teams for Forest Firefighting | 2695 |
| 112.2.1 | Vehicles and Sensors | 2696 |
| 112.2.2 | Decision-Making System | 2697 |
| 112.2.3 | The Perception System | 2698 |
| 112.3 | Fire Detection | 2698 |
| 112.3.1 | Sensors | 2699 |
| 112.3.2 | Perception Techniques | 2701 |
| 112.3.3 | Experimental Results | 2703 |
| 112.4 | Fire Monitoring and Measurement | 2706 |
| 112.4.1 | Sensors | 2709 |
| 112.4.2 | Perception Techniques | 2709 |
| 112.4.3 | Experimental Results | 2711 |
| 112.5 | Cooperation of UAS with Ground Infrastructure for Fire Extinguishing Tasks | 2714 |
| 112.5.1 | Sensors and Actuators | 2715 |
| 112.5.2 | Techniques | 2717 |
| 112.5.3 | Experimental Results | 2717 |
| 112.6 | Conclusions | 2719 |
| | References | 2721 |

L. Merino (✉)
Pablo de Olavide University, Seville, Spain
e-mail: lmercab@upo.es

J.R. Martínez-de Dios
University of Seville, Seville, Spain
e-mail: jdedios@us.es

A. Ollero
Robotics, Vision and Control Group, University of Seville, Seville, Spain

Center for Advanced Aerospace Technologies (CATEC), Parque Tecnológico y Aeronáutico de
Andalucía, La Rinconada, Spain
e-mail: aollero@us.es

Abstract

This chapter deals with the application of cooperative unmanned aerial systems to forest fires. Fire detection and fire monitoring and measurement to assist in fire extinguishing are discussed. The chapter presents a decision and control architecture for multi-UAS teams in forest firefighting. Then, the applications to fire detection and fire monitoring/measurement are studied by including in both cases the sensors, mainly infrared and visual cameras, the perception techniques (fire segmentation, geolocalization, and data fusion), and some experimental results. Finally the fire extinguishing is also considered, and some results are presented.

112.1 Introduction

Most studies on civilian applications of UAS have pointed out the interest in forest firefighting applications. Several validations in experimental fires and even in operational conditions have been presented ([►Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States](#)).

Hundreds of thousands of hectares are devastated by wildfires each year. Forest fires lead to the destruction of forest and the wildlife that inhabits them and have a disastrous social, economic, and environmental impact. Forest firefighting involves extensive human resources and is a very dangerous activity, which originates many casualties every year. In many cases, the lack of information about the dynamic evolution of fire plays an important role in these accidents. As very suitable platforms for information gathering in inaccessible places, UAS are a potentially powerful tool in the fight against forest fires (see Fig. 112.1).

Forest firefighting involves several different tasks, usually classified as pre-fire activities, during-fire activities, and post-fire activities.

- Pre-fire activities include all the tasks related to prevention of fire, like:
 - Estimation of vegetation maps, risk maps, and a general assessment of the situation.
 - Fire detection: the detection and localization of potential fire spots. An early fire detection is very important to avoid large damage.
- Once a fire has been declared, two main activities are considered:
 - Forest fire monitoring can be defined as the estimation in real time of the evolution the fire spread.
 - Forest fire extinguishing. Different means are employed to extinguish the fire and/or avoid its further propagation. For instance, manned helicopters are usually employed to carry and deploy water and retardants over the fire.
- After a fire has been extinguished, several additional tasks have to be performed:
 - Embers mapping and monitoring: the reminders of the fire have to be carefully monitored in order to avoid a reignition of the fire.
 - Burnt area assessment: an analysis of the affected zone is important for future actuations, insurance, etc.

Fig. 112.1 A view of a controlled fire from a helicopter. Aerial systems can be very valuable to obtain information about the fire evolution



- Reforestation activities: this task is devoted to the recovery of the affected burned area.

UAS can be employed in a greater or smaller degree in nearly all of these activities; in fact, there has been already systems demonstrated in several of them, like vegetation mapping (Bryson et al. 2010) and fire monitoring ([►Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States](#)). This chapter summarizes the works of the authors in the last 10 years on the topic. In particular, on the use of teams of UAS for the tasks of forest fire detection, monitoring and extinguishing. No major road blocker is seen to prevent the use of UAS even in more complex tasks like reforestation (for instance, by deploying seeds) or extinguishing by carrying water or other products over by the teams of UAS ([►Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation](#)).

In order to accomplish the missions related to forest firefighting, this chapter considers the use of fleets of UAS, instead of single, very powerful, UAS, discussing some of the advantages and drawbacks. Sometimes it can be helpful to have a team because one single UAS may not have the payload to carry the sensors and actuators required for the tasks. Also, in many missions, like fire detection, a large area has to be covered, and it may be more convenient to devote several UAS to the task. Moreover, in some applications, such as fire monitoring, measurements should be taken in different places at the same time.

112.2 An Architecture for Multi-UAS Teams for Forest Firefighting

As commented above, this chapter describes the application of UAS teams in different forest firefighting missions/tasks. From a high-level point of view, an

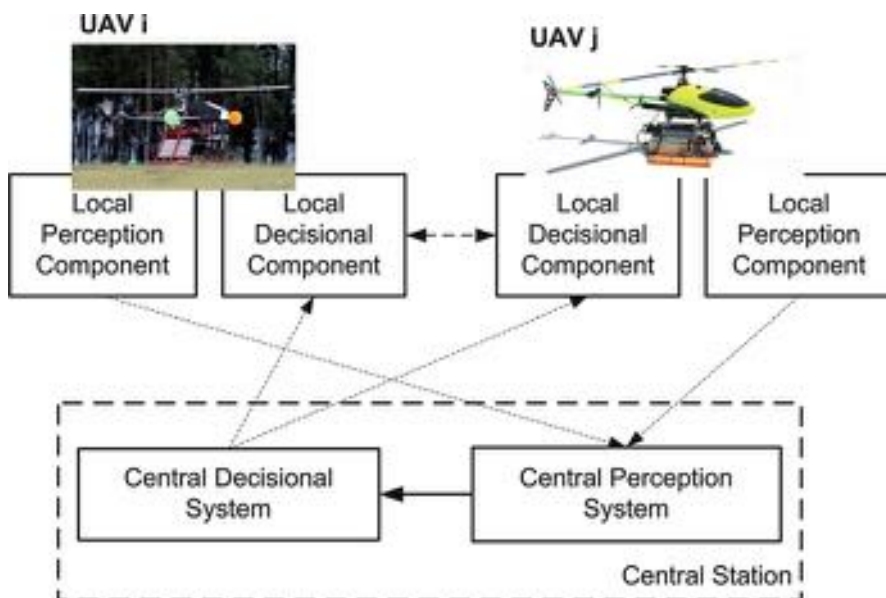


Fig. 112.2 The system consists of several vehicles and two main components: the decisional system and the perception system

architecture to integrate and control all the UAS in these missions consists of two main components (Ollero and Maza 2007): a decisional system, in charge of coordinating the UAS fleet, and a perception system to process the sensor payload.

The system considered in this chapter is composed of a team of aerial vehicles and a central station (see Fig. 112.2). Both, the decisional system and the perception system, consist of local components onboard the UAVs and components in a central station. Depending on the tasks performed by the local and central components, the system can be adapted from a more centralized system in which the UAS have less autonomy, to a more decentralized system in which more decisions are delegated to the UAS.

112.2.1 Vehicles and Sensors

The system was designed to integrate heterogeneous unmanned aerial vehicles such as helicopters, fixed-wing aircrafts, and airships. Of course, some minimal requirements on the platforms and their capabilities are needed:

Operational Autonomy: The UAS should be able to perform a set of basic (atomic) tasks, mainly to fly waypoints and to hover (or to perform flights in orbits around the desired points in the case of planes) at given places autonomously (Ollero and Merino 2004).

Localization: They should be able to localize themselves on the same reference frame. In general, the vehicles are equipped with differential GPS receivers and Inertial Measurement Units (IMUs), which allow them to localize themselves in a common world reference frame.

Perception Payload: For perception purposes, the vehicles carry a sensor payload that will depend on the particular application. For forest firefighting, infrared and visual cameras are the usual payload, together with additional sensors. In some firefighting missions, even additional actuators may be required. More information will be considered in the following sections.

Communications: The UAS carry onboard communication devices to be able to receive commands from the ground station, and to send information back to it, or to communicate directly with other UAS.

112.2.2 Decision-Making System

Conventional UAS systems can perform autonomously flights between waypoints. However, the integration of a fleet of UAS in a forest fire mission requires a decisional system for the coordination and control of the fleet.

This decisional system implements four main different mechanisms:

Task Allocation Arises in a multi-Unmanned Aerial System, where each of the UAS is able to perform tasks in response to the tasks requests. The issue is to decide which system should be endowed with each given task to be performed (for instance, which UAS should go to the different viewpoints to observe the fire). This requires the capability to assess the interest of assigning a certain system to a given task. This operation is especially difficult when the decision has to be done taking into account the current individual plans of the UAS as well as the tasks left to be assigned.

Task Planning Aims at building a sequence of basic tasks to perform, in order to achieve a given high-level mission, for instance, a fire monitoring mission.

Coordination Is a process that arises within a system if given resources (either internal or external) are simultaneously required by several components of this system. In the case of a multi-UAS system, a classic coordination issue to deal with is the sharing of space between the different vehicles, to ensure that each vehicle will be able to perform its plan safely and coherently regarding the plans of the other systems. For instance, as a fire detection mission involves complete coverage of a given area, the region should be divided among the available systems according to their relative capabilities (such as maximum speed, autonomy, field of view of the cameras).

Another example can be found in the context of fire monitoring: several synchronized perceptions of the event are required with convenient locations and orientations of the involved cameras.

Supervision Deals with the management (control) of the tasks execution, in several ways:

- A first concern is simply to keep the system aware of the tasks processing evolution during their executions.
- A second concern is to detect the possible task failures and (if possible) to react to such events in a way that will prevent the system to fail.

The decisional system employed in the UAS presented in this chapter has been evolving with the years, from more centralized implementations to decentralized implementations. The instantiations of the system are described in detail in Lacroix et al. (2007) and Maza et al. (2011) and deal with all the topics considered above.

112.2.3 The Perception System

The Perception System should obtain the high-level information required by the decisional system to achieve the firefighting mission.

It involves mainly the following tasks:

Sensor Low-Level Processing Each of the sensors carried out by the different UAS should be processed to remove noise, stabilize images, etc.

Information Extraction The relevant information for the tasks at hand should be extracted from the sensorial data. Different algorithms will be employed depending on the particular task.

Data Fusion One of the main advantages of having a fleet of UAS is that it is possible to combine the information gathered by all of the vehicles in order to reduce the inherent uncertainty due to noise, incomplete information, etc. Therefore, the main task of the perception system is to determine how the information from the different UAS should be combined to achieve a better assessment of the situation.

The instantiations of the system are described in detail in Ollero et al. (2005) and Merino et al. (2007). The next sections will describe in further detail the different subelements of the architecture for the particular applications of forest fire detection, monitoring, and extinguishing.

112.3 Fire Detection

The Forest Fire Detection Mission can be defined as the task of finding potential fire spots on a given area and determining their location in geographical coordinates, so the fighting means can be deployed as soon as possible.

Important issues for this mission are the typical requirement of coverage of large areas and the potential generation of false alarms, which should be minimized.

In general, fire detection is performed by persons that observe a fire and then call the local firefighting agencies. Systems have been developed for the automatic

detection of fires. Most automatic systems are based on cameras placed on ground as, for example, the BOSQUE system, which relies on infrared cameras and automatic image processing techniques for fire detection, including false alarm rejection (Arrue et al. 2000). Color ground cameras are also used for autonomous detection of forest fires (de Vries and Kemp 1994; Dierre et al. 1999). Some systems are based on smoke plume detection algorithms (Den Breejen et al. 1998; Gómez Rodríguez et al. 2002). Other sensors have been also employed for automatic forest fire detection, like LIDAR (Utkin et al. 2003), due to its interesting characteristics for smoke plume detection in situations where visual cameras cannot operate (as, for instance, during night). These systems have some drawbacks, such as the coverage and the lack of reliability of the automatic detection in changing environmental conditions.

The use of satellites for fire detection has been also considered (San Miguel Ayanz et al. 2005; Kelhä et al. 2003). Besides, manned airborne-based systems have been also used, like in the Airborne Wildfire Intelligence System (AWIS), in Canada (Campbell et al. 2002).

Here, it is described how a team of UAS can be used for automatic forest fire detection, confirmation, and localization. In this mission, a given area has to be patrolled looking for potential fire alarms. If an alarm is detected, its geographical position should be communicated to the authorities, discarding false alarms. The use of a multi-UAS team in this case has several benefits: the area can be divided into different zones, assigned to different UAS, and the UAS can collaborate, as it will be seen, to confirm the detected alarms by means of sensor fusion techniques (Merino et al. 2006).

112.3.1 Sensors

Automatic fire perception is a complex problem. Besides the general difficulties of autonomous perception in natural scenarios, due, for example, to uncontrollable and sudden changes in environmental conditions, there are others related to the particular characteristics of fire including the harsh environment (e.g., smoke) and the difficulties in predicting the spread of fire and smoke. The proposed system addresses the sources of error present in forest fire perception by exploiting complementarities between different types of sensors that are mounted on UAS.

The multi-UAS fire detection system relies on two main types of imaging sensors with different wavelength bands: infrared cameras and visual cameras, which have interesting synergies for fire perception. Visual cameras provide images in the visible range [0.4–0.7] m. Flames are visible in visual images if images are not occluded by smoke.

Infrared cameras generate images of the scene that contain the radiation intensity field within the infrared band. They can transform radiation measurements in temperature estimations and can also generate thermograms containing temperature of the objects in the scene (Hudson 1969). Infrared cameras can “see” in pitch black



Fig. 112.3 (Left) Visual image taken in one multi-UAS fire perception experiment. (Right) Infrared images of the same scene taken with one Raytheon 2000AS OEM infrared micro-camera

conditions and are not affected by smoke, being highly transparent compared to the high radiation levels originated in a forest fire. Infrared cameras in the mid-infrared spectral window [3–5] μm are usually preferred for fire detection mainly due to its lower atmospheric absorption ([► Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States](#)).

In the last decade infrared camera technology has evolved significantly. While in the decade of the 1990s, most infrared cameras weighted several kilograms, consumed hundreds of watts, and their cryogenic cooling systems required frequent maintenance; now it is possible to find radiometric infrared cameras that weigh less than 150 gram, consume less than 1.5 W, and require no maintenance since they have no cooling system. These advances together with a remarkable cost decrease have motivated their use in a growing number of applications.

Figure 112.3 shows two images, one infrared and one visual, of the same scene taken during a multi-UAS fire detection experiment. While the fire base is easier to be detected with the infrared images, the fire flames are better detected if the smoke does not hide them.

UAS can be equipped with one camera (visual or infrared) or two cameras (visual and infrared). UAS also have GPS or Differential GPS (DGPS) receivers and Inertial Measurement Units (IMUs) so that their location and orientation are known. The accuracy depends on the characteristics of the sensors and also of the satellite visibility, which deteriorate significantly in low-altitude flights in mountainous terrains, and communication conditions (corrections of the DGPS). The photograph of a helicopter UAS, Heliv, used in the experiments is shown in Fig. 112.4. It carries a pan and tilt device with one low-weight infrared micro-camera and one visual camera.

An additional sensor considered for fire detection is a fire detector, whose main component is a photodiode setup to limit its sensibility to the band of [185, 260] nm, normally associated to fires (see Fig. 112.5). The data of this sensor can be processed to obtain a binary value indicating if a fire is present or not within the field of view of the sensor. However, the fire positioning information obtained from these



Fig. 112.4 (a) Main sensors and components of Heliv helicopter used in forest fire perception experiments. (b) Detailed picture of the pan and tilt device with one Raytheon 2000AS OEM infrared micro-camera and one visual camera

sensors is poorer than that given by the cameras, as they do not provide any bearing information. Also, it is not possible to determine if a measure is due to a big fire far away or a nearby small fire, or if the fire is generated by one or several sources (the data cannot be segmented into contacts).

112.3.2 Perception Techniques

The detection and precise localization of fire spots require to address several issues within the perception system:

112.3.2.1 Fire Segmentation

When using cameras, the first step in the processing stack is to estimate potential fire spots on the image plane. For that, segmentation algorithms are employed.

Fig. 112.5 One of the helicopters, Marvin, has attached at its nose a fire detector sensor

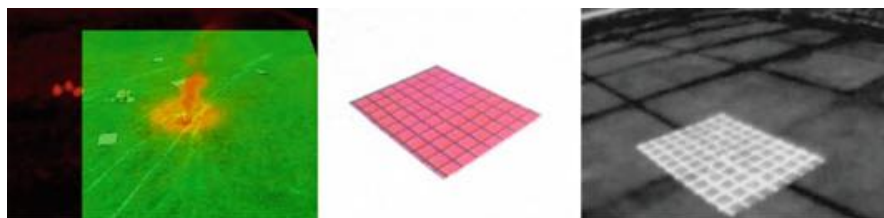


Fig. 112.6 The images from the cameras of Fig. 112.3 can be combined, obtaining a multimodal sensor. Resultant combined image: the image is obtained filling the *red field* with the infrared pixel intensities and the *green field* with the visual image. This is more effective in discarding false alarms. To relate both images, calibration patterns also seen in the infrared band are required

The nature of these algorithms depends on the type of the images, but in general they look for signatures related to the fire, like color signatures of infrared features. A summary of the main algorithms employed can be found in Martínez de Dios et al. (2008).

If a pair of infrared and visual cameras is available, it is even possible to combine them to create a multi-modal sensor, which allows for a more reliable fire detection (Martínez-de Dios et al. 2005), as shown in Fig. 112.6. In general the algorithms will have a certain probability of producing false alarms.

112.3.2.2 Geolocation

The fire spots detected on the image plane should be related to their geographical coordinates, to indicate to the fighting means where they should send their resources. In order to do that, the information from GPS and IMUs is combined with calibration information and a Digital Terrain Map to determine the coordinates corresponding to a given pixel on the image plane.



Fig. 112.7 During the demonstration, small controlled fires were performed

112.3.2.3 Data Fusion

As a result of the fire segmentation algorithms, not only fire spots, but also false alarms can occur. Moreover, the limited precision of the localization devices can produce large geolocalization errors.

Having a set of UAS, it is possible to use complementary sensors and points of view, so that the different sensorial information can be fused and, therefore, the number of false alarms can be decreased and the precision on the localization of the alarms greatly enhanced. Bayesian approaches have been employed for that, including grids and Kalman filters (Merino et al. 2006).

112.3.3 Experimental Results

The algorithms and architecture were tested in a set of experiments involving a fleet of three UAVs, namely, two helicopters (Heliv and Marvin) and one blimp (Karma). In the experiments, small controlled fires were set by firemen. Figure 112.7 shows a photograph of one of these fires. In every experiment, the position of the fire was recorded by using GPS receivers for validation purposes.

The mission usually consisted of a detection and a confirmation phase. In the detection phase, each UAS of the fleet is commanded to a certain area to patrol and

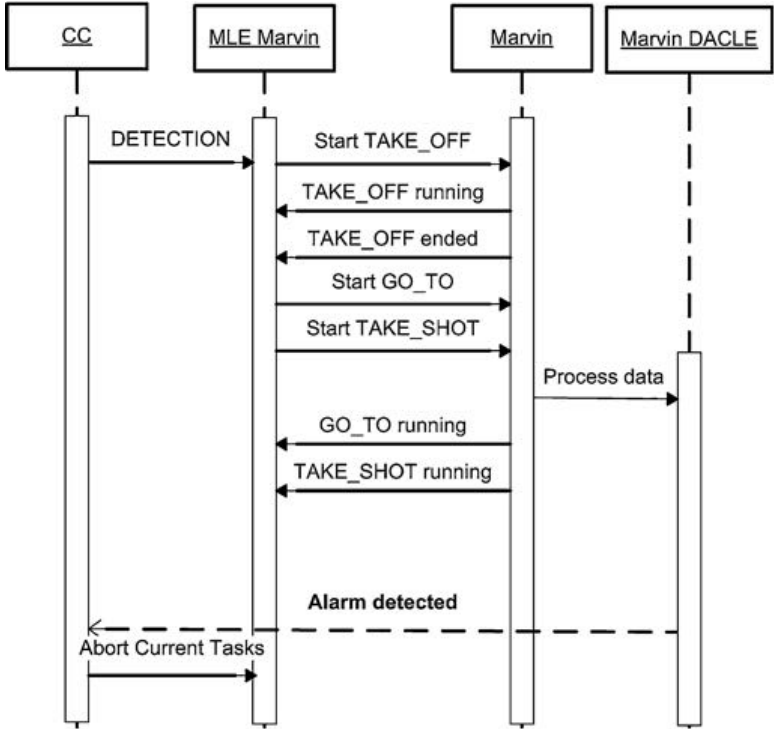


Fig. 112.8 Exchange of tasks for the detection phase. A detection mission is sent to Marvin from the Control Centre (CC). The local decisional component (called MLE) generates a set of waypoint tasks to cover the area. It also initiates the processing of data from the local perception component (called DACLE)

process their sensors, individually looking for potential fire alarms. If an alarm is detected, a confirmation phase is launched, in which a different UAS, if possible with complementary sensors, is sent to the same place. Data fusion is used to confirm (or discard) the alarm and to refine its estimated position.

In one of the performed experiments, during the first phase, the helicopter Marvin is commanded to perform a detection mission (see Fig. 112.8). The decisional layers decompose the mission into a set of atomic tasks: the helicopter takes off (TAKE_OFF) and determines a set of waypoints to cover the area (GO_TO). At the same time, it gathers information from its fire detector and commands its perception component to process the data, estimating the position of the potential fire alarms (TAKE_SHOT).

A grid representation is used to integrate measurements on time. Figure 112.9 shows the status of the grid in two instants of one particular experiment. After a while, several regions of high probability are detected. Each region is converted into a Gaussian representation for the position and then sent to the central station.

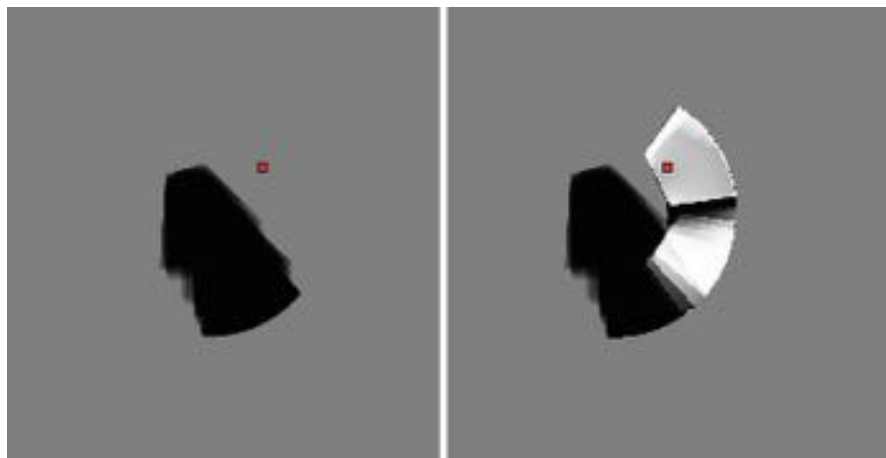


Fig. 112.9 Marvin employs a grid representation to integrate the data gathered by its fire sensor. Each cell of the grid represents an small portion of the scenario. The *red square* indicates the actual position of the fire. *Black* indicates a very low fire probability, while *white* indicates a high probability. High probability regions are extracted and approximated by Gaussian distributions. Due to false alarms, other high probability regions are also detected

Also, the mean probability of the cells of each region is used to estimate the probability that the region contains a fire.

Once an alarm (or several) has been detected, the central station aborts the current mission, and a fire confirmation phase starts (see Fig. 112.10). The first helicopter, Marvin, is moved to a safe place (GO_TO), and a second helicopter, Heliv, is sent to the place to confirm the alarm (GO_TO). Heliv carries an infrared camera, which is used to take images of the same zone where the alarms were detected (TAKE_SHOT). The images are segmented into potential fire objects, and these objects are geolocated. The information from Heliv is then fused with that from Marvin. If one of these objects is associated to an alarm already detected by Marvin, its position is used to reduce the uncertainty on the estimated position of the alarm. Moreover, the probability of being a real fire is increased for this alarm.

Moreover, the scheme allows using negative information. If one alarm is within the field of view of the Heliv's camera, but it is not associated to a segmented object, then its probability is decreased, although nothing can be said about its position. Figure 112.11 shows the potential alarms estimated by Marvin projected on the images of Heliv. It can be seen how the actual fire alarm is associated to the segmented fire object, while the false alarms do not correspond to fire regions. Eventually, the alarm is confirmed, when the current mission is finished.

Figure 112.12 shows the final trajectories of both UAVs and the final estimated position of the alarms. The estimated position of the actual fire alarms is within 1 m of the correct position. The evolution of the estimation on the position error is shown in Fig. 112.13. Also, the probability of the alarms is shown. It can be seen how the false alarms are discarded.

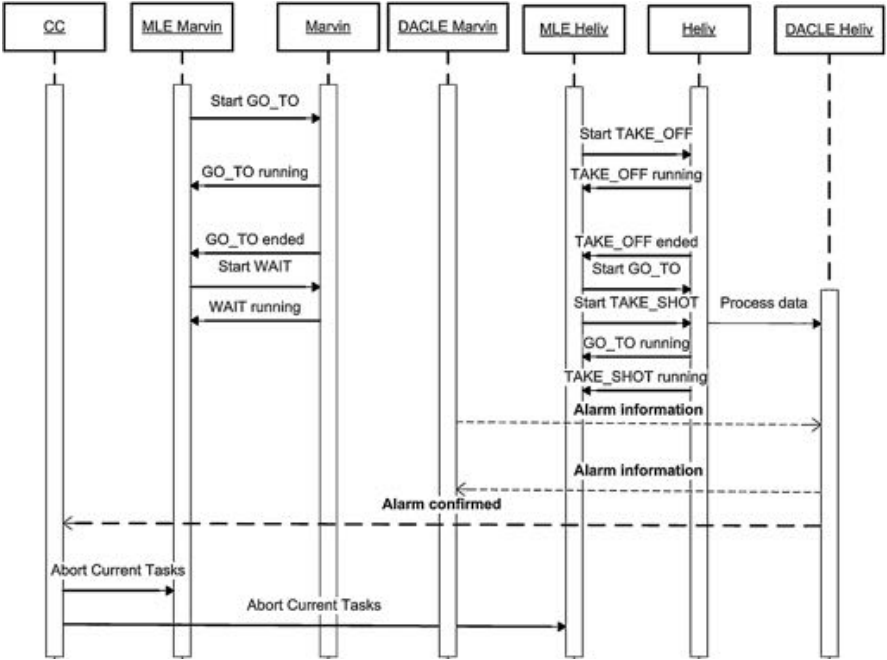


Fig. 112.10 Scheme of the exchange of tasks for the confirmation phase. Heliv is commanded to go to a close position of the detected alarm and to take and process images to confirm or discard the alarm. It uses the information provided by Marvin. Meanwhile, Marvin is commanded to wait at a safe place



Fig. 112.11 Heliv receives the information of the alarms detected by Marvin and fuses this information with the information obtained using its infrared camera. This figure shows the estimated position and uncertainty of the alarms detected by Marvin projected on the Heliv's infrared camera

112.4 Fire Monitoring and Measurement

Once a fire has been detected and confirmed, the firefighting activities begin. These fighting activities require to know, in real time, information about the evolution of the fire. In many cases, the lack of this information about the dynamic evolution

Fig. 112.12 Final complete trajectories of Marvin (*solid*) and Heliv (*dashed*). The ellipses represent the final estimated position of the different alarms. The *square* indicates the actual fire position. In the experiment, besides the correct alarm, two false alarms are generated. The scheme also shows the landing platform from where the UAVs take off

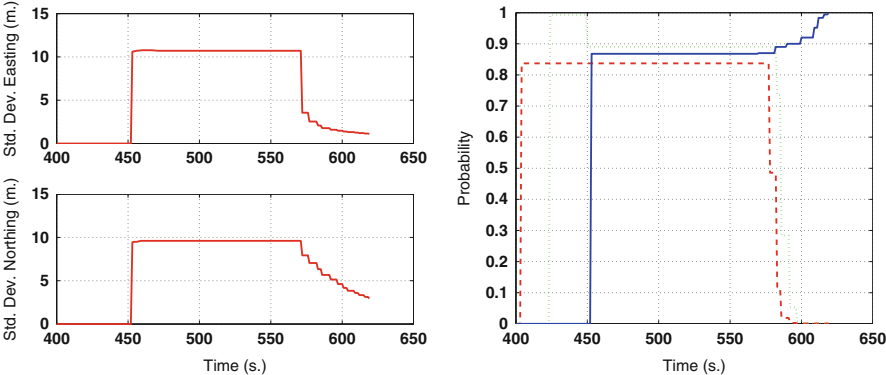
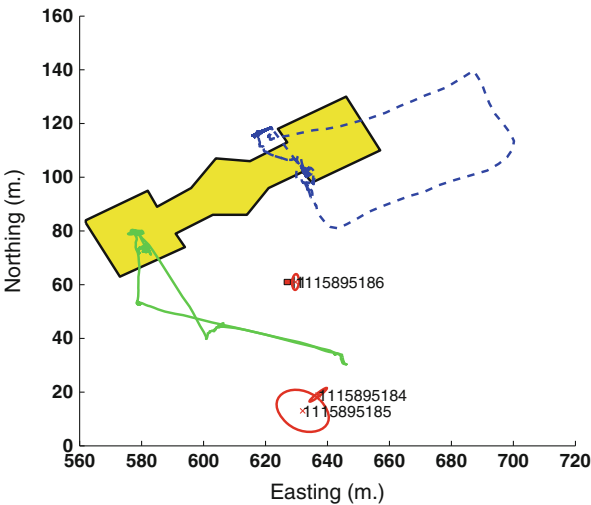


Fig. 112.13 *Left*: estimated standard deviation on the error. First, when Marvin detects the alarm the uncertainties are high due to the poor localization capabilities of the fire sensor. When Heliv fuses the information with its camera, the uncertainty decreases. Moreover, the probability of being an actual fire alarm is updated accordingly

of fire plays an important role in accidents that happen in firefighting activities. The objective of the fire monitoring missions is, then, to gather this information by using a team of UAS.

Figure 112.14 shows the state of an experimental controlled fire at several time instants. Among the most important parameters for firefighting management are the shape and position of the fire front, its rate of spread (how this front evolves with time), and the maximum height of the flames (Viegas 1998). If available, this information, integrated within a Geographical Information System (GIS), can be used by the fire brigades for firefighting planning, for instance, by predicting the potential evolution of the fire and determining the optimal location of fighting means.



Fig. 112.14 Different stages in the evolution of a fire

Fire monitoring is usually performed by experts that estimate, visually or from images gathered by cameras, the rate of spread and height of the flames. More recently, airborne systems are used in order to have a broad overview of the fire evolution, but still the monitoring activities are carried out by people. Satellite-based systems have been also proposed for forest fire monitoring (Chuvieco and Martin 1994; San Miguel Ayanz et al. 2005). The temporal and spatial resolutions of these systems are still very low for the requirements of forest firefighting in many cases. There are some fire analysis techniques based on computer vision for fires carried out in laboratories, as (Martínez de Dios et al. 2006; Pastora et al. 2006). However, the application of the same techniques to noncontrolled outdoor environments is not considered in those papers.

Here it is shown how a team of UAS can be also used for this mission. The objective of the fire monitoring mission is, therefore, to estimate spread of the fire front and other parameters in real time and to transmit this information to the firefighting brigades. The use of multiple UAS can provide several advantages: it is possible to devote a different UAS to different parts of a large fire, and/or the UAS can again cooperate to estimate the different parameters with more precision.



Fig. 112.15 Images from UAS onboard visual (*left*) and infrared (*right*) cameras taken during a fire monitoring experiment

112.4.1 Sensors

As in forest fire detection, monitoring and measurement require visual and infrared cameras as main perception sensors. Visual images can be used to obtain measurements of the base of the fire (location and shape). They can be also used to obtain flame measurements, but the accuracy in this case is often low due to the parallax problem. However, the fire can be occluded by the smoke in the visual images.

Infrared cameras are also useful for fire monitoring and measuring. The radiation intensity emitted by the base of the fire is considerably higher than that of the flames (Hudson 1969). Thus, infrared images can be used to obtain measurements of the fire base (location, shape) but are not useful for measuring the flames.

The fire measuring system can use thermal infrared cameras and also nonthermal cameras, which do not provide temperature measurements but do allow qualitative estimations of the radiation intensity. Although cameras in the mid-infrared spectral window [3–5] μm are preferred, the proposed system can indistinguishably use cameras in the mid-infrared or far-infrared windows. In fact, both types of cameras were used in most of the field experiments that have been carried out. For more details see Martínez-de Dios et al. (2011).

Figure 112.15 shows images of the same scene taken with a visual image and one infrared image during a forest fire measuring experiment.

112.4.2 Perception Techniques

112.4.2.1 Data Fusion

The system should consider all the information gathered by the different UAS to estimate the evolution of the fire applying data fusion techniques. The general picture is given by Fig. 112.16. In the system, each vehicle will locally process its images and provide features related to the fire front evolution. All this information

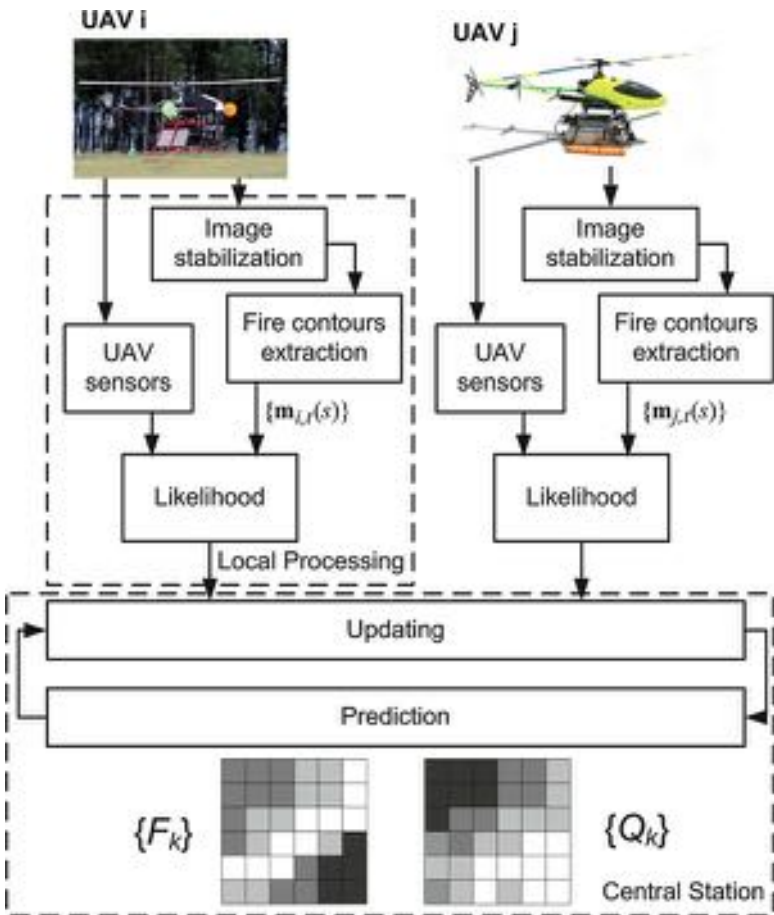


Fig. 112.16 The system estimates two probability grids (fire presence and fuel at each cell). The state of the grids is estimated incorporating data obtained from the fleet of UAVs. The prediction step incorporates the increase in uncertainty due to the motion of the fire

is received at the central station, in which the estimation takes place, taking into account all the data from the fleet.

Fire is dynamic and propagates, changing in size and shape. An evidence grid is chosen to represent the state of the fire at a given time instant. The area is divided into a rectangular grid, in which the state of each cell k is defined by two binary values: $F_{k,t} \in \{0,1\}$, indicating if there is fire in the cell, and $Q_{k,t} \in \{0,1\}$, indicating if the fuel in the cell is completely exhausted. The system cannot be certain about these values, and, therefore, two probability values $\{f_k, q_k\}$ are stored for each cell. These two values correspond to the probabilities $f_k = p(F_{k,t} = 1)$ (there is fire at a given cell) and $q_k = p(Q_{k,t} = 1)$ (the fuel in that cell is completely exhausted). Each cell also has an associated 3-D position, given by a

digital elevation map that is previously loaded in the system (and that can be the result of a previous mapping mission performed by a UAV Hygounenc et al. 2004).

The state of the cells is then recursively estimated from the information provided by the cameras onboard the UAS (see below) using a Bayesian filter. Prediction models can be used within the filter (Merino et al. 2012).

112.4.2.2 Information Extraction from the Images

The information about the fire front position is encoded in the images as the contour of fire regions on the image plane. The techniques described in Sect. 112.3.2 can be used to segment the fire objects on the images and to extract their contours. This contour is further characterized in order to distinguish the pixels of the contour related to the fire front and the pixels related to the top of the flames, therefore obtaining the height of the flames in pixel coordinates. The dynamic properties of the fire base and the flames are used for this characterization. The position of the fire-base pixels on the image plane generally changes more slowly than the position of the flame pixels (as the flames flicker). The application of a temporal low-pass filter over a sequence of consecutive segmented images is therefore used to filter out the flame pixels. Finally, these fire fronts are geolocalized to incorporate the information into the grid maps. Further information on these techniques can be found in Martínez de Dios et al. (2008).

112.4.2.3 Image Stabilization

Considering a UAS with hovering capabilities, unavoidable control errors, turbulence, and vibrations produce changes in the camera position which leads to image motion. This motion can affect to the previously described algorithms, and, therefore, it is necessary to cancel it. Electromechanic systems can be used to cancel vibrations, but these systems are usually heavy, expensive, and have a residual vibration.

Image processing procedures can be also used for software-based image motion estimation and cancellation. This can be achieved if the apparent motion between consecutive images is computed. In this system, a sparse image motion field is computed by a feature matching algorithm (see Fig. 112.17). Then, this sparse motion field is used to estimate a model of the motion of the complete image. Finally, the model is applied to all pixels to warp the current image to a common frame, therefore eliminating the background motion between the current image and the previous one. The details can be found at Merino et al. (2012).

112.4.3 Experimental Results

The fire monitoring system was also tested in the experiments mentioned in Sect. 112.3.3. The multi-UAS monitoring mission begins after the alarm has been confirmed and precisely localized as shown in this Sect. 112.3.3.

This fire monitoring mission is generated for Marvin, Heliv, and the airship Karma, involving synchronization tasks to take and process pictures of the event

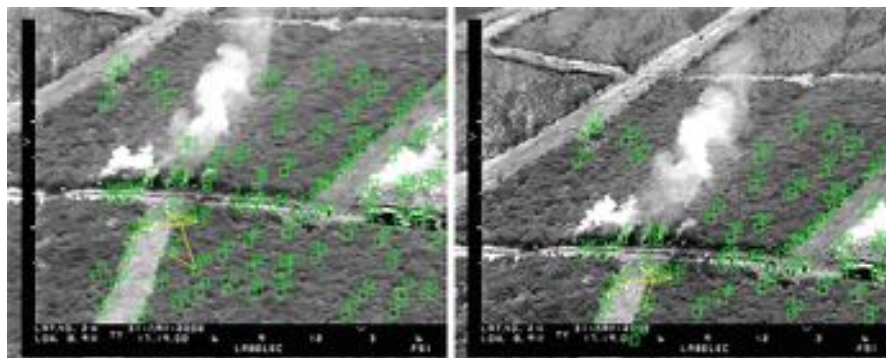


Fig. 112.17 Interest points extracted and tracked in two consecutive images to obtain a sparse image motion field

from the correct viewpoints at the same time (Fig. 112.18). The decisional layers of the vehicles (Lacroix et al. 2007) manage the task planning, allocation, and the synchronization signals between UAVs in a decentralized manner, and also the generation of the adequate viewpoints for monitoring the fire, covering the detected alarm from several directions (the vehicles are situated surrounding the fire with a relative orientation of 120°). When the synchronization is correctly achieved, Marvin and Heliv begin to obtain pictures of the fire simultaneously and for a given time. Also, Karma is commanded to take images of all the area from a high vantage point.

Each UAS, locally, stabilizes the images captured by its cameras and processes the results obtaining an estimation of the position of the fire front in pixel coordinates. The data from all the systems are sent to the central station, where all the information is fused for the estimation of the evolution of the fire. Figure 112.19 shows the images of the same fire from the three vehicles and how the fire contours are correctly extracted.

After the time for taking pictures has expired, Marvin and Heliv are commanded to return to home and to land. At the same time, a mapping mission is generated for Karma. Afterwards, the mission is terminated.

In the previous experiments the controlled fires performed are of small scale, and it is difficult to appreciate an evolution on the fire front, although Fig. 112.19 shows how the shape of the fire is correctly estimated. The techniques described have been also validated in many controlled medium-scale forest fire experiments carried out in Serra da Gestosa (Portugal). In these field experiments, square plots of up to 150 by 100 m were burned under controlled safety conditions. The rectangular plots burned in the experiments can be observed in the images. The experiments mobilized significant resources including 80 firemen and 5 firefighter trucks. A more detailed description of this kind of experiments can be found in Viegas et al. (2002).

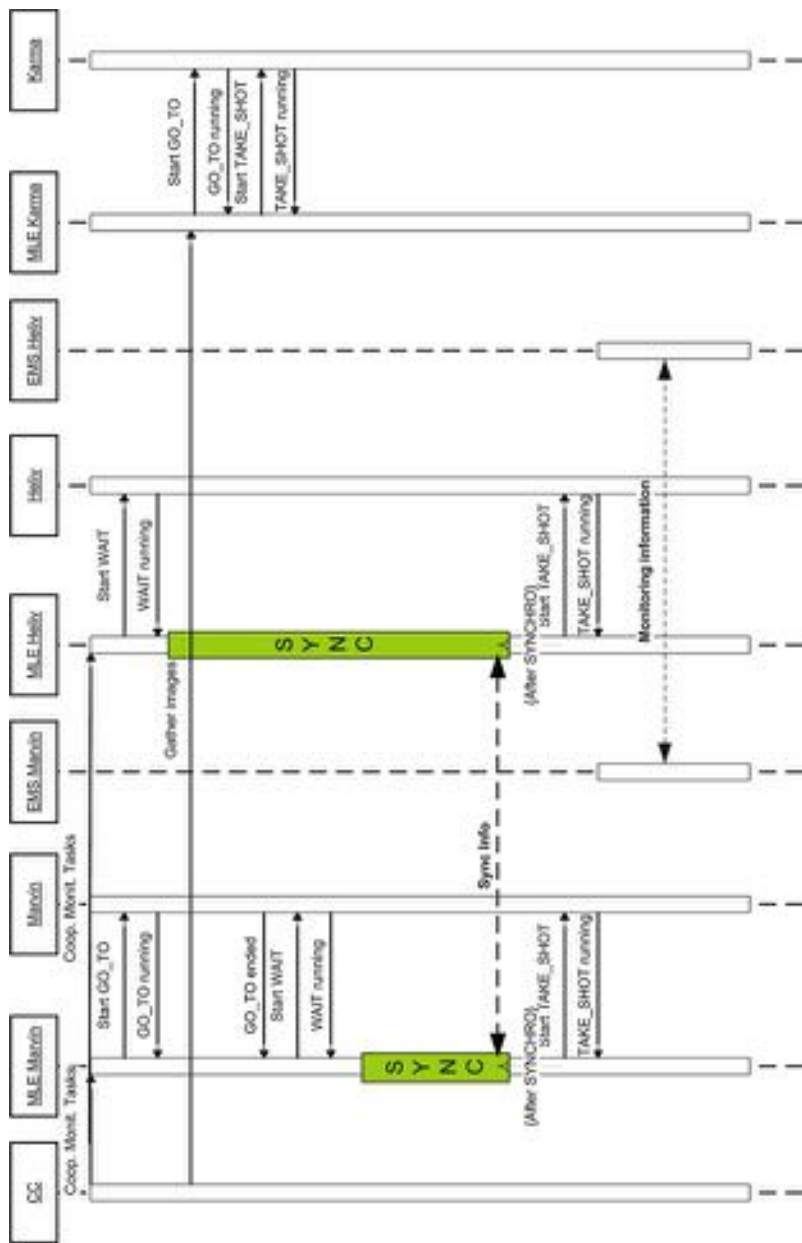


Fig. 112.18 Exchange of tasks for the monitoring phase. The UAVs can synchronize their tasks, and therefore they can begin the monitoring of the fire from different points of view at the same time. The synchronization signals are employed as preconditions for the tasks to be synchronized



Fig. 112.19 *Top: left, image from Heliv, after stabilization and feature extraction; middle, image from Marvin after stabilization and feature extraction; right, image from Karma. Bottom: details of the extracted contours. Green: fire front. Red: top of the flames*

Figures 112.20 and 112.21 show results of one of these experiments. In it, a plot of approximately 10,000 m² was burnt. In this case, a helicopter is commanded to gather aerial images.

Figure 112.20 shows several frames gathered from the helicopter, after the application of the image stabilization and feature extraction algorithms. It can be seen how the stabilization algorithm effectively removes the motion induced by the helicopter. It can be also seen how the fire front shape is adequately extracted from the images.

112.5 Cooperation of UAS with Ground Infrastructure for Fire Extinguishing Tasks

Fire extinguishing is a very dangerous task that has been traditionally carried out with low level of automation involving significant human resources. Although the idea of firefighting has been early proposed (Bradshaw 1991), few implementations have been carried out. The works developed have been focused more on the robotization of specific firefighting tasks rather in demonstrating a full robotized system for automatic firefighting.

The work (Su 2006) proposes an automatic multi-sensor fire detection system. The robot after detecting and localizing the fire extinguishes the fire autonomously. The Genbu robot (Kimura et al. 2001) is an articulated multi-wheeled firefighting mobile robot. This robot uses the fire hose's hydraulic energy as source of energy for its actuation. Fire trucks supply water at a high pressure through the fires hoses, so that the robot could powerfully thrust its way through the debris inside a fire scene.

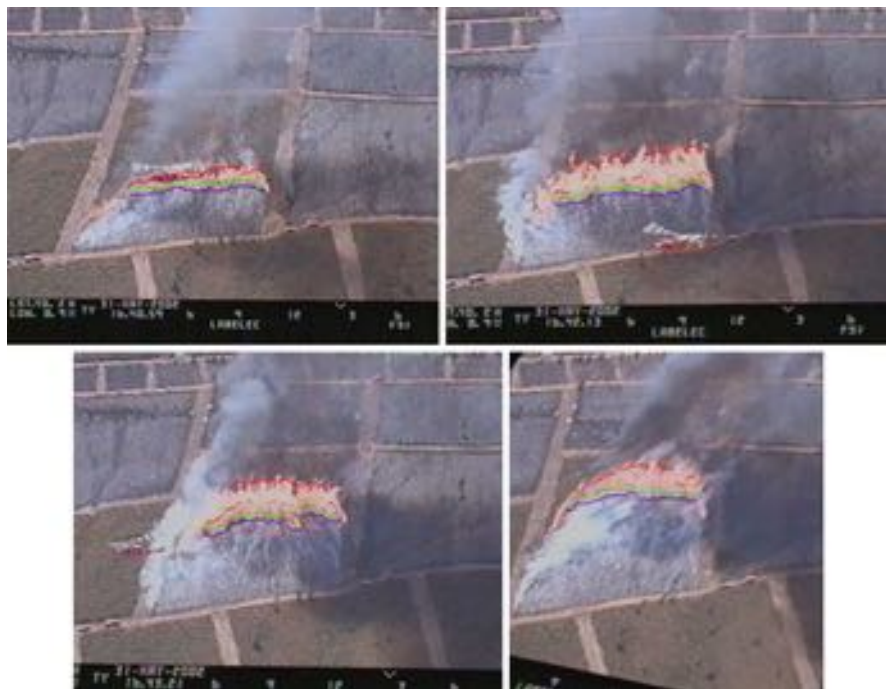


Fig. 112.20 Frames 3, 77, 144, and 219 of the aerial sequence gathered during the fire experiment 520, after automatic stabilization. The extracted fire front contour is also shown

The operation range of this robot is not sufficient for forest fire scenarios. Besides, this robot should involve human actuation since it is not equipped with a perception system that allows its fully autonomous operation.

Fire extinguishing takes place at the same time as fire monitoring. The information provided can be used to decide how to attack the fire. Fire extinguishing involves the use of water, chemicals, and other means. The UAS considered in the multi-UAS system do not have enough payload to carry the required actuators for this objective: cooperation with automatic ground means is necessary for this purpose. This section focuses on the cooperation of multi-UAS fleet with ground infrastructure. The integration of UAS with an automated fire extinguisher on a fire truck is presented. The cooperation of the multi-UAS system with Wireless Sensor and Actuator Networks (WSN) for fire extinguishing is also summarized. UAS integration with the fire extinguisher and with the WSN were tested in experiments with controlled real fires in close-to-operational firefighting conditions.

112.5.1 Sensors and Actuators

The multi-UAS team was integrated with devices that can be actuated to fight the fire. And thus, the network allows to communicate with a fire truck equipped with

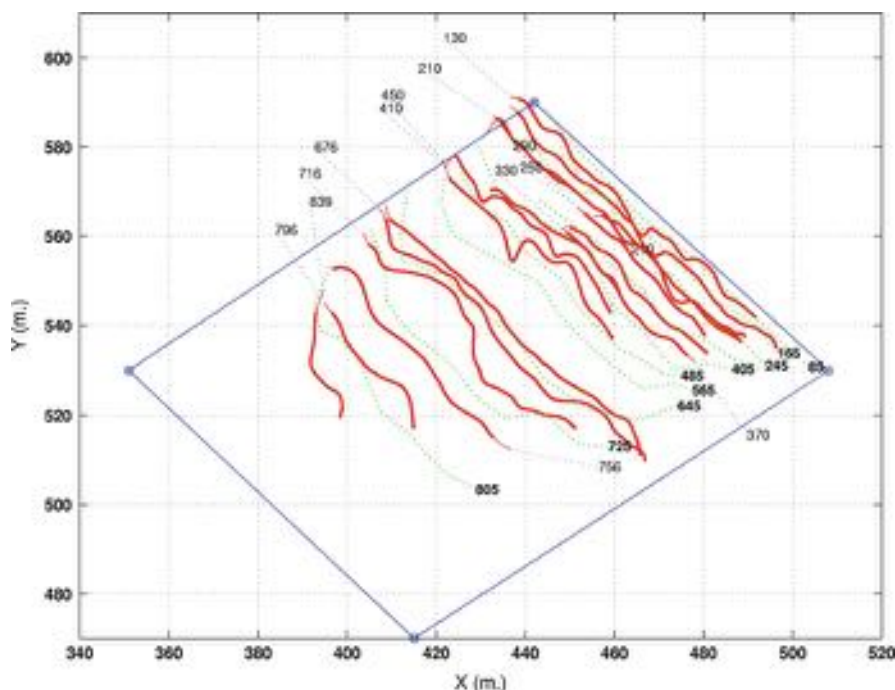


Fig. 112.21 Evolution of the fire front (in georeferenced coordinates) for plot 520 estimated from the images gathered by the helicopter (*solid*) and compared with the front obtained with a static camera (*dashed*)

an automated water cannon. The water cannon was an adaptation of a commercial system with manual operation. It could be pointed in pitch and jaw angles. Although the driving of the fire truck has to be done manually, the pointing, activation, and deactivation of the water cannon can be carried out automatically.

Accurate temperature and humidity measurements are very important in fire extinguishing. These readings cannot be obtained from the multi-UAS system, and in many cases local sensors are the only way to gather the measurements. For instance, making accurate temperature estimations from onboard infrared images requires knowing physical parameters such as emissivity of the material, ambient temperature, humidity, among others. In the approach presented here, UAS are used to deploy at interest locations Wireless Sensor Networks (WSN) nodes equipped with temperature and humidity sensors.

WSN are comprised of a large number of nodes with sensing, computing, and wireless communication capabilities. Low size, cost, and energy consumption are between their main features. A good number of cooperative methods have been developed in the recent years to endow WSN with high flexibility, robustness, and tolerance to failures. These features together with battery-based power, efficient and flexible deployment has extended WSN technology to a growing number of applications.

Cooperation between UAS and WSN provides interesting complementarities. Below are some examples. Autonomous deployment of WSN nodes using UAS for WSN repairing has been proposed in Corke et al. (2006). WSN data collection with a fixed-wing UAS has been proposed in Teh et al. (2008). A system in which a helicopter UAS is used for wireless powering and collecting a low number of nodes is described in Todd et al. (2007).

In this chapter, UAS-WSN integration for fire extinguishing is explored. UAS equipped with ad hoc mechanisms are used to deploy WSN nodes at interest locations in order to gather high-accuracy fire measurements necessary for firefighting.

112.5.2 Techniques

Fire extinguishing requires the fire detection and monitoring techniques described in previous sections. Besides, it requires extending the decisional system with modules to deal with the new components, like the fire extinguishing truck.

The water cannon pointing angles were computed by means of parabolic throw equations using the location of the fire computed by the multi-UAS fire perception system and the pose (location and orientation) of the fire truck. The truck was equipped with DGPS receiver and IMU. A simple protocol was implemented to integrate the fire truck operation within the fire detection and monitoring system. The protocol has commands for controlling the pitch and yaw angles and for activating and deactivating the water cannon. It should be noticed that the water cannon itself is not equipped with any sensor that allows it to know if the fire has been actually extinguished or not. It is the multi-UAS perception system which confirms extinction or corrects water cannon angles if necessary.

The integration of multi-UAS system and the WSN allows complementarities in firefighting tasks. In particular, UAS can be used to transport and accurately deploy WSN nodes at desired locations. UAS and WSN complement each other: WSN nodes provide accurate local information on the fire, while UAS obtains more global information. Also, the WSN nodes deployment locations can be determined dynamically depending on the fire evolution. This can also be interesting in case of risks of fire spreading to a dangerous area, e.g., a house in the wildland-urban interface, or to prioritize fire extinguishing efforts in presence of threats, e.g., barrels full of petrol in a factory environment.

Figure 112.22 shows an autonomous helicopter deploying WSN nodes. Although the UAV is just at 2 m while deploying the nodes, WSN housings are used for impact protection and ensure that the antenna is vertical after the deployment.

112.5.3 Experimental Results

The following shows an experiment of multi-UAS fire detection, monitoring, and extinguishing mission that was performed in 2009. The initial situation was as follows:



Fig. 112.22 Automatic deployment of WSN nodes using Marvin helicopter UAS

- A fire alarm was declared inside the building.
- Two UAS were ready to fly on the landing pads. UAS1 was equipped with one infrared camera and UAS2 equipped with the WSN node deployment device and three sensor nodes.

The multi-UAS fire confirmation task was triggered by the fire alarm. UAS1 took off. It detected the fire with the infrared images confirming the alarm and estimating its location. UAS1 also detected a number of fuel barrels near the building. As there was risk of fire propagation from the building to the fuel barrels, a WSN node deployment mission was specified in order to place several sensors in between at the locations of the waypoints wp1, wp2, and wp3 (see Fig. 112.23). According to this mission, UAS2 was launched, and it autonomously moved to the waypoints and deployed sensor nodes. Once deployed, the WSN node was properly integrated into the ground WSN and provided accurate readings to monitor the fire spread. After accomplishing the WSN node deployment mission, UAS2 landed.

Meanwhile UAS2, equipped with an onboard infrared camera, kept monitoring the fire. The fire monitoring system integrated infrared images gathered by UAS2 with the data from the WSN in order to obtain accurate fire location estimations. Once the fire was accurately located, the automatic extinguishing system was ready to operate. Then, the position of this fire was transmitted to proceed with the extinction, while the UAS1 hovered at a safe location. The fire extinguisher installed in a fire truck was commanded to point to the fire taking into account the fire

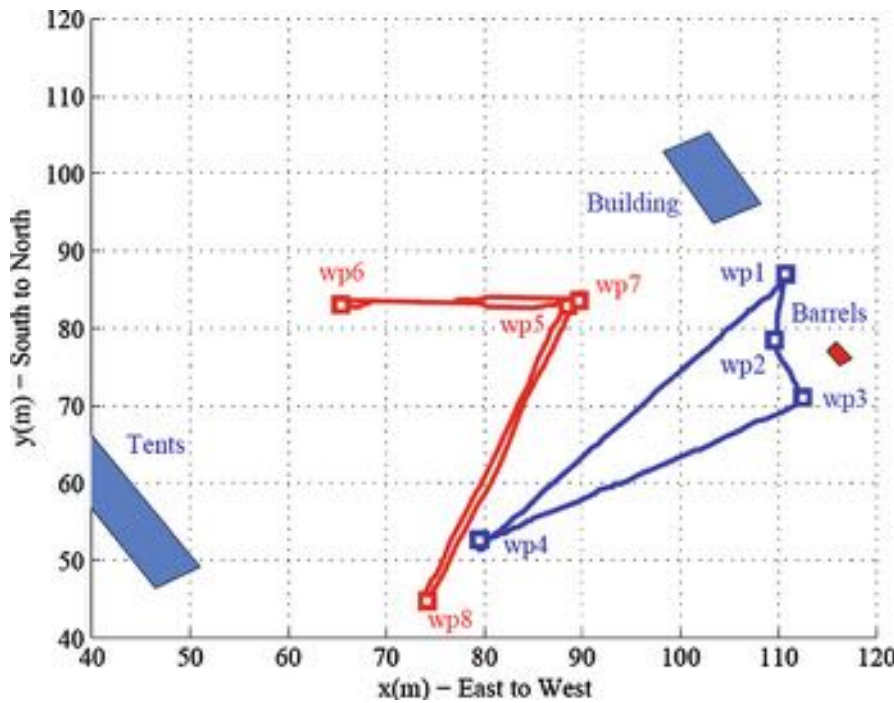


Fig. 112.23 Paths followed by UAS1 and UAS2 (red and blue, respectively) during the experiment

position, the positioning measurement of the truck GPS, and IMU data. The water cannon was activated, and the fire was extinguished, see Fig. 112.24. UAS2, which was waiting in hovering during the extinguishing, confirmed the extinction of the fire by means of the onboard infrared camera. The WSN nodes deployed near the fire also confirmed extinction and the mission finished.

112.6 Conclusions

The chapter describes the application of teams of UAS to several tasks related to forest firefighting, like fire detection, monitoring, and extinguishing. The main issues and techniques are described, and experimental results are shown, in which an actual fleet of UAS is tested in controlled setups involving fires.

The main conclusion of the chapter is that the use of UAS in firefighting is feasible, not facing any major unbearable problems. Of course, there are still many issues to be solved for a commercial application. One of them is the integration of UAS with manned aircraft usually present in firefighting operations. This is both a legal and a technical problem (► [Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States](#)). Here it should be noted that manned aircraft cannot operate by night, and then there are no conflicts with the



Fig. 112.24 Picture of the fire truck with the implemented fire extinguishing system during the experiment

UAS operation. Moreover, the weather conditions usually involve high wind gusts, and harsh conditions, which require the development of more robust UAVs and controllers. The application of firefighting also need of UAS with higher endurance and larger payloads. Nowadays there are UAS, mainly developed in the military field, that have the required endurance, and therefore the extension of the techniques for their use by environment management agencies seems affordable ([►Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States](#)).

Another issue is the particular UAS platform to be used in different applications. For the tasks of fire detection or large-scale fire monitoring, airplanes are more suitable than hovering-capable UAS. Also airships can be considered, although they should be big enough in order to withstand winds. On the other hand, small UAS with hovering capabilities could be employed by fire brigades for, at least, obtaining views of areas difficult to be accessed.

An additional question is if the use of several cooperating UAS offers advantages when comparing with the use of one complex UAS. For the task of fire detection (or in general, for exploration activities), a fleet of vehicles adds two different advantages. Firstly, the time needed to cover wide areas can be reduced. Also, the complementarities between UAS and their payload can help to reduce the false alarm ratio, which is one of the main problems with currently employed detection systems. For the case of fire monitoring, employing several UAS allows obtaining different and complementary views. Moreover, the application of teams of UAVs can be useful to perform simultaneously not only fire detection and monitoring but

also to have communication relays and to acquire high-resolution meteorological information. Future applications can include the transportation and deployment of a big payload by means of several vehicles as shown in ► [Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation](#).

The use of fleets of UAS requires a higher degree of autonomy on the UAS. Controlling the fleet would require more complex control centers or an amount of people if the UAS are not endowed with autonomous decisional capabilities. In this case, also decentralized autonomous perception functions are required in order to reduce the computational power needed to process the huge amount of data from the different vehicles of the team.

References

- B. Arrue, A. Ollero, J. Martínez de Dios, An intelligent system for false alarm reduction in infrared forest-fire detection. *IEEE Intell. Syst.* **15**(3), 64–73 (2000)
- A. Bradshaw, The UK security and fire fighting advanced robot project, in *IEE Colloquium on Advanced Robotic Initiatives in the UK*, London, 1991
- M. Bryson, A. Reid, F. Ramos, S. Sukkarieh, Airborne vision-based mapping and classification of large farmland environments. *J. Field Robot.* **27**, 632–655 (2010)
- D. Campbell, W.G. Born, J. Beck, B. Bereska, K. Frederick, S. Hua, Airborne wildfire intelligence system: a decision support tool for wildland fire managers in Alberta, in *Proceedings of the SPIE, Thermosense XXIV*, Orlando, vol. 4710, 2002, pp. 159–170
- E. Chuvieco, P. Martin, A simple method for fire growth mapping using AVHRR channel 3 data. *Int. J. Remote Sens.* **15**, 3141–3146 (1994)
- P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, G. Sukhatme, Deployment and connectivity repair of a sensor net with a flying robot, in *Experimental Robotics IX*, ed. by H.A. Marcelo Jr., K. Oussama. Springer Tracks on Advanced Robotics (Springer, New York, 2006)
- J.S. de Vries, R.A. Kemp, Results with a multispectral autonomous wildfire detection system, in *Proceedings of the SPIE Infrared Technology XX*, San Diego, vol. 2269, 1994, pp. 18–28
- E. Den Breejen, M. Breuers, F. Cremer, R. Kemp, M. Roos, K. Schutte, J. De Vries, Autonomous forest fire detection, in *Proceedings of the 3rd International Conference on Forest Fire Research*, Luso, Portugal, 1998, pp. 2003–2012
- D. Dierre, H. Hoff, M. Bouchet, RAPSODI: rapid smoke detection and forest fire control, in *International Symposium on Forest Fire: Needs and Innovations*, Athens, 1999, pp. 415–419
- F. Gómez Rodríguez, S. Pascual Peña, B. Arrue, A. Ollero, Smoke detection using image processing, in *Proceedings of the IV International Congress on Forest Fire Research ICFRR*, Coimbra, 2002
- R.D. Hudson, *Infrared System Engineering*. Wiley Series in Pure and Applied Optics (Wiley, New York, 1969)
- E. Hygounenc, I.-K. Jung, P. Soueres, S. Lacroix, The autonomous blimp project of LAAS-CNRS: achievements in flight control and terrain mapping. *Int. J. Robot. Res.* **23**(4–5), 473–511 (2004)
- V. Kelhä, Y. Rauste, T. Häme, T. Sephton, A. Buongiorno, O. Frauenberger, K. Soini, A. Venäläinen, J. San-Miguel-Ayanz, T. Vainio, Combining AVHRR and ATSR satellite sensor data for operational boreal forest fire detection. *Int. J. Remote Sens.* **24**(8), 1691–1708 (2003)
- H. Kimura, K. Nakaya, S. Hirose, Development of genbu: articulated multi-wheeled mobile robot, in *Proceedings of the TITech COE/Super Mechano-Systems Symposium*, Tokio, Japan, 2001
- S. Lacroix, R. Alami, T. Lemaire, G. Hattenberger, J. Gancet, Decision making in multi-UAV systems: architecture and algorithms, in *Multiple Heterogeneous Unmanned Aerial Vehicles*, ed. by O. Anfal, M. Iván. Springer Tracks on Advanced Robotics (Springer, Berlin, 2007)

- J. Martínez-de Dios, L. Merino, A. Ollero, Fire detection using autonomous aerial vehicles with infrared and visual cameras, in *Proceedings of the 16th IFAC World Congress*, Prague, 2005
- J. Martínez de Dios, J. André, J.C. Gonçalves, B. Arrue, A. Ollero, D. Viegas, Laboratory fire spread analysis using visual and infrared images. *Int. J. Wildland Fire* **15**, 175–186 (2006)
- J. Martínez de Dios, B. Arrue, L. Merino, A. Ollero, F. Gómez-Rodríguez, Computer vision techniques for forest fire perception. *Image Vis. Comput.* **26**(4), 550–562 (2008)
- J.R. Martínez-de Dios, L. Merino, F. Caballero, A. Ollero, Automatic forest-fire measuring using ground stations and unmanned aerial systems. *Sensors* **11**(6), 6328–6353 (2011)
- I. Maza, F. Caballero, J. Capitan, J.M. de Dios, A. Ollero, A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities. *J. Field Robot.* **28**(3), 303–328 (2011)
- L. Merino, F. Caballero, J. Martínez de Dios, J. Ferruz, A. Ollero, A cooperative perception system for multiple UAVs: application to automatic detection of forest fires. *J. Field Robot.* **23**(3–4), 165–184 (2006)
- L. Merino, F. Caballero, J. Ferruz, J. Wiklund, A. Ollero, Multi-UAV cooperative perception techniques, in *Multiple Heterogeneous Unmanned Aerial Vehicles*. Aníbal Ollero and Iván Maza. Springer Tracks on Advanced Robotics (Springer, Berlin, 2007)
- L. Merino, F. Caballero, J.M. de Dios, I. Maza, A. Ollero, An unmanned aerial system for automatic forest fire monitoring and measurement. *J. Intell. Robot. Syst.* **65**, 533–548 (2012)
- A. Ollero, L. Merino, Control and perception techniques for aerial robotics. *Ann. Rev. Control* **28**, 167–178 (2004). Elsevier (Francia)
- A. Ollero, I. Maza (eds.), *Multiple Heterogeneous Unmanned Aerial Vehicles*, Volume 37 of Springer Tracks on Advanced Robotics (Springer, Berlin, 2007)
- A. Ollero, S. Lacroix, L. Merino, J. Gancet, J. Wiklund, V. Remuss, I. Veiga, L.G. Gutiérrez, D.X. Viegas, M. González, A. Mallet, R. Alami, R. Chatila, G. Hommel, F. Colmenero, B. Arrue, J. Ferruz, J. Martínez de Dios, F. Caballero, Multiple eyes in the sky: architecture and perception issues in the COMETS unmanned air vehicles project. *IEEE Robot. Autom. Mag.* **12**(2), 46–57 (2005)
- E. Pastora, A. Águeda, J. Andrade-Cetto, M. Munoz, Y. Pérez, E. Planas, Computing the rate of spread of linear flame fronts by thermal image processing. *Fire Saf. J.* **41**, 569–579 (2006)
- J. San Miguel Ayanz, N. Ravail, V. Kelha, A. Ollero, Active fire detection for fire emergency management: potential and limitations for the operational use of remote sensing. *Nat. Hazards* **35**(3), 361–376 (2005)
- K. Su, Automatic fire detection system using adaptive fusion algorithm for fire fighting robot, in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Taipei, 2006, pp. 966–971
- S. Teh, L. Mejias, P. Corke, W. Hu, Experiments in integrating autonomous uninhabited aerial vehicles (uavs) and wireless sensor networks, in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA'08)*, Brisbane, Canberra, 2008
- M. Todd, D. Mascarenas, E. Flynn, T. Rosing, B. Lee, D. Musiani, S. Dasgupta, S. Kpotufe, D. Hsu, R. Gupta, G. Park, T. Overly, M. Nothnagel, C. Farrar, A different approach to sensor networking for shm: remote powering and interrogation with unmanned aerial vehicles, in *Proceedings of the 6th International Workshop on Structural Health Monitoring*, Stanford, 2007
- A. Utkin, A. Fernandes, A.V. Lavrov, F. Simões, R. Vilar, Feasibility of forest-fire smoke detection using lidar. *Int. J. Wildland Fire* **12**(2), 159–166 (2003)
- D. Viegas, Forest fire propagation. *Phil. Trans. R. Soc. Lond. A* **356**, 2907–2928 (1998)
- D.X. Viegas, M. Cruz, L. Ribeiro, A. Silva, A. Ollero, B. Arrue, J. Martínez de Dios, F. Gómez-Rodríguez, L. Merino, A. Miranda, P. Santos, Gestosa fire spread experiments, in *Proceedings of the IV International Congress on Forest Fire Research (ICFFR)*, Coimbra, 2002, pp. 1–13

Selection of Appropriate Class UAS/Sensors to Support Fire Monitoring: Experiences in the United States

113

Vincent G. Ambrosia and Thomas Zajkowski

Contents

| | | |
|---------|--|------|
| 113.1 | Introduction | 2724 |
| 113.2 | Fire Physics: Dictating Spectral Optimization for Sensing Systems | 2725 |
| 113.2.1 | Plume Characteristics Affecting UAS Fire Sensing | 2727 |
| 113.2.2 | Fire Weather Affecting UAS Fire Sensing | 2727 |
| 113.2.3 | Day vs. Night Fire Behavior Differences Affecting UAS Sensor Operations | 2728 |
| 113.3 | Fire Airspace Issues | 2728 |
| 113.4 | UAS Platform Decisions: Tactical vs. Strategic | 2730 |
| 113.4.1 | Strategic (MALE/HALE) UAS | 2731 |
| 113.4.2 | Tactical (LALE/LASE) UAS | 2732 |
| 113.5 | UAS Fire Imaging Sensor Characteristics | 2732 |
| 113.5.1 | Strategic Sensor Systems | 2733 |
| 113.5.2 | Tactical Sensor Systems | 2737 |
| 113.6 | History of Airborne Fire Imaging Systems Utility in United States | 2740 |
| 113.6.1 | History of UAS Fire Imaging Systems Developments in the United States | 2742 |
| 113.7 | Concluding Remarks | 2751 |
| | References | 2752 |

Abstract

As government agencies around the globe strive to improve their disaster monitoring capabilities and speed response and recovery efforts, requirements for more accurate and safer data collection mean that Unmanned Aerial System

V.G. Ambrosia (✉)

NASA-Ames Research Center, California State University – Monterey Bay, Moffett Field,
CA, USA

e-mail: Vincent.g.ambrosia@nasa.gov

T. Zajkowski

Next Gen Air Transportation (NGAT) Center, Institute for Transportation Research and
Education, North Carolina State University, Raleigh, NC, USA

e-mail: thomaszajkowski@gmail.com

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,
DOI 10.1007/978-90-481-9707-1_73,

© Springer Science+Business Media Dordrecht 2015

2723

(UAS) are being examined more closely for their potential roles in safely gathering data over those events. Wildfires are consistently one of the most common worldwide “disaster” events, and therefore the wildfire management community is experiencing a surge in UAS interest to support intelligence gathering and pertinent data delivery over those fire incidents. There are numerous, often confusing differences amongst UAS platforms and their sensors. Coupled with the various mission roles required by observation platforms in the wildfire domain, the UAS landscape can be a daunting place for a wildfire manager (or any disaster manager) to select the appropriate UAS solution capability that matches his/her requirements. This chapter describes some of the UAS lessons learned by both the U.S.. Forest Service (USFS) and the National Aeronautics and Space Administration (NASA) during their +10-year efforts on identifying appropriate UAS wildfire observation capabilities. The agencies, along with the UAS community, have demonstrated the use of both Low-Altitude, Short-Endurance (LASE) and Medium-Altitude, Long-Endurance (MALE) platforms for wildfire sensor intelligence gathering. Those experiences are identified in this chapter as well as an examination of the future direction of UAS systems for supporting the wildfire management community.

113.1 Introduction

The use of UAS systems to support disaster monitoring is the perfect civilian niche for the platform capabilities. Airborne assets have been used to provide remote sensing-derived information in support of wildfire events since the advent of the airplane. The USFS started using manned aircraft to provide information on fire activity in the vast western U.S. forests during the 1920s and 1930s, first through sketch-mapping fire locations on paper maps; later, camera systems were employed to image and then transfer to paper maps for delivery when the aircraft landed. As imaging systems improved and with the advent of thermal scanning systems more readily available in the late 1960s, a leap forward in technology provided the wildfire management community with more reliable fire imaging capabilities. With the introduction of Unmanned Aerial Systems (UAS), more disaster management agencies are starting to examine the use of UAS in lieu of manned assets to perform these hazardous tasks. As UAS platforms have improved and operations have been simplified, there is keen interest in the disaster community, and specifically the wildfire management community, in employing these resources in the wildfire battlefield.

This chapter will provide a perspective on the use of UAS systems to support wildfire observations, including the physics behind fire that dictate the appropriate sensor design characteristics, UAS platform characteristics, the airspace issues regarding UAS operations over fire, and the history of wildfire imaging UAS use in the United States. This information will improve the understanding of UAS platforms and appropriate sensor systems, to ease the transition into operational utility.

113.2 Fire Physics: Dictating Spectral Optimization for Sensing Systems

In order to employ the proper remote sensing sensor system for observing, it is important to understand the emittance characteristics of fire. Fires and combustion materials emit radiation. The Stefan-Boltzmann Law states that hot “radiators” emit more energy per unit area than cooler ones (this applies to thermal wavelengths shorter than microwaves) (Wolfe and Zissis 1989). The magnitude of emitted energy rises rapidly with increasing temperature of an object. Then, Wien’s Displacement Law states that the spectral distribution (wavelength) of that emitted energy varies with temperature. The energy from fires is emitted as electromagnetic energy within specific wavelength bands. Most of the energy emitted by heat from wildland fires is in the mid-wavelength infrared (MWIR, 3.0–5.0 μm) and long-wavelength infrared (LWIR, 8.0–14.0 μm) portion of the spectrum (Ambrosia and Brass 1988; Zajkowski et al. 2011). While the human eye detects energy in the short wavelength range of 0.3–0.7 microns (μm), energy from fires is emitted at wavelengths an order of magnitude longer (nominally, 2–14 μm) (Table 113.1). Most heat, or fire-mapping applications utilize LWIR sensors. The higher the temperature of the heat source, the more energy is emitted and the shorter the wavelengths that can be used to detect and map that source. In essence, fire detection lies in measuring radiance emitted in the mid-infrared region for typical encountered fire temperatures (Fig. 113.1). Although the spectral “range” for useful fire detection at MWIR is \sim 3.0–5.0 μm , an optimum center-wave measurement at 4.0 μm avoids the variable water vapor absorption frequencies at lower portions of that region and reduces reflected solar radiation influences to the 4.0 μm region by 40 % (Kaufman et al. 1998) (Fig. 113.2).

The two most common wavelength bands for fire remote sensing are 3–5 and 8–14 μm . These wavelength regions are consistent with both the wavelengths of typical fire sources and good atmospheric windows. Detectors or sensors that operate at shorter wavelengths “see” reflected energy and not emitted energy; this, coupled with atmospheric attenuation, limits the utility of shorter wavelength or optical systems to map fire emissivity.

Infrared energy and the sensors used to detect and map heat sources have limitations. These limitations may generally be broken down into a few categories: solar radiation effects, source temperature, water vapor attenuation, and saturation. Solar radiation (reflected sunlight) may mix with the emitted infrared energy;

Table 113.1 Wavelengths and temperature relationships for nominal fire condition classes

| Source | Temperature ($^{\circ}\text{C}$) | Nominal wavelength (μm) |
|---------------|------------------------------------|--------------------------------------|
| Background | 25 | 10.0 |
| Fuel ignition | 275 | 5.0 |
| Glowing | 550 | 4.0 |
| Cool fire | 725 | 3.0 |
| Hot fire | 1,200 | 2.0 |

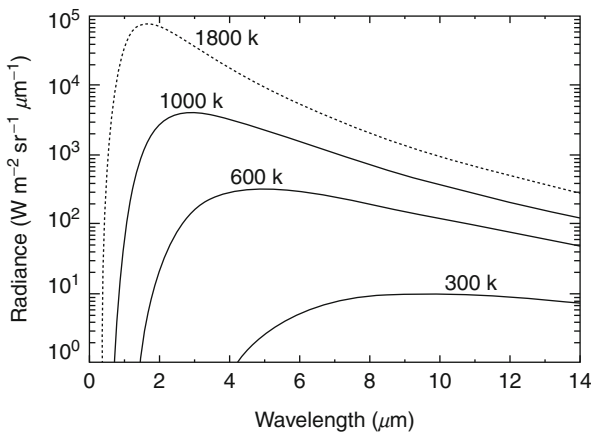


Fig. 113.1 Blackbody curves for wavelength regions for given temperature peaks. Typical background land surfaces are at ~300 K, while a smoldering fire can be at ~600 K and a flaming fire can be at 1,000 K. Therefore, fire measurements are best optimized at a wavelength region ~4 μm (Courtesy of/derived from Wolfe and Zissis 1989)

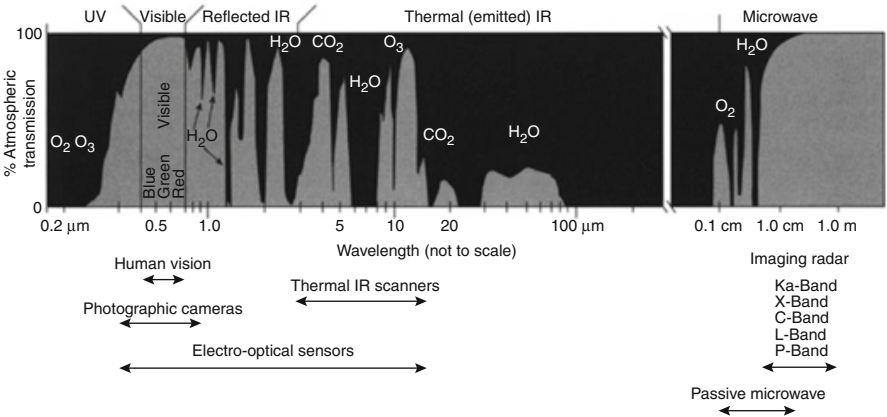


Fig. 113.2 Thermal infrared energy emitted by earth features ranges from 3 to 14 μm. Fires are detectable in this range, but various absorption frequencies and atmospheric effects restrict the useable “clear” windows to 3–5 and 8–14 μm (Courtesy of/derived from Kaufman et al. 1998)

thus, highly reflective surfaces (rock outcrops, water, snow, bare ground, metal roofs) may appear “hot” in thermal IR image. This can create false-positives (i.e., the sensor detects fire where there isn’t one). This problem can be mitigated by utilizing more than one thermal band to reduce false-positives. Often LWIR data is collected to portray background areas around the heat source. Because smoke does not prevent this long wavelength energy from passing through, one can discern the background (surface features) at these wavelengths. Visible wavelength sensors (aerial photography and visible bands of sensors) cannot see through the

smoke because it scatters and absorbs short-wavelength energy in those wavelength regions, much like the human eye cannot see through smoke. Fires are not perfect emitters of infrared energy. Emitted energy can be absorbed by a heavy canopy or overstory above the fire/thermal emitter. The ability to detect heat beneath a canopy may be reduced by heavy (moist) smoke, vegetation, or water vapor.

113.2.1 Plume Characteristics Affecting UAS Fire Sensing

Fire imaging in discreet thermal channels allows avoidance of spectral obscuration from smoke plume events. Smoke plumes (if minimal/no moisture is present in plume) are transparent in the MWIR and LWIR regions, though visible in shorter wavelengths (near infrared (NIR, 0.7–0.9 μm)) and visible wavelengths. This information is critical to selection of appropriate sensor systems for fire observations; the use of visible or NIR systems is generally inadequate for fire imaging purposes, since the particle sizes that constitute the plume attenuate the signal in those general wavelength regions. Results from a study by Uthe and colleagues (1982) indicate significantly greater plume attenuation at 0.53- μm wavelength than at 1.06 μm , indicating ~ 0.1 - μm mean particle diameters or the presence of gaseous constituents that absorb the visible radiation. The inclusion of black carbon particles in plumes also reduces the opacity index of visible and NIR imaging systems (Mazurek et al. 1991; Wong and Li 2002). Most small UAS (sUAS) systems are integrated with a visible wavelength camera system and a wideband LWIR camera, both of which can prove problematic when imaging wildfires.

113.2.2 Fire Weather Affecting UAS Fire Sensing

Dynamic fire weather conditions exist on every fire and can impact/hinder UAS observation missions. Due to the energy released by a fire, wind conditions can be dangerous to aircraft. Since the plumes over fires are composed of various organic compounds and carbonaceous materials, they pose a significant risk to gas-powered aircraft systems (not significant for electric motor systems). The ingestion of “dirty” air into gas-powered combustion engines can cause significant engine failure or damage internal engine components. This issue also hinders sUAS operations around fire regimes, especially due to their low flight altitude profiles where those constituents are most dense. There are significant updrafts and downdrafts that can affect flight performance and stability of a sUAS platform (or manned platform for that matter) and be a detriment to direct, overhead imaging operations.

As highly energetic fires produce convective systems, those systems can/will contain moisture. The presence of clouds or moisture (fog, rain, etc.) will obscure visible-NIR, MWIR, and LWIR observations of surface conditions and render those channels of limited use for fire observations. This would limit sUAS to less rigorous fire imaging assignments, such as post-fire hot-spot mop-up operations or operations during the night, when general fire weather conditions are usually less intense.

113.2.3 Day vs. Night Fire Behavior Differences Affecting UAS Sensor Operations

Fire behavior is markedly different between day and night. Early afternoon is generally the peak burning period when fires will burn the most intense, spread most rapidly, and tend to exhibit erratic fire behavior, because all of the weather elements are at the point where their influence on fire behavior is the greatest: radiation from the sun is at its maximum when the sun is directly overhead, the fire temperature is at its peak, relative humidity is at its lowest point, wind speed is at its maximum, wind direction is most variable, the atmosphere is the most unstable, and fuel will be the driest. Although this midday period may be the greatest time of need for fire behavior information provided by UAS remote sensing systems, it is also the most hazardous time for operations for small UAS platforms.

At night, the high-risk fire variables mentioned above reduce significantly in severity. Given that tactical aerial operations cease operations at night due to safety, it is a logical time for operations to integrate UAS into a less congested airspace under more benign fire weather conditions. Complex terrain combined with smoky conditions presents a flight safety hazard for nighttime-manned aircraft operations; this is especially true for rotorcraft systems that operate in close proximity to the ground. These operation periods are best handled by sUAS.

113.3 Fire Airspace Issues

The airspace over large wildfires can become very congested with multiple aircraft operating in or transitioning through the fire airspace working environment. Fixed-wing and rotorcraft platforms are used to deliver firefighters and supplies to the fire line in addition to those that are dropping water or retardant on the fire itself. In addition to these platforms, media and law enforcement aircraft are often allowed and present in the airspace as well. Additionally, a lead plane and an air-attack aircraft are almost always dispatched to fires and operate as observation platforms to control aircraft airspace, operating as an air-traffic manager for those fire events.

Most wildfires have segregated airspace designated for them at the onset of the fire, to allow firefighting assets to operate without interference from other aircraft in their vicinity. The FAA routinely grants Temporary Flight Restrictions (TFRs) to segregate aircraft and allow control of the area to the incident Air Operations Unit. Adding UAS into this mix is seen as disruptive to many current wildfire management personnel and other aircraft operators/pilots. How and when UAS are integrated into the airspace above a fire, as well as the altitude regimes, times, and locations of UAS fire flight integration, are still being evaluated by the USFS. To date, most of the UAS support on wildfire events (NASA Ikhana, USAF/Navy Global Hawk, CBP Predator-B) has been “strategic,” with those UAS operating well above the TFR and not interfering with normal wildfire aircraft operational altitudes. By flying at night or under conditions too hazardous for manned aircraft, UAS/manned

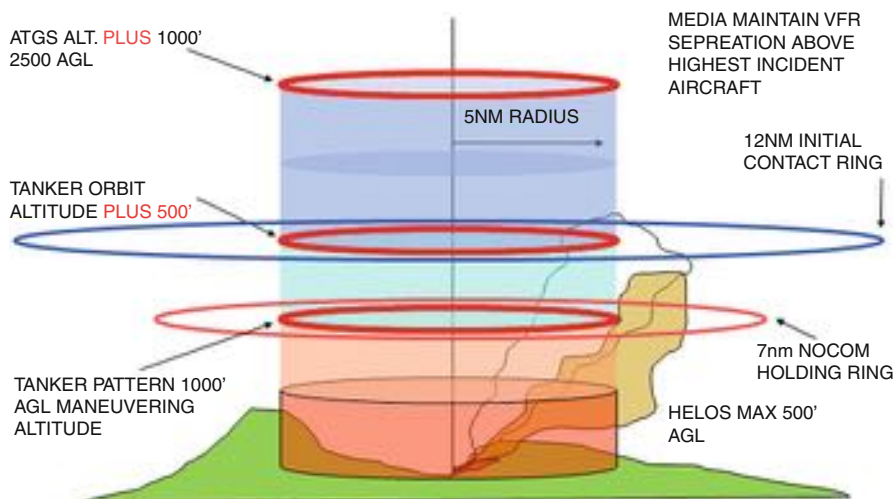


Fig. 113.3 The Fire Traffic Area defines when and where firefighting aircraft will contact air attack as well as approximate holding and maneuvering altitudes. The FTA can be expanded and modified as need to accommodate the size of the fire, local terrain, and airspace considerations. UAS integrating into the FTA must be able to maintain separations and follow the required airspace directions and procedures, like any manned aircraft in the FTA (Derived from: Interagency Aviation Management Council 2003)

platform “conflicts” are reduced, and UAS have the ability to satisfy requirements for supplying fire information when it is usually not available to management teams.

The Fire Traffic Area (FTA) developed as a guideline airspace management directive by the USFS Fire and Aviation Management is designed to identify the “rules” regarding general fire airspace separation/operations altitudes of various platforms. The FTA provides a standardized initial attack airspace structure to enhance air-traffic separation over wildland fire (or other) incidents (Interagency Aviation Management Council 2003). The FTA is constructed within the TFR that covers most large wildfires. The FTA separates aircraft horizontally and vertically with clear lines of communication amongst the traffic managers and the flight crews. If UAS are integrated into a FTA while other airborne platforms are present, the UAS pilot will need to communicate with the air attack officer constantly. In addition, pilots of the manned aircraft will need to know where the UAS is operating and avoid that location. As can be seen in Fig. 113.3, there are some key niche areas that sUAS may be able to operate in, between other working manned assets in the FTA.

A common misperception associated with integrating UAS into fire monitoring airspace is that the TFR over a fire “sterilizes” the airspace over that event and that a Certificate of Authorization (COA) from the FAA is not needed for UAS operations. This is erroneous; the TFRs that typically cover fires are defined in FAA regulations, FAR Part 91.137(a), (2) and allow for aircraft that are not participating in fire operations and are not under control of the air-attack supervisor to enter

that TFR. These intrusion allowances include law enforcement aircraft, aircraft on approved instrument flight rules flight plans, accredited news media, and aircraft that are based at airfields within the TFR. In the simplest term, a COA is a project aviation safety plan on how to mitigate the risk associated with operation of a UAS for a very specific location (and type/tail number of UAS). This process is constantly being streamlined, but it is time consuming to collect and enter/provide the required COA application information to the FAA, and it takes approximately 60 days to get approval from the FAA to operate in that region. This is not conducive to fast-moving wildfires and short-term UAS monitoring needs. Once approved, the COA holder can use their existing COA to get an emergency COA for another location with similar airspace approved quickly; NASA was able to obtain an emergency COA within a few hours to support the 2006, 2007, and 2008 California wildfires that were located outside of its original COA area.

As the FAA rules for sUAS are addressed and modified, the relief of operations restrictions should allow a much improved regulatory scenario for UAS operations in support of disaster and law enforcement activities. This will then allow a greater expansion of the role of UAS to support fire events, given that the fire agencies can succinctly identify their own airspace integration plans to avoid conflicts between UAS and manned assets.

113.4 UAS Platform Decisions: Tactical vs. Strategic

The mission requirements and needs for wildfire imaging will dictate the selection of the best UAS for a particular event. There are a number of questions that will have to be addressed prior to inclusion of UAS into operational fire imaging roles traditionally held by manned aircraft assets: Do you really need to use an unmanned platform? Where will you operate (altitude, area of operations of the UAS)? What is your sensor/payload? How will you observe FAA regulations? How much aircraft autonomy do you have/need? All of these questions must be answered to ensure successful operations and utility of information delivery.

If a sensor can fit on a sUAS, it can also fit on a small manned aircraft, and therefore the manned aircraft option should be considered when all other options are of equal importance or value. Often it is technically possible to get the capabilities you want with a sUAS, but cost-effectiveness (of platform utility) needs to be an important and highly regarded index of utility. A sUAS operation may have a large crew, which may increase costs over a manned, single-pilot platform. One must identify the technical components (data, imagery, sensor systems, etc.) that are needed to fulfill their mission requirements in the most efficient and effective means possible (Rango and Laliberte 2010; Zajkowski and Hinkley 2008a, b; Zajkowski et al. 2008a), to ensure cost-effective utility of resources. The following sections describe how these questions and factors need to be evaluated.

Where does one operate a UAS on a fire imaging mission? Wildfire locations are most often a complex mix of extreme environmental conditions (mountainous

terrain, heavy vegetation, and smoky, low-visibility situations), which make it difficult to maintain situational awareness of a fire's location, fire hazards, terrain features, and other inherent fire hazards. These situations can also affect control operations of UAS, especially if operating such out of visual range.

What is the appropriate payload? In addition to fire imaging payloads which are covered in detail in the next section, firefighters have identified another UAS mission for the fire management area, as communications relay platforms. In complex fire environments, terrain masking limits the standard radio linkages between frontline firefighting crews and the incident command teams. Employing a UAS as a “repeater-in-the-sky” can be an effective solution when the establishment of mobile, ground-based radio repeaters is often problematic. Another niche capability for UAS is to serve as meteorological observation aircraft, gathering information on the atmospheric environmental conditions while operating within and around the fire environment. This additional airborne information can supplement/enhance the ground-based meteorological data by providing a mobile measurement of conditions at critical altitude levels above a fire where manned aircraft assets may face fire-induced wind shears, unstable updrafts, and downdrafts. Therefore, the use of UAS within the wildfire operational environment should not be solely limited to image data gathering.

How will you operate the UAS? In fire applications one must take into account the fact that the tactical/sUAS crew may be close to the fire line (within the TFR or FTA), and those crew may have to be fire line qualified so that they do not hinder ground operations. This involves a prequalified level of training in emergency fire operations and protection/safety. Therefore, one cannot simply assume a UAS crew is given access to the interior perimeter of a working fire event without prior training, qualifications, and equipment. If the ground station/crew are not located at the incident and within the TFR/FTA, then one must also plan a robust communications plan and data-link system to communicate both with the aircraft, sensors, and instruments and with the incident command to ensure that the incident management team will receive the necessary data in a timely fashion.

Some of these issues are addressed in the following sections which highlight the differences in platform capabilities and sensor system needs in support of wildfire UAS observations.

113.4.1 Strategic (MALE/HALE) UAS

There are numerous categorizations of UAS, such as by size, weight, capability, altitude, and endurance ranges. For the wildfire imaging application, an identification of UAS by the altitude they operate and their endurance (Polski 2004) is offered. Medium-Altitude Long-Endurance (MALE) and High-Altitude Long-Endurance (HALE) UAS platforms include aircraft such as the MQ-9 Predator-B

(Ikhana) and the RQ-4 Global Hawk. These platforms are considered strategic assets because they would not be controlled operationally by a single fire incident team and would operate outside of the FTA (above) and can support multiple events due to their endurance capabilities. This capability comes at a price: the strategic UAS platform costs can be well into the millions of U.S. dollars and operational costs can be double to triple current manned aircraft operational costs (U.S. dollars/hour of operation). The advantages/situations where strategic UAS can support wildfire observations are:

- Long-endurance data collection flights over remote fire areas, covering a large number of incidents;
- Providing communications link (radio repeater) between incident command (IC) and ground resources;
- Providing real-time day and night sensor imagery of fire behavior to IC;
- Ability to fly above and out of way of all other fire aviation aircraft (out of TFR or FTA).

113.4.2 Tactical (LALE/LASE) UAS

Tactical UAS include Low-Altitude Long-Endurance (LALE) platforms such as the Insitu/Boeing ScanEagle and RQ-7 Shadow and Low-Altitude Short-Endurance (LASE) such as the RQ-11 Raven and RQ-16A T-Hawk and other similar platforms. The designation of “low altitude” is a relative term, where LALE systems are those that typically operate 500–3,000 ft (152–914 m) AGL, while LASE systems, often referred to as “sUAS,” generally operate under 500 ft (152 m) AGL. Larger LALE systems are capable of carrying multiple payloads and could perform missions similar to a strategic UAS over one fire. Most of these tactical/sUAS are hand-launched or catapult-launched and have belly-landing or net-captured recovery systems; therefore, they can be operated from within the wildfire TFR and do not need runway for launch/recovery. LASE or sUAS are typically small enough to be operated by a one-/two-person crew and could be used by “trained” first responders to monitor fire fronts that cannot be seen from their current location. The situations where tactical UAS can support wildfire observations are:

- Fire-line supervisors use to provide real-time photo/video and infrared imagery in tactical situations.
- Direct support of fire-line operations.
- Conditions where manned aircraft cannot operate (i.e., smoke, night, difficult terrain).
- Fire support by sUAS is required immediately and ability to launch quickly.

113.5 UAS Fire Imaging Sensor Characteristics

The use of UAS systems supporting wildfire information management is an issue of matching the appropriate platform characteristics with appropriate sensor systems.

Table 113.2 USFS-Specified Infrared Typing System (From Zajkowski et al. 2011)

| Components | Multiple incidents/large fires | | Single incident | | |
|----------------------------------|--------------------------------|--------|-----------------|----------|-----------|
| | Type 1 | Type 2 | Type 3a | Type 3b | Type 3c |
| Mount | Nadir | Nadir | Gimbaled | Gimbaled | Hand-held |
| Geo-corrected products | Yes | Yes | Yes | Optional | Optional |
| Thermal band(s) | 2+ | 1 | 1 | 1 | 1 |
| Production rate (acres per hour) | 100,000 | 10,000 | 1,000 | 500 | 100 |

First and foremost, the systems design must match the user needs/requirements; second, the UAS/sensor system must be compatible in integration capabilities; third, the system (UAS/sensor) must be able to provide access to the information/imagery collected in near real time; and finally, the UAS system must have cost-efficient operations or show marked improvements to current operations at similar or higher costs than manned aircraft operations.

The USFS has developed a guideline document that defines categories of wildfire airborne (manned) imaging support systems, known as the Infrared Typing System (Zajkowski et al. 2011). The Infrared Typing System can provide guidance on defining the strategic vs. tactical UAS sensor system capabilities. The typing system categorizes the ability of a sensor/platform to acquire either nadir or gimbaled sensor data over fires and further categorizes the ability of the sensor/platform to acquire multiple spectral bands, multiple incident/large fires (or small, single incidents), and the production/acquisition rate, measured in acres imaged per hour. The Infrared Typing System used by the USFS is shown in Table 113.2.

The Infrared Typing Systems can provide a guideline for identifying UAS fire imaging categories, where “strategic” UAS fire imaging operations would fall within the specifications of a Type 1 or 2 system and “tactical” UAS fire imaging operations would fall within the specifications of a Type 3a or 3b systems. The Type 3C category is not relevant to UAS discussions because it focuses on handheld systems operated out the door or window of a manned aircraft. The following subsections highlight examples of both strategic and tactical wildfire imaging systems that have been utilized by the U.S. fire community in UAS evaluations and operational missions.

113.5.1 Strategic Sensor Systems

Strategic systems are defined as providing broad-area coverage of a large single fire/multiple fire events during a single mission profile. The current National Interagency Fire Center (NIFC) National Infrared Operations (NIROPS) program employs two manned aircraft platforms with the wide-area (120° Field of View (FOV)) Phoenix sensor system. The Phoenix sensor on these two aircraft platforms is an example of a strategic acquisition system (<http://nirops.fs.fed.us/>). The Phoenix systems support multiple fires every mission night, providing near-real-time data

from their available MWIR/LWIR channels through a data communications link directly to NIFC where the data is accessed by GIS/interpreters reporting to individual incident teams. Given a longer platform endurance and larger payload capacity, UAS platforms in this category size class can be outfitted with specific-purpose imaging/MWIR/LWIR sensor systems of large size/weight. An example of a strategic UAS fire imaging system is the NASA Autonomous Modular Scanner (AMS) (<http://asapdata.arc.nasa.gov/ams/>), deployed on the NASA Ikhana (Predator-B derivative) UAS platform.

113.5.1.1 NASA AMS Sensor

The Autonomous Modular Sensor (AMS) is an airborne multispectral imaging line scanner capable of semiautonomous operations on both manned and unmanned aircraft that has operated on the NASA Ikhana (Predator-B) UAS. The sensor is able to provide flight endurances + 24 h, collect over multiple fire events, and easily map the entirety of large fire areas during single mission operations. Missions by the Ikhana/AMS generally cover >1,000,000 acres per hour of operation. The system is configured with 16 spectral bands ranging from the visible through near-, mid-, and longwavelength infrared (VIS-NIR-MWIR and LWIR) wavelengths (Fig. 113.4) with $\sim 0.5^{\circ}\text{C}$ temperature discrimination and measurement capabilities up to $\sim 650^{\circ}\text{K}$. The system is configured for onboard processing to derive user-specified products such as fire hot-spot identification from various channel algorithms, post-fire burn severity data, and vegetation vigor analysis. The products are delivered to user-specified locations within 10 min of collection. Product generation can be customized prior to and during flight operations. The AMS overall system weight is ~ 239.5 pounds (109 kg); therefore, the system is most appropriate on larger-class UAS such as a Predator or Global Hawk platform. A sensor pod was constructed to accommodate the AMS on the Ikhana UAS, where it was mounted under the wing.

The AMS is a line scanner imager with a Field of View (FOV) of 86° for 716 pixels across track, resulting in a ~ 3.1 -nmi (5.7 km) swath coverage width from an altitude of 10,000 ft (3,048 m) AGL. The Instantaneous Field of View (IFOV) of 2.50 mrad provides a pixel spatial resolution of ~ 25 ft (7.6 m at altitude 10,000 ft (3,048 m)). An example of AMS data collected over a large fire complex is shown in Fig. 113.5.

113.5.1.2 Wide Area Imager (WAI) Wildfire Sensor

The WAI sensor was developed under a NASA Small Business Innovative Research (SBIR) grant (2008–2012) by Xiomas Technologies, L.L.C., of Ann Arbor, Michigan (Fig. 113.6). The multispectral system is a unique design, incorporating Quantum Well Infrared Photodetector (QWIP) Focal Plane Arrays for the MWIR (4.0–5.0 μm) and LWIR (9.0–10.0 μm) channels that have an IFOV of 300 μrad which can provide a spatial resolution of 3.0 ft (1.0 m) from a platform altitude of 10,000 ft (3,048 m) AGL. Three additional spectral bands cover the visible and NIR wavelength region. The WAI is capable of fully remote/autonomous operations on UAS or adaptable to manned aircraft operations as well. The large area of coverage allows the system to be classified as a strategic asset, able to image 2.9 million

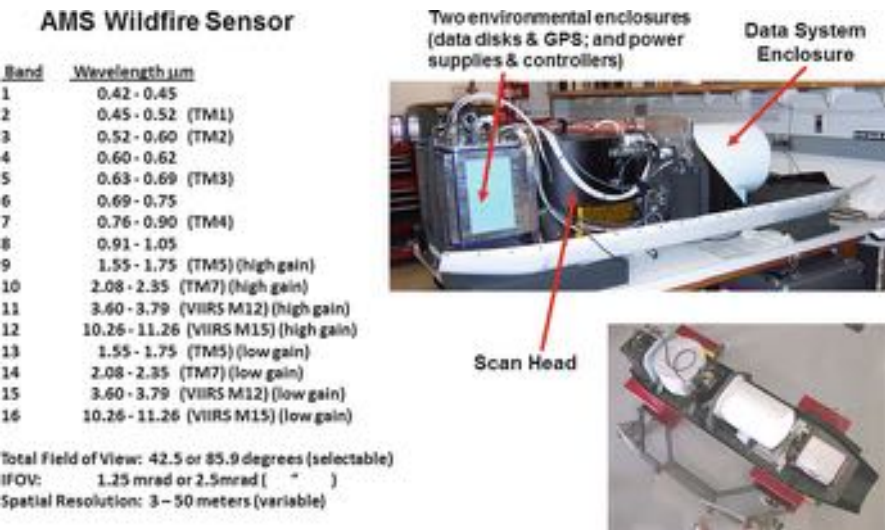


Fig. 113.4 The NASA AMS-Wildfire instrument arranged in the sensor payload pod tray. The scan head is located under the cylindrical white thermal blanket at the *top left* of the pod tray, while the pressure vessel in the center contains the digitizer and electronics. Other components include the power supply and various controlling electronics

acres per hour (1,173,588 hectares per hour) from an aircraft traveling at 300 MPH (260 KTS). The system is able to detect a 10²-in. (25.4²-cm), 600 °C hot spot from an altitude of 40,000 ft (12,192 m) AGL. The onboard processors can integrate positional information from an Applanix Inertial Measurement Unit (IMU) and provide near-real-time geo-rectified fire products through a data telemetry system link. An example of the WAI MWIR data is shown in sensor test and evaluation flight data in Fig. 113.7.

113.5.1.3 Strategic Asset Fire-Mapping Systems Costs

Most of the “strategic” multiple-channel fire imaging sensors in use/evaluation in the United States are one-of-a-kind systems, with high initial investment costs and high maintenance and calibration costs and procedures. The NASA AMS, as an example, is a one-of-a-kind multispectral imaging sensor. Costs for AMS system replication are estimated at \$1.3–\$2 M (U.S.). The market for capable, fully calibrated, wildfire scanning systems with MWIR and LWIR wavelength sensors is small (in comparison to VIS imaging systems), leading to a large development cost structure with little cost-recovery opportunity. The WAI sensor system is similarly developed from new technologies and requires a maintenance and calibration schedule to operate effectively. The cost estimates for replication of the WAI approach \$700 K–\$1.1 M (U.S.), dependent on the components, new (improved) QWIP sensors, and other additional elements.

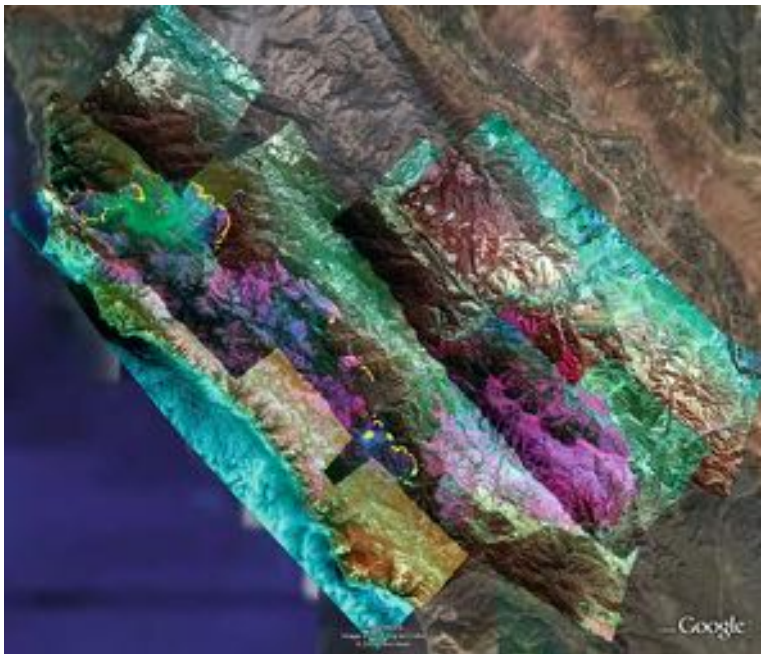
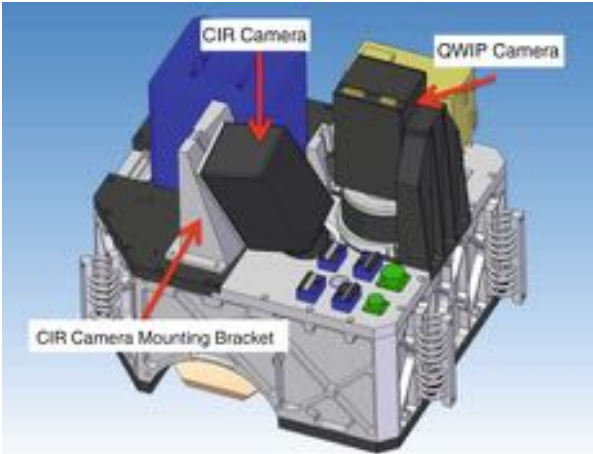


Fig. 113.5 AMS sensor data collected over the Basin/Indians Fire, Monterey County, CA, on 8 July 2008. This mosaic of flight track collections was geo-rectified and mosaicked in near real time onboard the Ikhana UAS platform and represents a composition of four to five flight segments covering the entirety of the fire, which was the third largest in California history (162,818 acres (65,890 hectares)). These flight segments were collected within a 1-h period. The *thin yellow lines* represent the real-time-derived hot spots of the fire line, while the *purple hues* represent the burned areas of the fire

Fig. 113.6 The Wide Area Imager (WAI) wildfire imaging scanner system developed by Xiomas, L.L.C. The WAI is capable of manned/UAS operations with remote/autonomous operations capabilities. The WAI is designated as a “strategic” fire sensor system able to provide data over greater than 1M acres per hour of operation (Courtesy of J. Green, XIOMAS, L.L.C.)



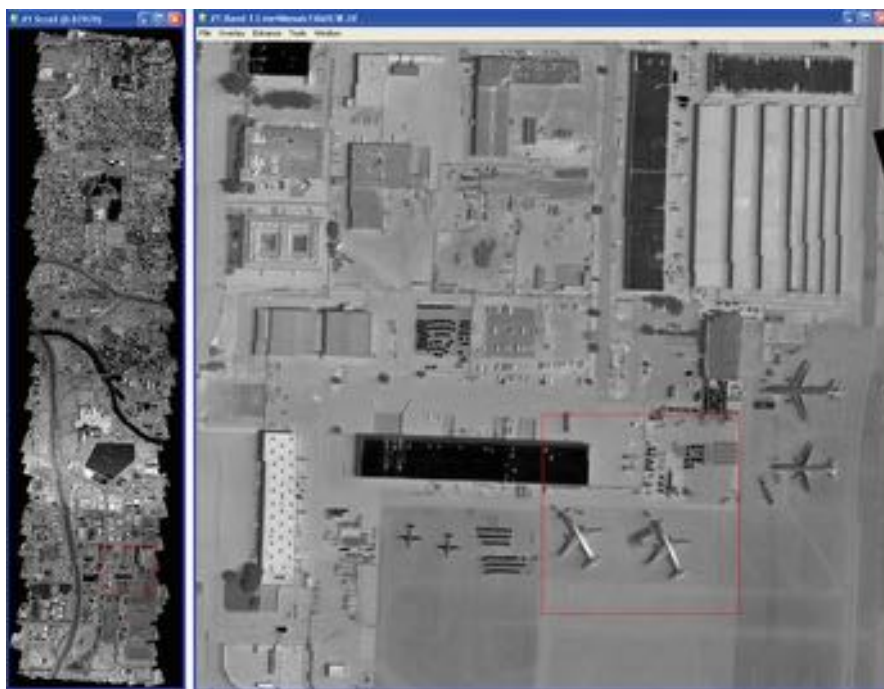


Fig. 113.7 The Wide Area Imager (WAI) MWIR data collected during test missions over Reno, Nevada, October 2011. The scene on the *left* depicts the geo-rectified data scenes for an entire flight track, automatically mosaiced together. The scene on the *right* is an enlargement covering the Reno, NV airport, showing aircraft on the tarmac and various terminal building thermal properties (Courtesy of J. Green, XIOMAS, L.L.C.)

113.5.2 Tactical Sensor Systems

Tactical systems are those providing small-area coverage of single fire events or portions of a fire to supply near-real-time data directly to fire teams on the ground. Using the Infrared Systems Type Handbook as a guide, tactical systems are those that rely on inexpensive, single band MWIR/LWIR, fixed or gimballed sensors, capable of collecting data at $\sim 1,000$ acres per hour (404 hectares per hour). The sensor systems were generally designed to support non-fire operations (night surveillance operations) and are therefore not optimized for fire, but still useful nonetheless. Some of the UAS platforms/sensors being considered by the fire community are U.S. Department of Defense (DoD) surplus sUAS systems. There are a limited number of private vendors who are developing specific-focus, sUAS/sensor platforms for fire observations, and they are currently engaged in demonstrations and flight trials with the USFS and other civilian agencies.

Fig. 113.8 An

AeroVironment Raven UAS system, operated by the department of Interior, U.S.. Geological Survey to support resource imaging projects, including wildfire imaging. The Raven is a typical small, hand-launched “tactical” UAS that will see expanded use in supporting quick assessments of fire conditions by first-on-scene responder crews, to perform small fire assessments and imaging of residual hot spots during fire mop-up operations (Image courtesy of M. Hutt, USGS)



113.5.2.1 Electro-Optical Video Systems (Gimbaled or Fixed Pointing)

Most of the tactical/sUAS in use today operate with a simple electro-optical video system. The U.S. Department of Interior, Geological Survey (USGS), operates AeroVironment Raven UAS to support various natural resource inventory and mapping efforts as well as their fire-support mapping needs on federal lands (Fig. 113.8). The Raven UAS have an electro-optical, visible-spectrum-range, real-time video sensor, with a LWIR sensor. The newer model Raven aircraft has lightweight, micro-gimbaled “Mantis[©]” VIS and LWIR sensor that provides video data over a communications link. The visible-wavelength E/O systems have limited wildfire use, where smoke precludes clear-acquisition imaging. Nonetheless, they can be an effective, inexpensive tool, useful in fire observations, especially by first responders on wildfire events, as well as utility in post-fire mop-up operations. The models with a LWIR video onboard are more effective sensor platforms for operations within a smoky fire event (Fig. 113.9).

113.5.2.2 Gimbaled LWIR Video Systems

The thermal imaging systems available on most sUAS systems rely on a single band LWIR for hot-spot discrimination. In a single band detection system, sensitivity to small unresolved fire spots is obtained by setting the threshold, or trip level, to a lower apparent temperature. The problem with this method is that warm large-area targets with the same apparent temperature will also be detected as fire hot spots (Hirsch 1971; Hirsch et al. 1971). This can prove problematic for discreet fire detection, as most sUAS thermal imaging systems are optimized for temperature discrimination closer to earth ambient temperatures (~ 300 K). These systems are generally optimized to discern warm objects, slightly above the background temperature of the imaging area, such as a person hiding in a complex brushy terrain or a warm vehicle sitting amongst a cooler camouflaged background. When these systems encounter higher-temperature objects (fire), the

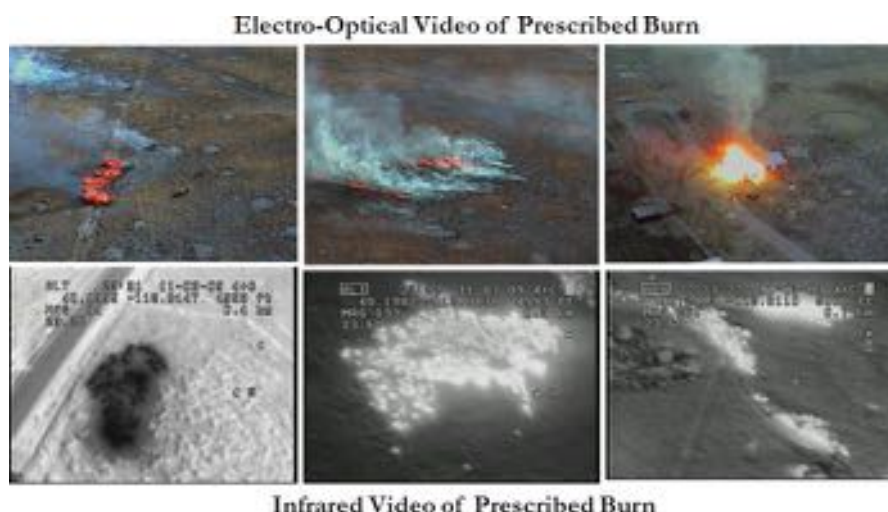


Fig. 113.9 The three images across the *top* are captures of an electro-optical (visible wavelength) video camera system flown on a U.S. Geological Survey Raven UAS. Note the potential for smoke obscuration of the fire in the visible wavelengths. With a larger plume, this can be problematic for fire hot-spot identification. The three images across the *bottom* are captures of a LWIR wavelength video system for three fire areas. The image in *bottom left* is a reverse-mapped scene (dark = warmer temperatures). These three scenes illustrate the sensitivity limitations of simple LWIR systems, where saturation occurs over the fire area and the sensor thermal discrimination capabilities are too low to capture discrete separation from burning and smoldering components. For these missions (quick-look assessment of localized events), these systems are adequate and cost-effective (Images courtesy of M. Hutt, USGS)

sensor becomes “saturated,” which complicates the identification of small hot areas and other features in the data (Fig. 113.10).

Despite their limitations, tactical/sUAS LWIR gimbaled systems are useful for fire support. They allow crews to identify new fire starts, monitor fast-moving fire-fronts, and identify smoldering fires during the mop-up phase. LWIR sensors are also able to detect subsurface heat that is not visible to the human eye such as burning root systems, which can later erupt into surface fire events. The small, LWIR micro-bolometer sensors are constantly improving with the sensor getting smaller, the sensor array sizes getting bigger (increased spatial and pixel resolution), and unit costs decreasing. Several manufactures are working on more efficient and smaller QWIP sensors that can be installed in tactical/sUAS in the near future, which will increase their detection thresholds, spatial resolution, and spatial coverage for each image area while lowering costs. SWIR systems (0.9–1.7 μm) are also being installed on sUAS. As with most LWIR sensors, SWIR systems, optimized for security applications, can be used for fire monitoring. An advantage to SWIR sensors is that the 0.9–1.7- μm wavelength region imagery is easier to interpret than LWIR imagery, especially for the basic-trained image analysts, who are familiar with more commonly available NIR/SWIR image data. One must be cautious with

Fig. 113.10 Subset image of single channel LWIR video of fire in Arizona taken from a DHS-CBP Predator-B UAS. Note the saturation/blooming of pixel response for the hot areas shown in *white*. Not all the area is on fire, but since the sensor is not optimized for high-temperature discrimination, the white areas therefore represent regions which exceed a temperature threshold, usually just 10–50 °C above background ambient temperatures (Courtesy of T. Zajkowski, RedCastle Resources)

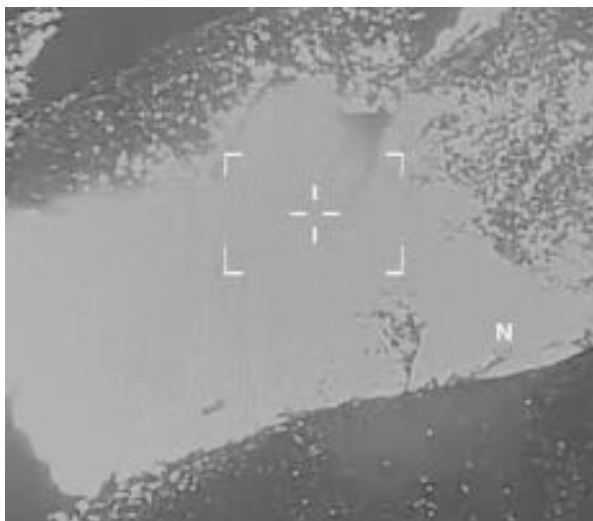


Fig. 113.11 A TASE 150, with a FLIR 320 × 120 array Proton sensor, shown mounted in a sUAS payload tray, along with the data-capture computer and electronics board. This model shows the 5-in. (12.7-cm) gimbal unit mounted in a retractable frame, allowing the unit to be retracted during takeoff and landing and in the down (imaging) position during data-acquisition flights. The system weighs 2.3 pounds (1.04 kg) (Image courtesy of D. Yoel, AAAI)



SWIR systems, as they are also more prone to smoke obscuration than MWIR and LWIR systems. An example of a sUAS gimbaled LWIR system, the TASE 150, with a FLIR 320 × 120 array Proton sensor is shown in Fig. 113.11. An image acquired from that system is shown in Fig. 113.12.

113.6 History of Airborne Fire Imaging Systems Utility in United States

The USFS began using manned aircraft for fire patrols in 1917. Early research by Johnson and Thomas (1951) and others reported the value of a Polaroid Land Camera for assessing forest damage due to fire. The camera was used in a near-real-time mode by air dropping prints to fire camps within minutes of acquisition. Since 1964,

Fig. 113.12 Imagery shows an aerial ignition (via air-dropped ignition spheres) of a prescribed fire on the George Washington Jefferson National Forest, Virginia. A curving road with a few fire vehicles on such can be seen running from *top left* to *bottom center* of the image. The linear feature of the aerial ignition fire line can be seen as distinct white lines running parallel to the road (Image courtesy of D. Yoel, AAAI)



the USFS has used airborne thermal infrared line scanner systems to map forest fires. In the 1980s, it began adopting turret-mounted thermal-infrared sensors on aircraft and, specifically, rotorcraft to also observe wildfires. These systems have proven to be very useful for supporting incident teams managing wildfires.

In the early 1960s, the USFS initiated a fire monitoring research program at the Northern Forest Fire Laboratory in Missoula, Montana, with the objective to develop a system capable of detecting and mapping fires through all atmospheric conditions (Hirsch 1963). At the conclusion of this program, Hirsch (1964) listed the attributes of an ideal remote sensing system designed for fire monitoring: including detection of fire in its early stages, effective operation day or night, and the ability to prioritize fires. Hirsch also points out in this report that the most important characteristic of a fire-mapping system is the ability to produce data related to fire-line location and size in relation to topography and road access and communicated in a timely manner.

Between the 1960s and 1970s, the USFS developed a bi-spectral imaging system based on a Daedalus scanner, sensitive in the 3–4 and 8.5–11- μm EM spectrum (Hirsch et al. 1971). System characteristics included the ability to detect small (0.09 m^2), hot (600°C) targets against backgrounds ranging from 0 to 50°C , at altitudes of $5,000\text{ m}$ above ground level. Hirsch (1968) indicated that the information these systems provided was fully integrated into the command and control

process for fire management. Hirsch also listed the limitations of these early systems as (1) non-rectified imagery, (2) slow delivery to fire management personnel, and (3) limited ability to use ancillary data sets such as digital map and digital terrain data. Some of these same limitations still exist today, causing many disaster management agencies to look to new technologies to expand existing capabilities.

In the early 1980s, Daedalus Enterprises, Inc. built the airborne Thematic Mapper Simulator (TMS) scanner for NASA. The TMS was used to simulate the LANDSAT Thematic Mapper satellite sensor system, with 12 VIS/NIR/MWIR/LWIR channels, to support earth science research. In the 1980s the TMS was modified to support fires research and management (Brass et al. 1987; Ambrosia and Brass 1988). Ambrosia (1990) summarized the unique advantages of the TMS, operating on the NASA high-altitude ER-2 aircraft, to cover large fire or multiple fires efficiently and effectively. Various data telemetry systems linked to the TMS on the NASA ER-2 high-altitude aircraft were used to demonstrate data transmission to fire management personnel in near-real-time. These ad hoc wildfire imaging support missions culminated in the 1996 NASA Wildfire Experiment, utilizing a 4-channel scanning sensor (Airborne Infrared Disaster Assessment System (AIRDAS)), demonstrating the utility of real-time data on a prescribed fire (Ambrosia et al. 1998). These experiments and demonstrations, coupled with evolving sensor technologies and improving data telemetry systems, showcased the real-time provision of MWIR and LWIR imagery over wildfire events.

Since 1998, the USFS has utilized the Phoenix imaging system in operational fire imaging. The two Phoenix systems are flown on the USFS manned aircraft (Cessna Citation jet and a Beechcraft B200 King Air). These systems have been improved with the addition of new data telemetry capabilities (Aircell system) and improved electronics.

113.6.1 History of UAS Fire Imaging Systems Developments in the United States

The first documented use of a UAS in support of wildfire intelligence gathering was in 1961 by M. Schroeder of the USFS Forest Fire Laboratory, Riverside California. The low-altitude retrievable probe (LARP) was a modified radio-controlled airplane designed to fly in and around smoke columns to gather fire weather information (Fig. 113.13). The report on the LARP noted that it was difficult to find qualified pilots to conduct the missions (Wilson and Davis 1988).

In October 1996, the USFS demonstrated the Israeli Aircraft Industries (IAI) “Firebird 2001” UAS for fire applications in Missoula, Montana, showcasing fire imaging capabilities with a TV camera and FLIR imaging system onboard (http://www.fire.uni-freiburg.de/iffn/tech/tech_11.htm) (Fig. 113.14). For various reasons the technology was not adopted at the time. Widespread interest in civilian use of UAS technology was not realized until the technological progress and operational experience gained by DoD during military conflicts from the 1990s forward. As systems supporting military intelligence gathering from UAS proliferated, the

Fig. 113.13 The low-altitude retrievable probe (LARP), a radio-controlled “drone” airplane. The LARP was designed to fly in and around the smoke column of a fire, gathering weather information (From Wilson and Davis 1988)



Fig. 113.14 Israeli Aircraft Industries Firebird 2001 UAS during flight demonstration supporting wildfire imaging in Missoula, Montana, October 1996 (Image courtesy of USFS)

broader civilian community embraced the systems for the similar roles that UAS forms could support domestically.

Since 2003, the USFS and NASA have collaborated to enhance wildfire monitoring capabilities through the Wildfire Research and Applications Partnership (WRAP) project (Hinkley and Zajkowski 2011). The WRAP has flown numerous UAS fire imaging missions that have demonstrated their ability to collect/deliver near-real-time intelligence and to serve as communications relay platforms.



Fig. 113.15 *Left to Right:* RNR Products APV-3 UAS, MLB Company Bat-3 UAS, and a modified Yamaha RMAX rotorcraft UAS, on display during fire imaging flight demonstrations at NASA-Ames Research Center, Moffett Field, California, July 2005

The intent of this effort has never been to replace manned, fire aviation assets, but rather to augment such in niche applications that are currently underserved, such as the collection of wildfire tactical intelligence. The objective for the sUAS demonstration series was to develop the Concepts of Operations (CONOPS), Standard Operating Procedures (SOPs), and contracting requirements for UAS integration to operations. This documentation will allow the USFS to phase-in UAS wildfire operation as federal aviation regulations evolve.

113.6.1.1 Small UAS Fire Imaging Demonstration: 2005/2006

The USFS and NASA collaborated on demonstrations of sUAS utility for wildfire support, with the goal to develop UAS operational guidelines, safety standards, and contracting rules so that firefighters can include sUAS in their toolkit. The first demonstrations took place at the NASA the Ames Research Center in July 2005. Three UAS systems, an MLB Company Bat-3 UAS, the RNR Products APV-3 UAS, and a modified Yamaha RMAX rotorcraft UAS, demonstrated imaging capabilities over a small fire hot spot (Fig. 113.15). The demonstration proved successful and led to expanded demonstration capabilities in 2006.

A second UAS fire-support demonstration was conducted in June 2006 in the Military Operations Area (MOA) of Fort Hunter Liggett, near King City, California



Fig. 113.16 UAS in June 2006 USFS/NASA Fire Imaging Demonstration, Ft. Hunter Liggett, CA (*Upper left*) APV-3 UAS used as an over-the-horizon communications relay platform; (*upper right*) In Situ (Boeing) A-20 ScanEagle on catapult launch assembly; (*lower left*) AeroVironment Puma, hand-launched UAS; and (*lower right*) IntelliTech Vector-P UAS, in preflight tarmac test

(Fig. 113.16). This demonstration required UAS to operate day/night over a large controlled burn, 5 mi (8 km) from the UAS airfield, and transmit that imagery back in near-real-time. The UAS included an AeroVironment Puma, an In Situ (Boeing) ScanEagle, an RnR APV-3 (communications relay platform), and an IntelliTech Vector-P. The 2006 demonstrations helped inform the UAS community of the requirements of fire management personnel to operate safely and efficiently in a wildfire scenario. The UAS commercial-provider community demonstrated their capabilities at their own costs and provided a wealth of information allowing the fire community to evaluate the use of UAS for fire imaging support and communications relay. It also allowed the agency to initialize policy objectives for integration of UAS into operational utility in support of wildfire management efforts.

113.6.1.2 NASA Altair and Ikhana (Predator-B) Western States UAS Fire Imaging Missions: 2006–2009

In 2006, NASA-ARC, the USFS, and NASA-Dryden Flight Research Center (DFRC) collaborated to demonstrate capabilities of long-endurance UAS and sensors to support near-real-time wildfire information data provision over numerous wildfires in the western United States. The missions were designed to evaluate the capabilities of the NASA Altair and the Ikhana (Predator-B) UAS. In 2006, the partners relied on the Altair UAS to support fire imaging demonstrations, while



Fig. 113.17 (a & b) (top) The NASA Altair UAS (modified Predator-B) platform in flight during 2006 Western States UAS Fire Imaging Mission Series. The AMS sensor was carried in the pod mounted below the fuselage. (bottom) The NASA Ikhana UAS (Predator-B) platform in flight during the 2007 Western States UAS Fire Imaging Mission Series. The AMS sensor was carried in the pod mounted under the wing (Images courtesy of NASA)

the Ikhana was prepared for delivery to NASA. The Altair, shown in Fig. 113.17a, is a higher-altitude version of the Predator-B. The Ikhana (Fig. 113.17b), flown in support of fire imaging missions in 2007–2009, housed the AMS sensor, to support wildfire observations with onboard, real-time, fire-derived product generation (Merlin 2009; Ambrosia et al. 2011a, b).

During the Western States UAS Fire Imaging Missions, the Ikhana flew over 20 missions in the western United States over 60 fires, demonstrating UAS/sensor capabilities (Fig. 113.18). Near-real-time, geo-rectified AMS fire imagery was delivered to wildfire incident management teams within 10–15 min of acquisition (Zajkowski et al. 2008b). This allowed an improved understanding of fire conditions by the incident teams resulting in maximization of firefighting resources. The Ikhana was also the first public-use UAS to receive a Certificate of Authorization (COA) to operate in extensive segments of the U.S. National Airspace System (NAS) without requirements for chase aircraft or ground observers. The Ikhana was operated in coordination with six FAA Air Route Traffic Control Centers (ARTCC) covering the entire western U.S.. The platform was given FAA COA support during the October 2007 Southern California firestorms that burned over 500,000 acres



Fig. 113.18 The NASA Ikhana UAS Western States UAS Fire Imaging Mission flight tracks overlain in GoogleEarth. The four missions, spread throughout the summer of 2007, ranged in duration of 9.5–20 h, and covered from 1,400 to 3,200 nmi for each mission

(770 mi², 2,000 km²), caused the evacuation of ~500,000 people, took 9 lives, and injured 85 others. During that mission period (October 24–28), the Ikhana was granted access to airspace over densely populated regions of southern California, an event never before allowed by the FAA. The Ikhana flew at a constant altitude of FL230 (7,000 m), under control of a NASA pilot through some of the world's busiest airspace, supporting image data collection over wildfires near the San Diego and Los Angeles metropolitan areas (Fig. 113.19).

This mission series demonstrated the safe operations of MALE UAS in support of wildfire/disaster event monitoring. The collaborations with the FAA allowing access to the NAS provided new opportunities for UAS to support national disaster events and have helped spur the UAS developments of other agencies. Table 113.3 lists the wildfire imaging support missions flown by the Altair and the Ikhana UAS during the Western States UAS Fire Imaging Mission's 2006–2009 fire seasons.

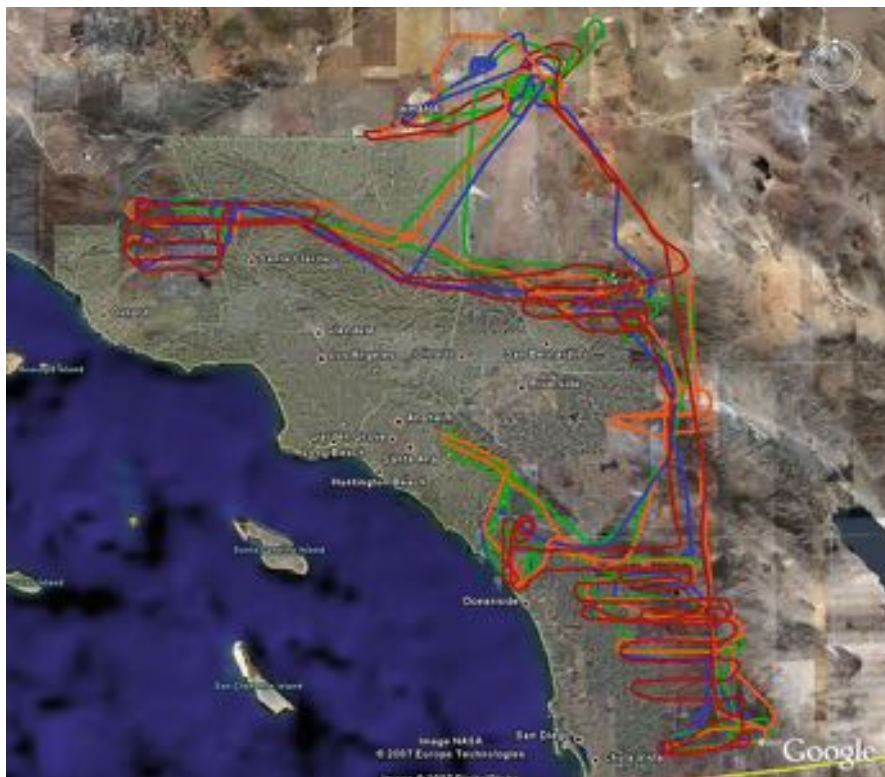


Fig. 113.19 Four days of aircraft tracks of the NASA Ikhana (Predator-B) UAS covering the October 2007 Southern California Fire Storm region. At least 11 fires were overflown every day (some twice a day), and imagery was delivered in real time from the AMS sensor on the Ikhana to incident command centers on the ground. The Ikhana UAS missions originated at NASA Dryden Flight Research Center (DFRC) at Edwards Air Force Base (EAFB), California, located in the *upper center* of the map image

113.6.1.3 DoD Predator-B UAS Fire Mission Support: 2008

In June 2008, numerous dry thunderstorms started thousands of wildfires, most occurring in northern California forestlands. At its peak, over 2,700 fires were burning throughout northern California, straining available resources. In addition to the NASA Ikhana UAS, the U.S. Air Force and U.S. Navy flew fire imaging support mission with their RQ-4 Global Hawk UAS platforms, operating from Beale Air Force Base, near Marysville, California. In addition to providing intelligence on wildfires, these mission were the groundbreaking collaboration element between the DoD and the national disaster management community, which helped forge agreements and procedures for sharing wildfire management needs/requirements between the DoD and civilian agencies.

Table 113.3 Summary of Western (United States) States Fire Mission (WSFM) flights, 2006–2009

| Year | Aircraft | Flights | Hours | Fires flown |
|------|----------|---------|-------|---|
| 2006 | Altair | 4 | 68 | Mono Lake Prescribed Fire, Esperanza Fire (CA) |
| 2007 | Ikhana | 12 | 89 | Zaca, Tar, Colby, Babcock, Jackrabbit, Butler, North, Fairmont, Grouse, Lick, Bald, Moonlight, Zaca, Trapper Ridge, Castle Rock(ID); WH Fire (MT); Columbine, Hardscrabble, Granite Creek (WY); GW Fire, Big Basin (OR); Domke Lake, South Omak (WA); SoCal Firestorm Complex (Witch Creek, Harris, Poomacha, Horno/Ammo, McCoy, Rice, Ranch, Buckweed, Santiago, Slide, Grass Valley, Magic) |
| 2008 | Ikhana | 4 | 21 | Piute, Clover, Silver, North Mountain, American River, Cub Complex, Canyon Complex, Basin, Gap, Camp, Cascadel, Hidden (CA) |
| 2009 | Ikhana | 2 | 11 | Piute, Station Fire (CA; post-burn assessments) |

Note: CA California, ID Idaho, MT Montana, WY Wyoming, OR Oregon, WA Washington

113.6.1.4 AUVSI Fire Fighting Table Top Exercises: 2009–2010

The USFS collaborated with the Association of Unmanned Vehicle Systems International (AUVSI), the Naval Postgraduate School (Monterey, CA), and U.S. Army Dugway Proving Grounds (Dugway, UT) to host two Firefighting Table Top Exercises (FF-TTX) in 2009 (CA) and 2010 (UT). These exercises provided an exercise for wildland fire personnel to assess the integration of UAS into fire management operations and allowed the unmanned systems community and firefighters to directly interact with each other in conditions similar to those found in an actual wildfire. The firefighters were able to interact with the UAS operators, giving them the opportunity to get hands-on training to direct sensors or give commands to UAS autopilots. In return, the UAS community received valuable direct feedback from the firefighters on UAS integration issues in the wildfire airspace. The TTX workshops exposed both communities to the issues and capabilities involved in supporting operational wildland fire events and the difficulties encountered on wildfires given safety and shifting situational environments (AUVSI 2010).

113.6.1.5 Fort Pickett UAS Fire Imaging Demonstration: 2010

A USFS-led UAS fire imaging demonstration project at the U.S. Army National Guard Maneuver Training Center, at Fort Pickett, Blackstone Virginia in 2010, concentrated on a UAS Long-Endurance Data Collection and Relay Mission. The UAS flight provided the incident commander multiple looks of a remote fire area to maintain situational awareness of a fire’s location, hazards, and terrain features, providing tactical/strategic information. The demonstration exemplified the capabilities of sUAS to provide improved, near-real-time fire-line safety information and fire behavior data to the incident team/first responders.

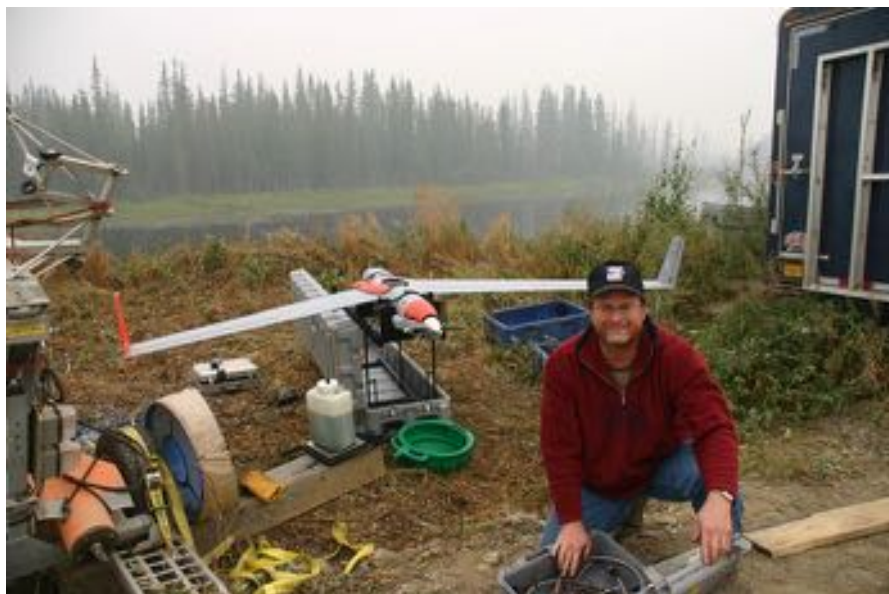


Fig. 113.20 The University of Alaska Fairbanks InSitu ScanEagle UAS being prepared for a fire observation mission over the Crazy Mountain Complex fires north of Fairbanks, Alaska, on 5 August 2009. The ScanEagle supported the fire management personnel when manned aircraft were grounded due to the flight hazard created by the large smoke pall. Note the background, hazy conditions caused by the smoke plume in the background (Image courtesy of K. Rich (UAF))

113.6.1.6 University of Alaska Fairbanks/Alaska Fire Service UAS Fire Missions, Crazy Mtn. Fire: 2009

In 2009, one of the four InSitu ScanEagle UAS owned by the University of Alaska Fairbanks supported the 447,000-acre (180,894-hectare) Crazy Mountain Complex fires burning north of Fairbanks. A large smoke pall covering the region had grounded manned aircraft assets for safety reasons, so there was no opportunity to track fire progression except for the UAF ScanEagle UAS. The ScanEagle, with a 10-foot wingspan, weighs about 40 pounds (18.14 kg), is capable of speeds between 40 and 70 MPH (64–113 KPH), and has an endurance of ~20 h and a ceiling of ~16,000 ft (~4,877 m) AGL (Fig. 113.20).

The operational use of the ScanEagle sensors allowed the incident management team to assess the fire position relative to a critical natural fuel break (stream); this assessment saved aircraft retardant drops and ground-crew firefighting efforts, by showing that the fire had not crossed the stream into heavy fire-prone areas. The UAF UAS Program is based at Poker Flat Research Range and is currently working with the Alaska Fire Service and federal land management agencies in Alaska to support more wildfire assessment needs into the future.

113.6.1.7 USGS Fire Imaging Demonstrations

Established in 2008, the USGS National Unmanned Aircraft Systems Project Office has made significant contributions in operational utility of UAS platforms supporting the U.S. Department of Interior (DOI) agencies in resource inventories, hazards assessments, and disaster imaging support on federal lands managed by DOI. Using RQ-11 Raven sUAS, the Office has flown prescribed fires over Dugway Proving Grounds in Utah (2009) and Eglin Air Force Base in Florida (2010). The USGS UAS Program Office has transitioned their UAS fire mission operations capabilities to the DOI Aviation Management Directorate, which is currently developing procedures to fly operational mission. These shared-agency developments will help improve guidelines and operational procedures with common language, objectives, goals, and directions for integration of UAS into operational utility within the federal government. It will also provide the framework for state-level agencies to build and develop their capabilities, based on these agency models.

113.7 Concluding Remarks

In the United States, the USFS and other federal agencies are testing and evaluating UAS systems on wildfire incidents, developing UAS “Roadmap” documents directing future adaptation of UAS into operational utility. The first step of that adaptation-into-operations framework is to evaluate system capabilities and appropriateness for use in various situations. The USFS has partnered in evaluation of strategic systems for national observation capabilities and is also evaluating small/tactical UAS systems. Given the expense of strategic UAS operations, the focus in strategic observations will be on continuing usage of the manned aircraft platforms/sensors and sharing usage on partner agency strategic UAS assets. Many of the capabilities demonstrated during those strategic UAS mission operations (autonomous sensor operations/processing, real-time data communication strategies) are being leveraged and adapted to manned aircraft assets, showcasing dramatic improvements in wildfire observation capabilities. When the larger, strategic assets reach a higher level of operational maturity, as well as an improved price point, they will see increased use, possibly as “shared” assets amongst a consortium of civilian agency users.

The U.S. wildfire management agencies are focusing their UAS integration to operations on tactical/sUAS platforms. Their focus is based on their mission requirements for “over-the-hill” tactical data supporting fire operations, needs for nighttime operations when manned assets are grounded/unavailable, and cost structures similar to/cheaper than their current manned aircraft (typically rotorcraft) cost structure. The USFS is also evaluating the cost trade-offs of owning/operating their own fleet of UAS vs. contracting for UAS services. Current FAA regulations favor “public-use” UAS in the NAS, but those regulations are being modified and could change dramatically within the year. When that occurs, there will likely be a surge of private UAS service providers moving towards capturing a larger market share of the federal/state agency airborne fire imaging/disaster imaging community.

Whether the wildfire management agencies decide on “in-house” or contracted UAS capabilities, that wildfire community will have to be further educated and informed of UAS capabilities and limitations, as well as the role they can play in their operational structure.

The use of UAS to support wildfire observations must be primarily dictated by the observation objectives of the fire incident team: strategic or tactical observations. These decisions then must be tempered by the capabilities of the sensors and the platforms, their ability to meet the needs of the missions, and the expected costs vs. benefit of the UAS operations. There are a number of critical factors that should be part of the needs evaluation for UAS support to wildfire events including UAS selection based upon mission needs (strategic observations, tactical, post-fire assessments of residual burning, etc.); sensor requirements (payload mass/volume, spatial coverage and spatial resolution, electrical load of sensors on UAS, communications, go/no-go criteria); and, on-station requirements (ferry distance from launch/recovery to fire area), measurement duration, weather considerations, and many more, including cost considerations.

Each of these factors should be on every checklist for determining UAS/sensor viability to support wildfire imaging. The UAS and sensor community would do well to remember that the driving objectives of the missions are to *support wildfire observations and intelligence gathering in the safest and most efficient methods possible*. If any of the criteria listed above is not “optimized” for a particular UAS use, then the mission should not be undertaken. Remembering that the role of the firefighter is to manage wildfire events with the best and most effective tools in his/her toolbox is of utmost importance. By objectively addressing the needs, requirements, capabilities, timeliness, efficiency, costs, and effectiveness of using UAS/sensor systems, the community will readily adopt UAS and associated technologies in support of wildfire “intelligence” gathering.

The way forward in UAS use for wildfire agencies is through collaboration, since no single entity currently has the assets or knowledge to support large operational UAS wildfire imaging programs. But together, the agencies can pool assets, leverage resources, and develop standard tasking, processing, exploitation, and data dissemination procedures. They can also share awareness of capabilities with each other and work cohesively to collect the data necessary to develop UAS operating plans and procedures with the FAA. Only then will UAS be readily integrated into operational wildfire imaging support environments.

References

- V.G. Ambrosia, High altitude aircraft remote sensing during the 1988 Yellowstone National Park Wildfires. *Geocarto Int.* **5**(3), 43–47 (1990)
- V.G. Ambrosia, J.A. Brass, Thermal analysis of wildfires and effects on global ecosystem cycling. *Geocarto Int.* **3**(1), 29–39 (1988)
- V.G. Ambrosia, S.W. Buechel, J.A. Brass, J.R. Peterson, R.H. Davies, R.J. Kane, S. Spain, An integration of remote sensing, GIS, and information distribution for wildfire detection and management. *Photogramm. Eng. Remote Sens.* **64**(10), 997–985 (1998)

- V.G. Ambrosia, S. Wegener, T. Zajkowski, D.V. Sullivan, S. Buechel, F. Enomoto, E. Hinkley, B. Lobitz, S. Schoenung, The Ikhana UAS western states fire imaging missions: from concept to reality (2006–2010). *Geocarto Int. J.* (Taylor & Francis Publishing) **26**(2), 85–101 (2011a)
- V.G. Ambrosia, D.V. Sullivan, S. Buechel, Integrating sensor data and geospatial tools to enhance real-time disaster management capabilities: wildfire observations, in *Societal Challenges and Geoinformatics: Geological Society of America Special Paper 482*, ed. by A.K. Sinha, D. Arctur, I. Jackson, L. Gundersen (2011b), pp. 1–12. doi:10.1130/2011.2482(01)
- Association for Unmanned Vehicle Systems International, Fire fighting table top exercise 2010 AUUSIFF-TTX 2010, Naval Post Graduate School for Center for Asymmetric Warfare, 12 Dec 2010, pp. 1–51
- J.A. Brass, V.G. Ambrosia, P.J. Riggan, J.S. Myers, J.C. Arvesen, Aircraft and satellite thermographic systems for wildfire mapping and assessment, in *Proceedings, 25th aerospace sciences meeting, AIAA*, Reno, No. 87-0887, 1987, pp. 1–7
- E.A. Hinkley, Zajkowski, USDA forest service – NASA: unmanned aerial systems demonstrations – pushing the leading edge in fire mapping. *Geocarto Int. J.* (Taylor & Francis Publishing) **26**(2), 103–111 (2011)
- S.N. Hirsch, Application of remote sensing to forest fire detection and suppression, in *Proceedings of the 2nd Symposium on Remote Sensing of Environment*, Willow run labs, Ann Arbor, MI., 1963, pp. 293–308
- S.N. Hirsch, Forest fire detection systems, in *Western Forest Fire Research Council Proceedings*, Portland, Oregon, 1964, pp. 3–5
- S.N. Hirsch, Project fire scan, summary of fire years progress in infrared fire detection, in *Proceedings of the 5th Symposium on Remote Sensing of Environment*, Willow run labs, Ann Arbor, MI., 1968 pp. 447–457
- S.N. Hirsch, Application of infrared scanners to forest fire detection. International remote sensing workshop, Environmental Research Institute of Michigan (ERIM), Ann Arbor, 1971, pp. 1–12
- S.N. Hirsch, R.F. Kruckeberg, F.H. Madden, The bispectral forest fire detection system, in *Proceedings of the 7th Symposium on Remote Sensing of Environment*, 1971, pp. 2253–2272
- Interagency Aviation Management Council, Interagency Airspace Coordination Guide, Office of Aircraft Services, Department of Interior and Aviation and Fire Management, USDA Forest Service, 29 July 2003, pp. 1–288
- C.E. Johnson, L.R. Thomas, The Polaroid Camera in fire control. *For. Serv. Fire Control Notes* **12**(2), 24–25 (1951)
- Y.J. Kaufman, C. Justice, L. Flynn, J. Kendall, E. Prins, D.E. Ward, P. Menzel, A. Setzer, Potential global fire monitoring from EOS-MODIS. *J. Geophys. Res.* **103**, 32215–32238 (1998)
- M.A. Mazurek, W.R. Cofer, J.S. Levine, Carbonaceous aerosols from prescribed burning of a boreal forest ecosystem, in *Global biomass burning: atmospheric, climatic, and biospheric indications*, ed. by J.S. Levine (MIT, Cambridge, MASS, 1991), pp. 258–263
- P. Merlin, *Ikhana Unmanned Aircraft System: Western States Fire Missions*. NASA Monographs in Aerospace History #44, SP-2009-4544 (National Aeronautics and Space Administration, Washington D.C., 2009), pp. 1–70
- P. Polski, DHS view of unmanned aerial vehicle needs, in *AIAA 3rd Unmanned Unlimited Technical Conference*, Chicago, 20–23 Sept 2004. Available online at: <http://www.authorstream.com/Presentation/Kestrel-53392-uu04polski-DHS-View-Unmanned-Aerial-Vehicle-Needs-Outline-Department-Homeland-as-Education-ppt-powerpoint/>
- A. Rango, A.S. Laliberte, Impact of flight regulations on effective use of unmanned aircraft systems for natural resource applications. *J. Appl. Remote Sens.* **4**(1), 043539 (13 July 2010), doi:10.1117/1.3474649 (2010)
- U.S. Forest Service, National Infrared Operations (NIROPS). Website: <http://nirops.fs.fed.us/>. Last accessed 6 Apr 2012
- E.E. Uthe, B.M. Morley, N.B. Nielsen, Airborne lidar measurements of smoke plume distribution, vertical transmission, and particle size. *Appl. Opt.* **21**(3), 460–463 (1982). <http://dx.doi.org/10.1364/AO.21.000460>

- C. Wilson, J.B. Davis, *Forest Fire Laboratory at Riverside and Fire Research in California: Past, Present, and Future*. General Technical Report PSW-105 (Pacific Southwest Forest and Range Experiment Station, Forest Service, U.S. Department of Agriculture, Berkeley, 1988), 22p
- W.E. Wolfe, G.J. Zissis (ed.) *The Infrared Handbook – Revised Edition* (The Infrared Information Analysis (IRIA) Center, Environmental Research Institute of Michigan, for the Office of Naval Research, Department of Navy, Washington D.C., 1989), pp. 1–30
- J. Wong, Z. Li, Retrieval of optical depth for heavy smoke aerosol plumes: uncertainties and sensitivities to the optical properties. *J. Atmos. Sci. (American Meteorological Society)* **59**, 250–261 (2002)
- T. Zajkowski, E. Hinkley, Current USFS UAS projects, in *IEEE Robotics and Automation Conference, UAVs: Civilian and Commercial Opportunities*, Pasadena, 19 May 2008
- T. Zajkowski, E. Hinkley, United States forest service unmanned aerial systems integration status, in *Proceedings of the Pecora 17 Conference*, Denver, 18–21 Nov 2008
- T. Zajkowski, E. Hinkley, K. Brewer, Using a small unmanned aircraft system (UAS) to collect forest inventory data, in *2008 Forest Inventory and Analysis (FIA) Symposium*, Park City. Proceedings: RMRS-P-56CD, 21–23 Oct 2008
- T. Zajkowski, S. Johan, J. Mendelsohn, Integration into incident management teams (IMT) and emergency operations centers (EOC) processes and use, in *Proceedings, Twelfth Biennial USDA Forest Service Remote Sensing Applications Conference, RS-2008*, Salt Lake City, 15–17 Apr 2008
- T. Zajkowski, E. Hinkley, L. Queen, T. Mellin, Infrared Field Users' Guide. Project Report, Rapid Disturbance Assessment & Service, Fire and Aviation Management, U.S. Department of Agriculture Forest Service – Engineering, Remote Sensing Applications Center, RSAC-1309-RPT3, Mar 2011, pp. 1–11

Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation

114

Konstantin Kondak, Aníbal Ollero, Ivan Maza, Kai Krieger,
Alin Albu-Schaeffer, Marc Schwarzbach, and Maximilian Laiacker

Contents

| | | |
|---------|---|------|
| 114.1 | Introduction | 2756 |
| 114.2 | Model for the Interaction | 2758 |
| 114.3 | Aerial Manipulation | 2761 |
| 114.3.1 | Existing Work | 2761 |
| 114.3.2 | General Considerations | 2762 |
| 114.3.3 | Precision Control of the Helicopter | 2764 |
| 114.3.4 | Force Interaction Between Helicopter and Environment | 2765 |
| 114.3.5 | Phase Circles | 2768 |
| 114.3.6 | Movement of the CoG | 2770 |
| 114.3.7 | Summary | 2774 |
| 114.4 | Load Transportation and Deployment | 2776 |
| 114.4.1 | Existing Work | 2776 |
| 114.4.2 | Load Transportation and Deployment by Several Helicopters | 2777 |
| 114.4.3 | Load Transportation and Deployment by a Single Helicopter | 2782 |
| 114.5 | Conclusions | 2784 |
| | References | 2784 |

K. Kondak (✉) • K. Krieger • A. Albu-Schaeffer • M. Schwarzbach • M. Laiacker
Institute of Robotics and Mechatronics, DLR (German Aerospace Center),
Oberpfaffenhofen-Wessling, Germany
e-mail: Konstantin.Kondak@dlr.de; kai.krieger@dlr.de; alin.albu-schaeffer@dlr.de;
marc.schwarzbach@dlr.de; maximilian.laiacker@dlr.de

I. Maza
Robotics, Vision and Control Group, University of Seville, Seville, Spain
e-mail: imaza@us.es

A. Ollero
Robotics, Vision and Control Group, University of Seville, Seville, Spain

Center for Advanced Aerospace Technologies (CATEC), Parque Tecnológico y Aeronáutico de
Andalucía, La Rinconada, Spain
e-mail: aollero@cartuja.us.es; aollero@catec.aero

Abstract

The applications of Unmanned Aerial Systems (UAS) are typically related to information exchanges that are useful in the context of surveillance, mapping, target tracking, and similar tasks presented in other chapters of this handbook such as forest fires, environment monitoring, measurements in volcanic environments, and maritime applications. However, other applications of interest can involve physical interactions between UAS and environmental objects as well as between several UAS. This chapter presents the state of the art in two branches of this novel research area: aerial manipulation and load transportation. In addition, the particular case of using autonomous helicopters equipped with robotic arms for aerial manipulation is detailed. Finally, recent field experiments of joint load transportation with multiple autonomous helicopters are presented.

114.1 Introduction

The applications of Unmanned Aerial Systems (UAS) are typically related to information exchanges of the UAS with objects in the environment, with ground stations, and between several UAS. These information exchanges are very useful for applications such as surveillance, mapping, target tracking, and the ones presented in other chapters. However, other UAS applications can be envisaged. These include applications that involve physical interactions between several UAS and between UAS and objects on ground. This chapter is devoted to these applications that can be considered as a natural progression of the abovementioned UAS applications related to information exchanges. The level of maturity of the functionalities required to support these applications is still low, and only prototypes have been designed. However, the potentialities are very high.

Physical interactions between aircrafts are today performed in applications such as the in-flight refueling or air-air refueling between manned aircrafts. Increasing the endurance of the missions is also an objective to be achieved in the UAS field. This could be performed in the same way by means of linking manned and UAS or UAS to UAS (Mammarella et al. 2008). In all these applications, the interactions are in the air.

Another strategy is the application of perching on rods or beams allowing UAS to increase the endurance of their missions if perches are equipped with battery charging stations. In this second case, the interactions are between the UAS and elements fixed on the ground. In Doyle et al. (2011), a bio-inspired concept and prototype that allow helicopters passive perching on cylindrical-type surfaces are presented.

Another functionality carried out by means of manned aircrafts, mainly helicopters, is the slung load transportation. This functionality is used in many applications such the transportation of line towers in remote areas or in the logging industry that use helicopters to transport logs in areas inaccessible by ground.

The slung load transportation has been also performed by means of unmanned helicopters (Bernard and Kondak 2009). The weight of the load is critical because the cost of the UAS increases exponentially with this load. In order to avoid this limitation, the joint transportation of a single load by several helicopters was proposed and demonstrated with three helicopters in the project AWARE funded by the European Commission (Bernard et al. 2011). Notice that in this case the physical coupling between UAS is involved through direct interactions of each unmanned aerial vehicle with the joint load.

Load transportation and deployment by one and several helicopters are very useful for many applications including the delivery of first-aid packages to isolated victims in disasters (floods, earthquakes, fires, industrial disasters, and many others) and are also basic technologies for other future applications, such as the building of platforms for evacuation of people in rescue operations and the installation of platforms in uneven terrains for landing of manned and unmanned VTOL aircrafts. This later application would first require the installation of the supporting units defining the horizontal surface and later the installation of the surface itself.

There are other possible applications such as the remote inspection by contact that have been targeted by the European AIRobots (2010) project. Also there are applications that involve aerial manipulation in sites that are very difficult to be accessed by ground. These include collecting samples of materials or inspection and maintenance tasks that can be solved by a UAS equipped with a special device or with a robotic arm.

Another application is the cleaning of windows or walls by applying forces while maintaining flight stability. In Albers et al. (2010), an additional actuator was added to generate forces in physical contact while the UAV stays horizontal.

The next step is to provide the aerial robots with assembly and manipulation capabilities. The assembly functionality of an aerial robot could be useful to replace parts in inaccessible sites. Another application is the construction of structures. In some of these applications, the limitation of payload is a significant drawback that could be avoided by using the cooperative assembly and manipulation. Moreover, this cooperation enables applications that require physical intervention in more than one place at the same time. The development of a Cooperative Assembly System by means of aerial robots equipped with robotic arms is the main subject of the ARCAS (2011) integrated project recently launched by the European Commission.

The on-orbit servicing and the assembly of space structures have been also the subject of research over the last two decades and include free-flying servicer concepts for handling and assembling and a few theoretical studies concerning the dynamics and control of the motion of grasped bodies in space (Moosavian and Papadopoulos 2007). However, only simplified practical demonstrations of these free-flying manipulators have been developed.

Figure 114.1 shows potential UAS applications involving physical contacts between UAS or between UAS and ground.

In many of the above applications, the use of piloted helicopters is very risky or even impossible. Moreover, the use of multiple manned helicopters is also problematic, and only simple operations, like load transportation with two

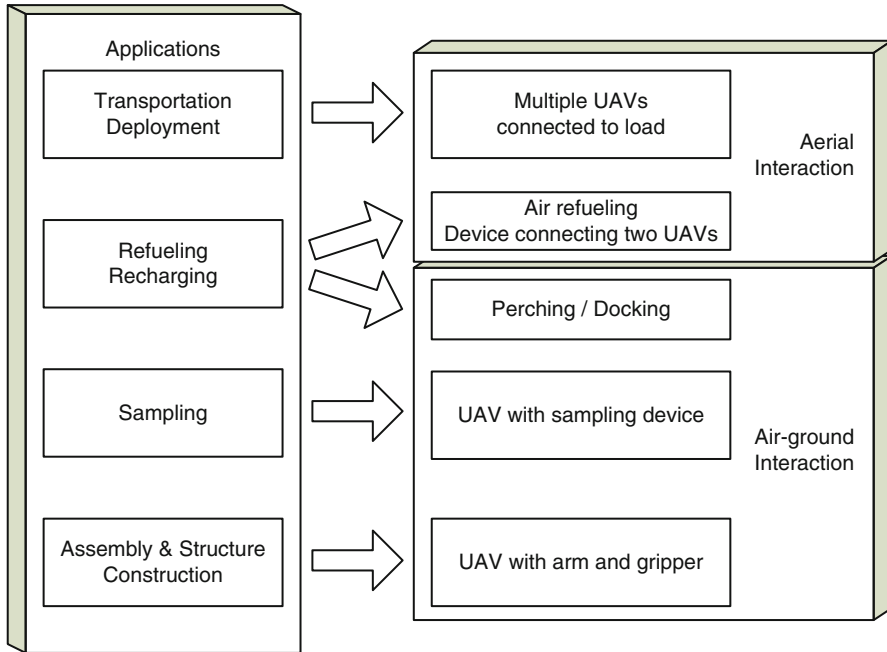


Fig. 114.1 Applications of UAS physically interacting

helicopters, can be performed by extremely skillful and experienced pilots. The level of stress is usually very high, and practical applications are therefore rarely possible.

The next section of this chapter presents a general framework for the cooperation of multiple UAS that includes possible physical interactions. The following sections are devoted to two relevant cases involving physical interactions. Thus, Sect. 114.3 presents aerial robot manipulation as an application of interest for the UAS equipped with a robotic arm and gripper in Fig. 114.1. Section 114.4 deals with load transportation and deployment, as an exemplary case of physical interaction between multiple UAS connected to a load. Finally, the last two sections are devoted to the conclusions and references.

114.2 Model for the Interaction

Robotic systems composed of a physical process and a decisional and control system can be modeled as hybrid systems (Fierro et al. 2002; Chaimowicz et al. 2004; Fagiolini et al. 2007; Li et al. 2008). Figure 114.2 shows a simplified hybrid model of a robot interacting with the environment and/or other robots.

Let us consider that the robot can perform k different tasks $\Omega = \{\tau^1, \tau^2, \dots, \tau^k\}$ with n logical conditions requiring a change of task in the current plan. Let

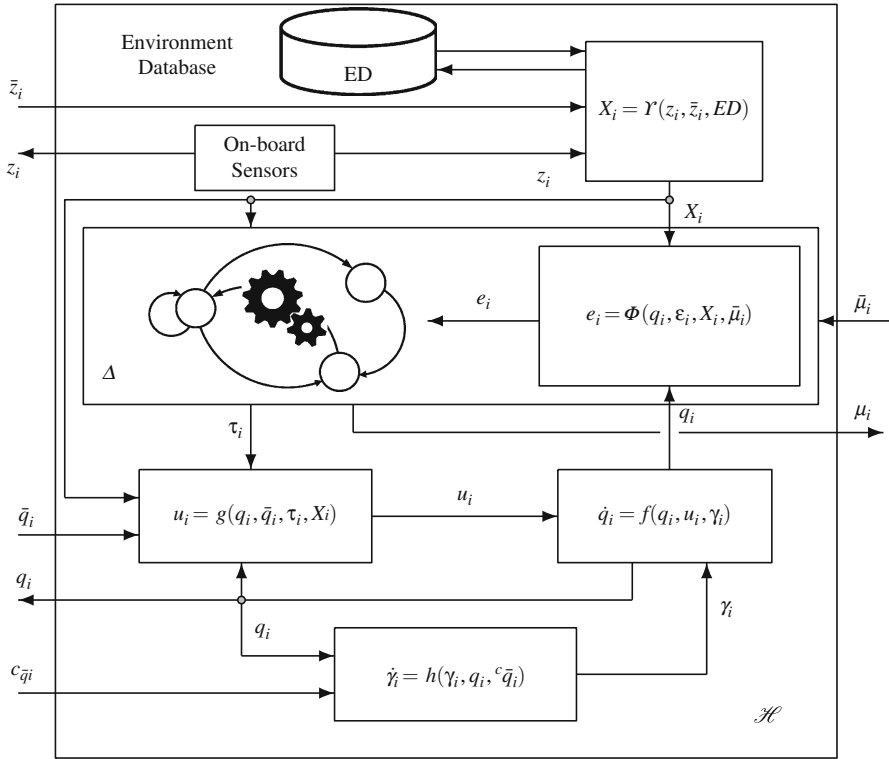


Fig. 114.2 General blocks and interactions considered in the hybrid model of each robot

$\mathcal{E} = \{e^1, e^2, \dots, e^n\}$ be a set of discrete events associated with n logical conditions requiring a change of task during the execution. The i th robot's current task has a discrete dynamics $\delta : \Omega \times \mathcal{E} \rightarrow \Omega$, i.e.,

$$\tau_i^+ = \delta(\tau_i, e_i), \quad (114.1)$$

where $e_i \in \mathcal{E}$ is an event (internal or external) requiring a change of task from τ_i to τ_i^+ , both from the set of tasks Ω . As an example, for the aerial manipulation of an object, the following sequence of tasks should be executed:

- τ^1 : approach from home
- τ^2 : precise positioning for manipulation
- τ^3 : grasping and manipulation
- τ^4 : departure
- τ^5 : return to home

It should be noticed that each task can have a different control algorithm or a different set of parameters for the same controller. The control reconfiguration

is triggered in the transition between tasks associated to different events. Event activation is generated by

$$e_i = \Phi(q_i, \varepsilon_i, X_i, \bar{\mu}_i), \quad (114.2)$$

where ε_i represents the internal events (such as changes in the execution states of the tasks) and $\bar{\mu}_i$ is a vector $\bar{\mu}_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{iN_m})$ containing the messages coming from N_m robots cooperating with the i th robot. Those messages are used for example in the negotiation processes involved in the task allocation mechanisms and are generated in each robot by a decisional module Δ (see Fig. 114.2). This module encompasses high-level reasoning and planning, synchronization among different robots, negotiation protocols for task allocation and conflict resolution purposes, task management and supervision, complex task decomposition, etc. Following the previous example, the transition from τ^1 to τ^2 will be triggered by the module Δ once the aerial robot is hovering in a suitable location for the manipulation of the object.

The information about the objects in the environment will be inferred from all the measurements z_i taken by the sensors onboard the robots and \bar{z}_i gathered by the fleet of N_s robots that can communicate with the i th robot $\{\bar{z}_j, j = 1, \dots, N_s\}$. The latter vector can be completed with the measurements from sensors located around the environment such as static cameras or sensors deployed in the area of interest. In addition, a database ED with “a priori” knowledge about the environment, including static obstacles, objects of interest and threats can be available and updated with the new information gathered. Notice that z_i also contains the forces/torques derived from the interaction with the environment that are measured with the sensors on board.

On the other hand, let $q_i \in \Theta$ be a vector describing the state of the i th robot taking values in the configuration space Θ , and let $\tau_i \in \Omega$ be the task τ that the i th robot is currently executing. This robot’s configuration q_i has a continuous dynamics

$$\dot{q}_i = f(q_i, u_i, \gamma_i), \quad (114.3)$$

where $u_i \in \mathcal{U}$ is a control input and $\gamma_i \in \Gamma$ models the dynamics associated to the possible physical interaction with the environment and/or other robots

$$\dot{\gamma}_i = h(\gamma_i, q_i, {}^c\bar{q}_i), \quad (114.4)$$

with vector ${}^c\bar{q}_i = (q_{i1}, q_{i2}, \dots, q_{iN_c})$ containing the configurations of the N_c neighbors physically connected to the i th robot. Then, $\gamma_i \neq 0$ only if there is physical interaction.

Regarding u_i , it is a feedback law generated by a low-level controller $g : \Theta \times \bar{\Theta} \times \Omega \times \Upsilon \rightarrow \mathcal{U}$, i.e.,

$$u_i = g(q_i, \bar{q}_i, \tau_i, X_i), \quad (114.5)$$

so that the robot's trajectory $q_i(t)$ corresponds to the desired current task τ_i taking into account the configurations of the N neighbors $\bar{q}_i = (q_{i_1}, q_{i_2}, \dots, q_{i_N})$ with influence in the control of the i th robot. This influence can be found, for example, in the control problem of swarms and formations. On the other hand, Eq. (114.5) also includes the vector $X_i \in \mathcal{Y}$ taking values in the environment model space \mathcal{Y} that encompasses estimations about forces/torques derived from the interaction with the environment, targets to be tracked, obstacles detected during the mission and/or known "a priori," threats to be avoided, etc.

In conclusion, the hybrid dynamics \mathcal{H} of the i th robot shown in Fig. 114.2 has \bar{z}_i , $\bar{\mu}_i$, \bar{q}_i , and ${}^c\bar{q}_i$ as inputs and z_i , μ_i , and q_i as outputs. This diagram is not intended to be exhaustive or to cover all the possible architectures and existing systems. Instead, it is aimed at providing a general overview of all the possible interactions in order to put into context the approaches presented in the next sections of the chapter.

114.3 Aerial Manipulation

In this section, a brief summary of the state of the art in the aerial manipulation area is presented. The main part of the section details the particular case of an autonomous helicopter equipped with a robotic arm for aerial manipulation. The approach presented fits well in the general scheme described in Sect. 114.2 and provides useful insights into the complexity of the associated control problem.

114.3.1 Existing Work

In the following, existing work on interactions between Unmanned Aerial Systems and objects in the environment is reviewed. The quadrotor in Albers et al. (2010) is able to carry a tool, such as a cleaning brush, and apply contact forces to a wall. An extra propeller in the quadrotor provides horizontal thrust to flight in close proximity to a wall. In addition to the quadrotor flying control system, a microcontroller is used to collect and process the data from the ultrasonic ranger to the wall and implement the control loops for the horizontal propeller. The setpoint is given by the pilot, and the basic automatic flight stabilization stays active.

In the video (Pounds and Dollar 2010), a small RC helicopter with a gripper is able to grasp an object while in flight. The gripper mechanism is mounted ventrally between the aircraft's skids; the landing gear is raised during grasping to avoid contact with the ground or target objects. The underactuated compliant gripper allows for positional errors between the helicopter and the target object. To acquire an object, the helicopter approaches the target, descends vertically to hover over the target, and then closes its gripper. Once a solid grasp is achieved, the helicopter ascends with the object. In Pounds et al. (2011), the stability of the helicopter in contact with the object and/or the ground is studied. Particularly, the longitudinal dynamics is studied, and the payload effect is modeled by means of a pitch moment. The stability analysis is carried out by considering the effect of the payload on the

mass of the helicopter, the rotational inertia of the helicopter, and the height of the rotor plane above the Center of gravity. The adjusted parameters are calculated and the stability is assured by applying the Routh-Hurwitz criterion. The paper also examines the hovering grasping as the helicopter picks up objects from the ground. The task consists of the hovering coupled to a fixed object, the transition to a free flight, and the object retrieval while hovering.

Grasping and cooperative transporting loads with quadrotors are also presented in Mellinger et al. (2010). In Mellinger et al. (2011), several lightweight, low-complexity grippers that allow quadrotors to grasp and perch on branches or beams and pick up and transport payloads are presented. The inertial parameters of the grasped object are estimated and used to adapt the controller and improve performance during flight. The next subsections also describe an autonomous electric helicopter with a hand that can be used for general manipulation tasks.

Korpela et al. (2012) presents the progress and results toward a design and physical system to emulate mobile manipulation by an unmanned aerial vehicle with dexterous arms and end effectors. An hybrid quadrotor-blimp prototype to test some manipulation concepts is presented.

Autonomous construction by means of autonomous systems has been also the subject of research work. Thus, decentralized algorithms for the cooperative assembly of 3D objects that consist of multiple types of parts, using a networked team of ground robots, have been presented (Kook Yun et al. 2009). Recently, construction experiments have been carried out with indoor quadrotors (Lindsey et al. 2011). This research did not include joint manipulation in the air, 3D positioning of the parts, and other cooperative assembly functionalities.

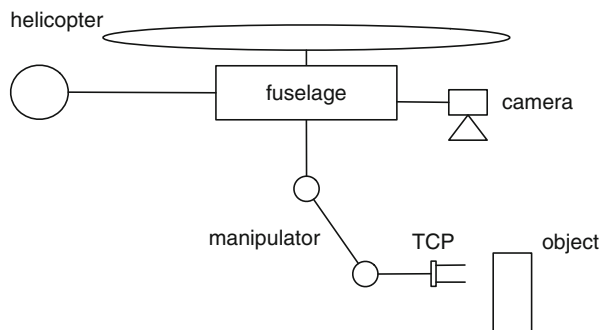
In the following, the particular case of an autonomous helicopter equipped with a robotic arm for aerial manipulation is detailed. The approach presented provides useful insights into the complexity of the associated control problem.

114.3.2 General Considerations

Let us consider systems which are similar to the one shown in Fig. 114.3. This system is composed of four main components: a helicopter, a manipulator mounted on the fuselage, sensor (e.g., a camera) to track the position and orientation of a manipulation object, and the object itself. Further, the helicopter is equipped with a position sensor (e.g., GPS) and an orientation sensor (e.g., IMU) for estimation of the fuselage position and orientation in an inertial frame. The task of this system is to fly closely enough to the object, to activate the sensor for object tracking, to go closer to the object, and, in the last phase, to perform a manipulation task. The manipulation task depends on the application and can range from taking sole probes and picking up objects to performing force interaction with an object like for assembly operations. The control of the helicopter and manipulator in the last phase can be realized using one of the following approaches or their combinations:

- Completely decoupled control
- Coupling on the kinematical level
- Coupling on the dynamical level

Fig. 114.3 Scheme of the system for aerial manipulation based on an autonomous helicopter



In case of completely decoupled control, the helicopter and the manipulator are controlled independently. Usually, the helicopter will try to maintain a hovering position trying to compensate the disturbances from the manipulator; the manipulator will try to keep the tool center point (TCP) on the object compensating the disturbances caused by the helicopter movement around the hovering position. In case of kinematical coupling, the controller for helicopter and manipulator is coupled by the positions and velocities of both subsystems. In the case of dynamical coupling, the overall controller for both subsystems considers the dynamical model of the whole system. The last case is, of course, the most general one and potentially will provide the best possible performance from the theoretical point of view. However, the practical considerations make this general conclusion not obvious. Some of the reasons for that are:

- The quality of the sensor signals, often, does not allow precise state estimation of the coupled system.
- The dynamical model of the coupled system is complicated, and the model-based controller will be usually sensitive to system parameters, non-modeled effects, and disturbances.
- For some system setups, where the masses of a helicopter are much higher than the mass of a manipulator, the dynamical coupling cannot improve the performance of the whole system significantly.

It is very likely that for different system setups, different approaches should be chosen. For example, the system composed of the helicopter with the take-off mass of 100 kg and of a 7DoF, high-dynamic, lightweight robotic manipulator with the mass of 15 kg could be controlled using the first and second approaches. On the other hand, the system composed of a helicopter with a take-off mass of 10 kg and a 5DoF degenerated manipulator could be better controlled in case of force interaction using dynamical coupling approach between the helicopter and the manipulator. Therefore, the decision about the control approach depends also on the particular system setup and application.

Regardless of the fact which control approach and which system setup will be chosen, the following aspects should be investigated in order to design a particular control scheme and to estimate the achievable control performance:

- Precision control of the helicopter
- Force interaction between helicopter and environment
- Movement of the center of gravity (CoG) due to the manipulator movement

The understanding of the above-listed aspects constitutes a base for further development of approaches and technologies for aerial manipulation. In the following subsections, each of these aspects is described in more detail. Based on presented considerations, a control approach with coupling on kinematical level is suggested. Some preliminary experimental results are also presented.

114.3.3 Precision Control of the Helicopter

The objective to reduce the take-off weight implies manipulator constructions with reduced working space or even with less than six DoF. Therefore, it is desirable to increase the helicopter positioning precision in order to facilitate the manipulation task. In same system setups, it could be even required to realize some DoF of TCP by moving the helicopter fuselage.

A suitable approach for precise placement of the helicopters relatively to the grasping object is to use a local sensor for object tracking. In Fig. 114.4, an experimental setup for investigating the performance of our helicopter control for precision positioning is shown. This setup is composed of an aerial robot with the DLR lightweight robotic arm (LWR III). This robotic arm has seven degrees of freedom for manipulation and a mass of 15 kg. The flight platform which has been additionally equipped with onboard sensor devices (IMU, GPS, etc.) is a proprietary platform and experimental system based on the Swiss UAV helicopter with a mass of 90 kg with fuel. The system has a total take-off mass of 120 kg, a rotor diameter of 3 m (3 blades), and an engine-turbine with 16 kW and 1,100 RPM. A vision-based object tracking system is integrated on the helicopter. The control scheme implemented in the system is shown in Fig. 114.5. The main controller is composed of an inner loop for orientation control, \mathbf{R}_{ori} , and an outer loop for position control, \mathbf{R}_{pos} . The output of the vision system – position of the object relative to the helicopter – is integrated at the input side of the outer loop. The performance of the whole system is given by the performance of the main controller. The combination of linear controller with nonlinear dynamical and kinematical inversion blocks is used. More detail on the main controller can be found in Kondak et al. (2007).

Using this setup, flight experiments were used to validate that the grasping of a simple object is even possible without translational movement of TCP relative to the helicopter fuselage. In the first phase of this grasping experiment, the helicopter is controlled using only the DGPS and is guided to some position not far from the object. In the second phase, the vision system is switched on and starts to track the object position. The vision system calculates the position of the object relative to the helicopter. This position is used to guide the helicopter to the specified state where the grasping is possible. In our experiments, the DGPS was used as sensor for absolute position in both phases. The output of the vision system was used only to calculate the appropriate desired position of the helicopter, as shown in Fig. 114.5.

Fig. 114.4 Experimental setup for precision control with integrated vision system. This is the first worldwide setup in outdoors involving an autonomous helicopter equipped with a robotic arm for manipulation

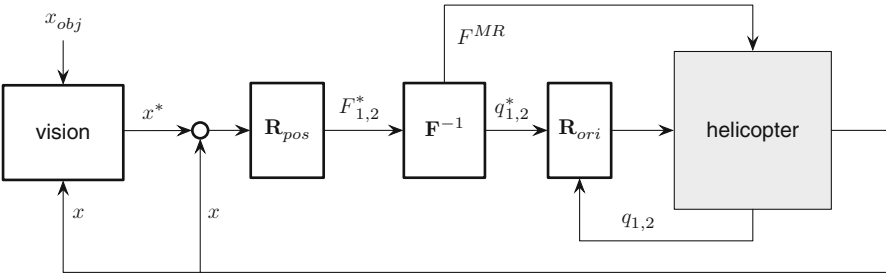


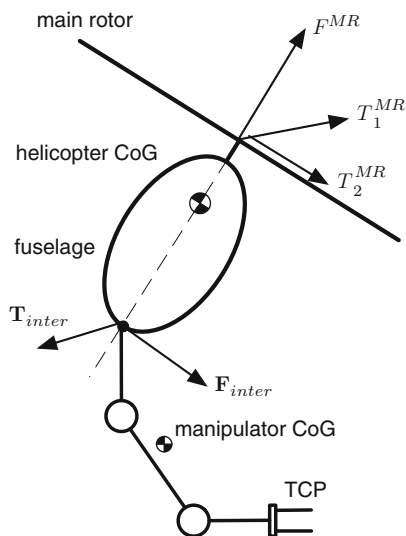
Fig. 114.5 Control scheme used for experimental setup

This was due to the higher reliability of the DGPS on our test ground compared to the vision system which had well-known problems with different light conditions and shadows in outdoor environment. With ideal weather conditions, a precision of TCP positioning up to 1 cm could be achieved.

114.3.4 Force Interaction Between Helicopter and Environment

A manipulator or other grasping device imposes forces and torques on the helicopter fuselage which influence the helicopter motion. This is illustrated on Fig. 114.6. The description and analysis of the interaction between helicopter and manipulator are complicated. The translational movement of the helicopter generates the interaction forces between manipulator and the fuselage. These forces change the orientation of the fuselage and the main rotor plane which again results in changes of the translational movement. Please note that the dynamical equations for translation and

Fig. 114.6 Force torque interaction between manipulator and helicopter fuselage



orientation of the helicopter without manipulator are coupled only in one direction. It means that the equations for the rotation are independent from translational movement, e.g., see Eqs. (114.6). The translation depends, of course, on the orientation of the main rotor plane. For the system composed of a helicopter with a manipulator, there is coupling between translational and rotational dynamics in both directions. The equations of rotational dynamic depend on the translational movement. This fact makes the analysis of the system and, therefore, the model-based controller design complicated. Regarding Fig. 114.2, $\gamma_i \neq 0$ if the manipulated object has some dynamics. The term X_i in Eq. 114.5 includes the measurements from the onboard force/torque sensors required to compute the control law.

Thus, in the scheme depicted in Fig. 114.2, there is a term $\gamma_i \neq 0$ in this case. These dynamics should be taken into account in the design of the controller. The term X_i in Eq. 114.5 includes the measurements from the onboard force/torque sensors required to compute the control law.

The same approach developed by the authors for the load transportation system based on autonomous helicopters is also proposed here. The key element in this approach is the orientation controller in the inner loop. As it is shown in Kondak et al. (2006), the rotation dynamics for a single helicopter, represented by two rigid bodies for the fuselage and the main rotor, can be expressed by the following equations:

$$\begin{aligned} T_1^{MR} + K_{12}u_2 + K_{11}\dot{u}_1 &= 0, \\ T_2^{MR} + K_{21}u_1 + K_{22}\dot{u}_2 &= 0, \end{aligned} \quad (114.6)$$

the position vector connecting sensor attaching point m and CoG of the system. The resulting orientation controller is composed of the orientation controller for a helicopter without manipulator and the compensator block **C** (see Fig. 114.7), where \mathbf{T}_r is calculated and subtracted from torques calculated in **D**. The usage of the compensator **C** has the following main advantages:

- The closed-loop system becomes very robust against variation of system parameters and disturbances.
- The orientation controller for such complicated system becomes quite simple.

There are two reasons for the robustness of the proposed orientation controller: first, the actual influence of the manipulator on the fuselage is measured through $\mathbf{F}_{\text{inter}}$, $\mathbf{T}_{\text{inter}}$, and, therefore, the compensation becomes independent from the parameters and state of the manipulator. Second, as long as the orientation of the helicopter is known, the calculated compensation torque is always in the correct phase.

Two types of influences imposed on the helicopter from a manipulator are distinguished: coherent and noncoherent influence. In case of coherent influence, the forces and torques imposed on the helicopter by the manipulator change with frequencies close to those of the helicopter movement, whereby in case of noncoherent influence, they do not change in time significantly or change with frequencies different to those of the helicopter movement.

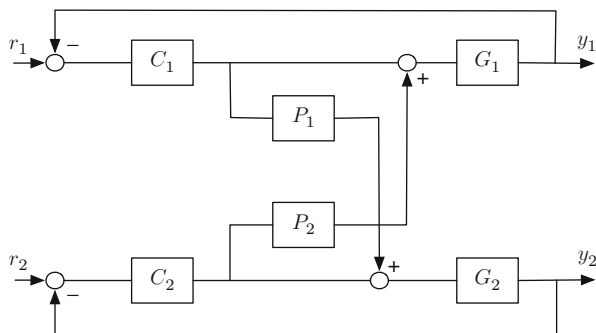
The small-size helicopters are insensitive to the noncoherent influences. This can be verified in simulation or by analysis of the dynamical equations as well as in flight experiments. Even large noncoherent torques imposed on the fuselage are compensated by the proposed controller with compensator **C**.

The opposite is the case for the coherent influence. Even small interaction forces imposed on the fuselage periodically in proper phase could yield to low-frequency instabilities and oscillations, so-called phase circles. Unfortunately, these phase circles appear often in the case when manipulator is compensating the helicopter movement in hovering, as explained in Sect. 114.3.6 in more detail. In this case, the proposed controller with compensator **C** works only with a perfectly modeled system. Therefore, for practical applications, in addition to this controller, a coupling between the helicopter and manipulator controllers should be used in order to prevent phase circles.

114.3.5 Phase Circles

Let us consider two feedback controlled systems S_1 and S_2 with plants G_1 , G_2 and controllers C_1 , C_2 , as shown in Fig. 114.8. These feedback systems can correspond to two separate physical systems or be two parts of one physical system. For example, a helicopter with position control allowing an independent motion in x - and y -directions can be considered as composed of two systems with $y_1 = x$ and $y_2 = y$. Often, especially in case where S_1 and S_2 belong to one physical system, there is a coupling between these two systems, as denoted in Fig. 114.8 with blocks P_1 and P_2 . This undesirable coupling can be caused by model parameter

Fig. 114.8 Scheme of the system with a coupling between two outputs y_1 and y_2



uncertainties, by systematic errors in the sensors, or by non-modeled physical interconnections between systems S_1 and S_2 . This coupling can lead to periodic or coherent energy flow into the systems and induce low-frequency oscillations in S_1 and S_2 . This is demonstrated for the linear case in the following. Let us consider the transfer function $G_{11} = y_1/r_1$ for the coupled systems in Fig. 114.8. It can be shown that

$$G_{11} = \frac{T}{1 + T}$$

with

$$T = G_1 C_1 - G_1 C_2 P_2 \frac{G_2 C_1 P_1}{1 + G_2 C_2}.$$

Considering symmetric systems with $G_1 = G_2 = G$, $C_1 = C_2 = C$,

$$G_{11} = \frac{GC (1 + GC (1 - P_1 P_2))}{1 + 2GC + G^2 C^2 (1 - P_1 P_2)}, \quad (114.7)$$

where only the product of both coupling blocks $P_1 P_2$ has influence on the system behavior. If one of the blocks, P_1 or P_2 , is zero, the system behavior is the same found without coupling

$$G_{11} = \frac{GC}{1 + GC}.$$

In this case, the influence from the second system can be considered as a disturbance.

On the other hand, it can be seen from the denominator in Eq. (114.7) that by increasing the gains in blocks P_1 and P_2 , the product $P_1 P_2$ can move the system poles to the right half of the s-plane, which means the existence of the oscillations in the form

$$y_1 = e^{\lambda t} \cos(\omega t),$$

$$y_2 = e^{\lambda t} \sin(\omega t).$$

If the outputs y_1, y_2 are the coordinates x, y , then the system moves in cycles with increasing radius for $\lambda > 0$ and with the period $T = \omega/2\pi$. Often, the existence of P_1 and P_2 is caused by the same physical phenomenon, and both transfer functions differ only in their phase. These are reasons for calling this type of oscillations in context of the moving platforms phase circles.

One well-known example for phase circles is the low-frequency diverging oscillations of hovering UAVs caused by the compass offset. In hovering, the compass offset induces the phase circles, and the movement of an UAV is similar to the one shown in Fig. 114.12.

The phase cycles could also appear as a result of the movement of the manipulator connected to the helicopter platform as it is described in the next paragraph in more detail.

114.3.6 Movement of the CoG

The setup for aerial manipulation shown in Fig. 114.3 is considered in the following. Let us imagine that the TCP is fixed on the object (does not move relative to the ground) and the manipulator is compensating the movement of the helicopter caused by a wind gust or the ground effect; see Fig. 114.9. The displacement of the fuselage Δx_F is compensated by the manipulator fixing the TCP at the same position $x_{TCP} = 0$. The resulting displacement of the manipulator CoG Δx_{CoG} causes additional torque on the helicopter fuselage ΔT . This additional torque changes the orientation of the helicopter, which yields to some further displacement of the fuselage Δy_F . The particular issue of a small-size helicopter with a single main rotor is that the torque applied to the fuselage yields the displacement in the same axis as a torque vector or, in our example, perpendicular to the direction of Δx_F . The fact that $\Delta y_F \perp \Delta x_F$ is important because this is the reason for existence of the phase circles as it will be shown later.

To see that $\Delta y_F \perp \Delta x_F$, it is considered the solution of the Eqs. (114.6) for constant input torques T_1^{MR} and T_2^{MR} which is given by

$$\begin{aligned} u_1(t) &= -\frac{T_2^{MR}}{2\beta} + [C_1 \cos(\tilde{\alpha}t) - C_2 \sin(\tilde{\alpha}t)], \\ u_2(t) &= \frac{T_1^{MR}}{2\beta} + [C_2 \cos(\tilde{\alpha}t) + C_1 \sin(\tilde{\alpha}t)]. \end{aligned} \quad (114.8)$$

Here, β depends on the rotor properties and its rotation speed around the rotor axis (which is assumed to be constant). Furthermore,

$$\tilde{\alpha} = 2\omega_{MR}\gamma,$$

where γ is a function of mass and geometry properties of the fuselage and the rotor and ω_{MR} is the rotation speed of the rotor. Depending on the particular system or its

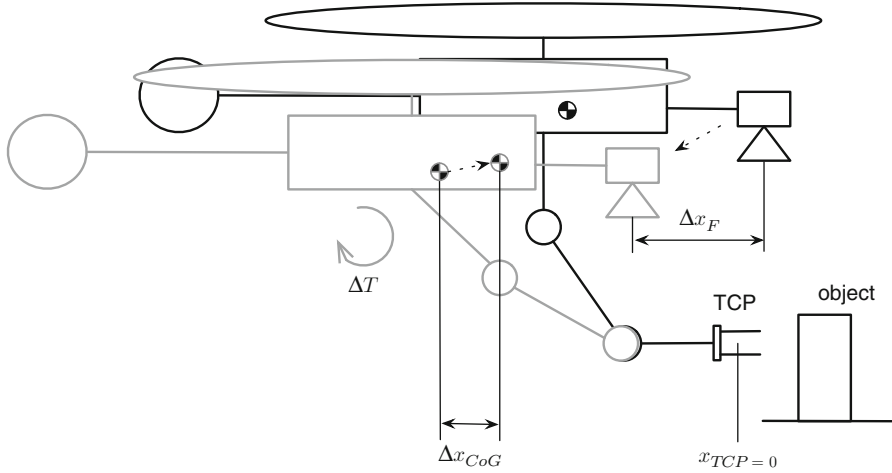


Fig. 114.9 Movement of the manipulator CoG while compensating the displacement of the helicopter

value of γ , the oscillations in Eqs. (114.8) can be neglected or not. In any case, the first part

$$\begin{aligned} u_1 &= -\frac{T_2^{MR}}{2\beta} \\ u_2 &= \frac{T_1^{MR}}{2\beta} \end{aligned} \quad (114.9)$$

shows that, due to the gyroscopic effect of the main rotor, the applied torque induces the rotation speed of the fuselage in perpendicular direction, e.g., $T_2^{MR} \rightarrow u_1$, and the corresponding change in the orientation generates additional component of the lifting force or acceleration a_2 which is perpendicular to the axis of u_1 . Therefore, the complete chain in our example is the following:

$$\Delta x_F \rightarrow \Delta T_2 \rightarrow u_1 \rightarrow q_1 \rightarrow a_2 \rightarrow \Delta y_F, \quad (114.10)$$

where coordinates x, y correspond to axes 1, 2 and q_1 is the orientation angle around axis 1.

Taking into account that $\Delta y_F \perp \Delta x_F$ and the chain (Eq. 114.10), the interaction between the helicopter and manipulator by fixing TCP relatively to some object in the environment could be described by the scheme shown in Fig. 114.10. Without the blocks P_1^* and P_2^* , this scheme represents the motion of a controlled helicopter along x - and y -axis. Here, a simplified form for rotational dynamics represented by one integrator is used. $C'_{1,2}$ are orientation controllers and $C'_{1,2}$ are translation

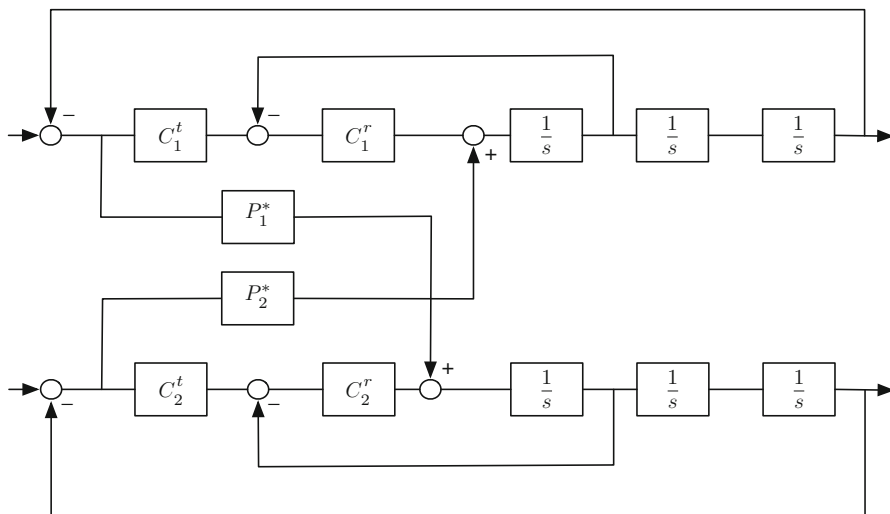


Fig. 114.10 Simplified scheme of the helicopter with manipulator using decoupled control

controllers for axes 1, 2. Using this scheme, whether the phase circles exist in this case will be investigated.

To get the transfer function G_{11} , Eq. (114.7) can be used replacing P_1 and P_2 by

$$P_1 = \frac{1}{C_2^r} \frac{1}{C_1^t} P_1^*,$$

$$P_2 = \frac{1}{C_1^r} \frac{1}{C_2^t} P_2^*.$$

The resulting expression for G_{11} in case of symmetric system with $C_1^{r,t} = C_2^{r,t}$ is then

$$G_{11} = \frac{C^r C^t s^3 + (C^r)^2 C^{tr} s^2 + (C^r C^t)^2 - P_1^* P_2^*}{s^6 + 2C^r s^5 + (C^r)^2 s^4 + 2(C^r C^t)^2 s^3 + 2(C^r)^2 C^t s^2 + (C^r C^t)^2 - P_1^* P_2^*}.$$

Using for $C^r = K_q$, for $C^t = K_d s + K_p$ with $K_q = 3$, $K_d = 2$, $K_p = 1$, the following G_{11} transfer function is obtained:

$$G_{11} = \frac{3(3 + 12s + 15s^2 + 7s^3 + 2s^4) - P_1^* P_2^*}{(3 + 6s + 3s^2 + s^3)^2 - P_1^* P_2^*}.$$

Like in (Eq. 114.7), $P_1^* P_2^*$ has influence on system poles. In Fig. 114.11, the poles movement for the gain of $P_1 P_2$, changing in interval $[-40; 40]$, is shown. As it can be seen in points $Re(P_1^* P_2^*) = -25$ and $Re(P_1^* P_2^*) = 9$, one of the system

Fig. 114.11 Movement of system poles λ while increasing gains in $P_1 P_2$

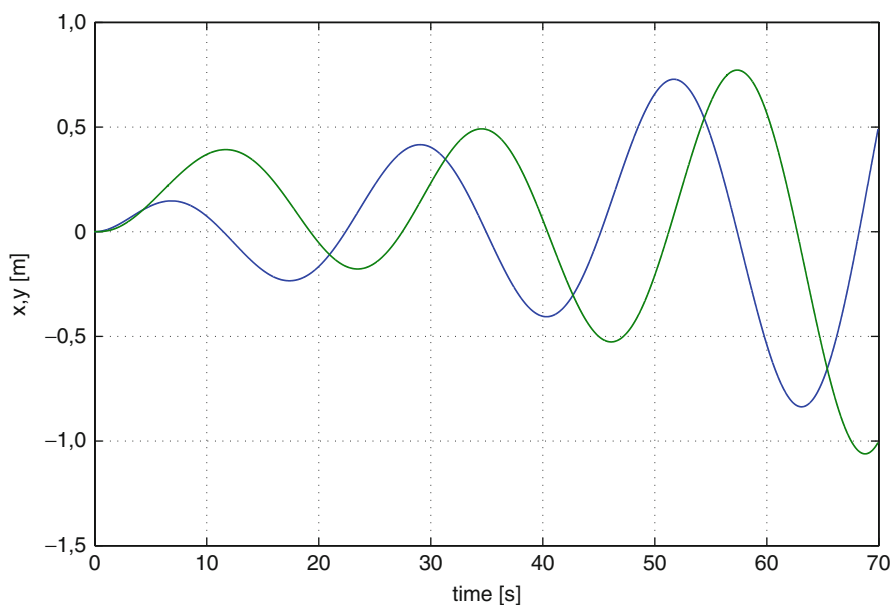
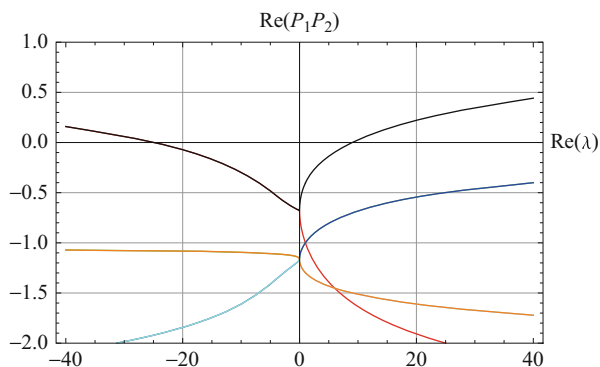


Fig. 114.12 Movement of the helicopter induced by a manipulator which moves its CoG in two dimensions (horizontal plain)

poles gets positive real part or moves to the right half of the s -plane. This means that the system starts to move on the phase circles as explained in Sect. 114.3.5. In Fig. 114.12, an example of such movement is shown. The different sign of the two points $Re(P_1^* P_2^*) = -25$ and $Re(P_1^* P_2^*) = 9$ means that phase circles can appear for P_1^* and P_2^* with the same as well as with a different sign.

One possible solution to avoid phase circles is to set P_1^* or P_2^* to zero. In our case, it can be done by restricting the motion of manipulator CoG by setting

$$\Delta y_{\text{CoG}} = 0. \quad (114.11)$$

This means that manipulator CoG moves only in xz -plane, so $P_2^* = 0$ and the phase circles cannot appear for any set of system parameters. The condition (Eq. 114.11) can be realized, e.g., using a manipulator with 7DoF.

In Fig. 114.12, the two horizontal coordinates (x , y) of a helicopter moving in a simulation experiment are shown. In this experiment, the TCP position is fixed in an inertial frame, and the manipulator compensates the movement of the helicopter. The helicopter and the manipulator are controlled independently, and condition (Eq. 114.11) is not satisfied. As it can be seen, the phase circles appear, and the helicopter controller is not able to stabilize its hovering configuration. In Fig. 114.13, the motion of the same system in simulation is shown. Different to previous case, here condition (Eq. 114.11) is satisfied, and the manipulator CoG is moving only along the x -axis. The helicopter controller stabilizes the hovering configuration, and the phase circles do not appear.

Please note that even if the displacement of the manipulator CoG and the corresponding torques acting on fuselage are small, compared to torques generated by the main rotor, they act in the phase with the fuselage movement and can induce phase circles. The disturbance torque, imposed on the fuselage, which is not changing in the phase with the fuselage movement, e.g., static displacement of the CoG, does not cause the phase circles described above and can be rejected by helicopter controller well even if its absolute value is comparable with the torques generated by the main rotor.

Condition (Eq. 114.11) and the results from the second simulation experiment shown in Fig. 114.13 are verified in flight experiments using the setup shown in Fig. 114.14. An electrical helicopter with the take-off weight of 15 kg is equipped with a pendulum, which can move a mass of 1 kg along the roll axis of the fuselage. The distance between the pendulum CoG and its rotation point is 0.12 m. The displacement of the pendulum CoG, while hovering, did not induce the oscillations as shown in Fig. 114.12.

114.3.7 Summary

Considering the aerial manipulation, three important aspects have been described: precision positioning control for helicopters, force interaction between helicopter and environment, as well as movement of the manipulator CoG. The simplified scheme was proposed to study the principal behavior of a complex system composed of a helicopter and a manipulator. It has been shown that a completely decoupled control of helicopter and manipulator is problematic if manipulator has enough DoF to move the TCP along x - and y -axis independently. The low-frequency diverging oscillations, so-called phase circles, can appear even for systems where manipulator movement is negligible in a static case. Therefore, at least a coupling on kinematical level between helicopter and manipulator controllers is needed. One possibility to couple the controllers in order to avoid phase circles is to restrict the motion of manipulator CoG in one direction. In addition to simulations, some experimental

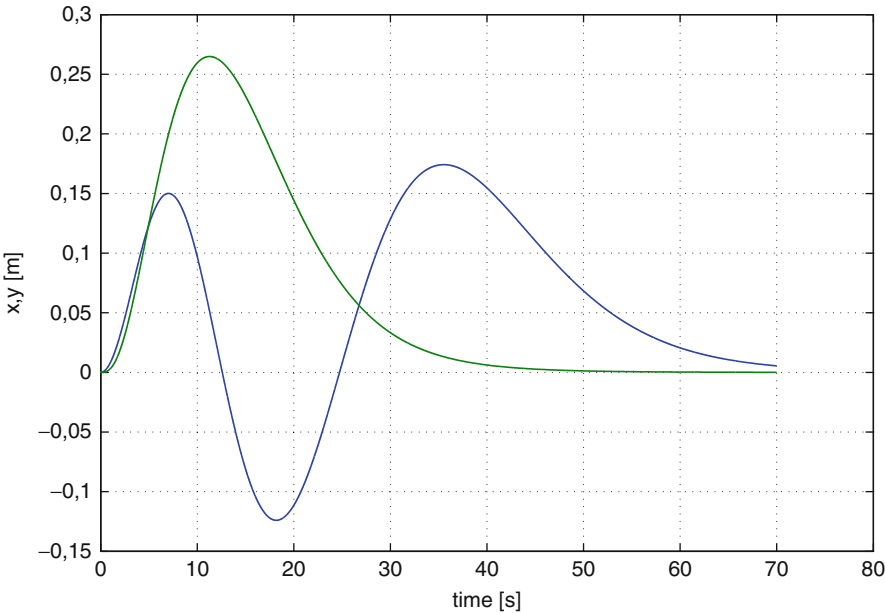


Fig. 114.13 Movement of the helicopter induced by a manipulator which moves its CoG only along one dimension



Fig. 114.14 Experimental platform with pendulum for displacement of the helicopter CoG

results were presented. An integral experiment with 7DoF manipulator mounted on helicopter is being currently prepared.

114.4 Load Transportation and Deployment

In this section, the load transportation by using autonomous helicopters is considered. The cases of single and multiple helicopters connected to the load by means of flexible ropes are considered. In both cases, helicopters experience force influence from the rope caused by dynamics of the load and its interaction with the environment or with other helicopters connected to the load. First, a brief summary of the state of the art in load transportation is presented. The main part of the section details the particular case of the load transportation by one or several autonomous helicopters. Using autonomous small-size helicopters, a load transportation system was implemented, which was successfully used for sensor node deployment in areas with a difficult access. The control approaches presented also fit well in the general scheme described in Sect. 114.2, and several field experiments using this system are presented.

114.4.1 Existing Work

The single-lift configuration, whereat a long rope couples one helicopter and one load, is the only configuration commercially utilized for the transportation of slung loads. Several textbooks (see, e.g., Wagendonk (2006)) provide information about the correct attachment of the slung loads and important safety procedures. However, the manual maneuvering of a helicopter with an attached slung load is very difficult and requires a skillful and experienced pilot. In particular, the active damping of load oscillations is a difficult task, which most pilots avoid. Instead, the pilots stabilize only the helicopter and wait for the load oscillation to die down. To support manual-piloted slung load operations, the iMAR GmbH and the German Aerospace Center DLR developed the “iSLD-IVC” (iMAR Slung Load Damping based on inertial stabilized vision control) system, which uses an artificial horizon instrument to guide the pilot. The authors are not aware of a completely autonomous system for full-size helicopters. The system presented in this work provides the full autonomous control of single-lift slung load configurations (based on small-size helicopters), including the active damping of load oscillations.

The load transportation by several autonomous systems involves physical interactions between these systems. The joint transportation of a large and heavy object by means of several ground robots has been proposed and demonstrated since the 1990s. The coordinated control of the motion of the ground robots should consider the involved forces.

A suitable approach for the coordinated control is the leader-follower scheme. In this scheme, the desired trajectory is the trajectory of the leader. The followers estimate the motion of the leader by themselves through the motion of the transported

object. This approach can be extended to multiple followers and to robots with non-holonomic constraints (see Kosuge and Sato 1999) by means of decentralized compliant motion control algorithms. The method has been implemented in an experimental system with three tracked mobile robots with a force sensor. In Sugar and Kumar (1998), the decentralized control of cooperating mobile manipulators is studied with a designated lead robot being responsible for task planning. The control of each robot is decomposed (mechanically decoupled) into the control of the gross trajectory and the control of the grasp. The excessive forces caused by robot positioning errors and odometry errors are accommodated by the compliant arms. In Borenstein (2000), the Omnimate system which uses a compliant linkage platform between two differential drive mobile robots (Labmate) is presented. In Huntsberger et al. (2004), distributed coordinated control of two rovers carrying a 2.5-m-long mockup of a photovoltaic tent is presented and demonstrated as an example of the CAMPOUT behavior-based control architecture.

Lifting and transportation of loads by using multiple helicopters has been also a research topic for many years motivated by the payload constraints of these vehicles and the high cost of helicopters with significant payload. Particularly, the lifting and transportation by two helicopters (twin lift) has been studied since the beginning of the 1990s by means of nonlinear adaptive control (Mittal et al. 1991) and H_∞ control (Reynolds and Rodriguez 1992). This research work has been done only in simulation. In Lim et al. (1999), an interactive Modeling, Simulation, Animation, and Real-Time Control (MoSART) tool to study the twin-lift helicopter system is presented. However, until recently, only simulation experiments have been presented. In December 2007, lifting and transportation of loads by means of three helicopters was demonstrated experimentally in the framework of the AWARE project. The authors presented in Bernard et al. (2011), the field experiments involving the lifting and transportation of loads by means of one (single-lift) and three (multi-lift) autonomous helicopters. Experimental results with a team of aerial robots to manipulate and transport a payload in three dimensions via cables have been recently presented (Michael et al. 2011). The authors propose two quality measures for motion plan design that minimize individual robot motion and maximize payload stability along the trajectory.

The joint transportation of a single load by means of several manned, manual piloted helicopters has been also proposed in the literature. In experiments with two manned helicopters, it was determined that the control of two coupled helicopters is a very challenging task, which emphasizes the need of automatic helicopter stabilization.

114.4.2 Load Transportation and Deployment by Several Helicopters

The coordinated control of the motion of each helicopter should consider the involved forces induced by the other helicopters and the load itself. The transported object moves with a dynamic behavior that can be expressed by means of Eq. 114.4.

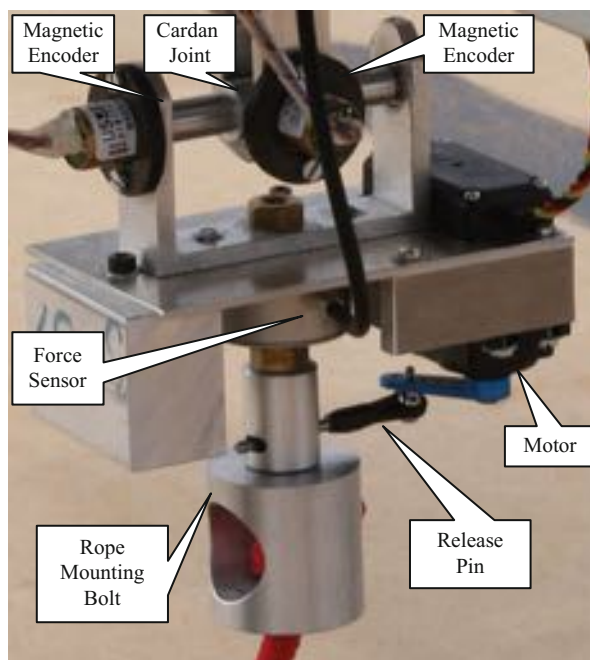
Thus, in the scheme depicted in Fig. 114.2, there is a term $\gamma_i \neq 0$ in this case. These dynamics should be taken into account in the design of the controller. The term X_i in Eq. 114.5 includes the measurements from the onboard force/torque sensors required to compute the control law. For example, a force sensor in the rope can provide a measurement of the influence of the other helicopters and the load being transported as proposed by the authors.

The orientation controller proposed in Bernard et al. (2011) features a measurement-based torque compensation and is applicable for single-, dual-, and multi-lift load transportation. Therefore, the theoretical results of this work are applicable to all the configurations presented in Cicolani and Kanning (1992), except for the twin rotor helicopter configurations, which require an adaptation of the used helicopter model. The load is connected to the helicopters using ropes, and the orientation controller follows the scheme depicted in Fig. 114.7. In this case, the inputs used in the compensator block are the measured rope force \mathbf{F}_r and the helicopter angles $q_{1,2,3}$. The block estimates the disturbance torques T_{r1}, T_{r2} that allow to decouple the orientation control for each helicopter from the rest of the system. This makes the use of the same orientation controller, independent of the number of coupled helicopters, possible. In the following, the field experiments carried out using this orientation controller are summarized.

The first successful experiment was conducted in Berlin, December 2007, in the framework of the AWARE project. For these flight experiments, three identical helicopters were equipped with a multi-UAV modular autopilot system developed at TUB. The rope is attached to the helicopter by means of the so-called load transportation device (LTD), which is mounted between the landing skids (see Fig. 114.15). A load of 4 kg was transported by means of three helicopters. In this experiment, ropes with a length of 13 m were used. The helicopters were arranged as an equilateral triangle on the ground, with a distance of 8 m between the helicopters. In Fig. 114.16, the coordinates of all three helicopters during the whole flight are shown. The x_{h123} and y_{h123} coordinates describe the horizontal motion of the helicopters, and the z_{h123} coordinates describe the vertical motion. The coordinates of the helicopters are shown in different ground-fixed frames, which have the same orientation but different origins (take-off position of each helicopter); therefore, there are no offsets between the helicopter trajectories. The mapping of helicopters and line colors is ambiguous. This is acceptable, since the motions of the helicopters are only discussed together and not individually.

The load was lifted when the helicopters reached approximate 12.4 m. The weight of the load was not considered in the controller, and therefore a small disturbance in the z_{h123} trajectories can be observed at the moment the load was lifted from the ground as well as during strong acceleration in x , y -direction. A position error of each helicopter in hovering was approx. ± 0.3 m. During the whole flight, the triangular formation of the helicopters with a precision of about ± 0.3 m and the load was moved very smoothly. To our knowledge, these were the very first successful flight experiments to transport a load using multiple autonomous helicopters.

Fig. 114.15 Detail of the so-called load transportation device (LTD) onboard each helicopter. The device is equipped with a force sensor to estimate the influence of the other helicopters and the load itself – term γ_i in Eq. 114.4



Another experiment was carried out in May 2009 within the scope of the AWARE project: A fire alarm had been declared in a building, and the objective was to place a wireless camera with pan-tilt on the top floor. The camera could provide continuous real-time video to monitor the operations of the firemen and the health status of the victims on the roof of the building. Several coupled helicopters were required for this task, since the camera, together with its associated communication equipment and batteries, was too heavy for a single helicopter. To the best of the authors knowledge, this was the first field experiment involving the transportation of a load from the ground to the roof of a building with three autonomous helicopters.

The experiment was executed as follows: The load transportation system (LTS), composed of three autonomous helicopters and the coupled payload, was ready for operation, and the helicopters were waiting on the three take-off and landing pads. The platform user specified a load transportation task to deploy the wireless pan-tilt camera on the top floor, and a plan builder module generated the full set of ordered tasks for the LTS.

It should be noted that the enormous complexity of the load transportation system composed by three helicopters was hidden to the user, who only needed to specify the deployment task, by providing the desired GPS location of the load deployment. The altitude specified for the deployment was several meters above the top floor of the building. The autonomous decision-making software of the platform had access to the map of the area in order to plan the deployment task decomposition properly,

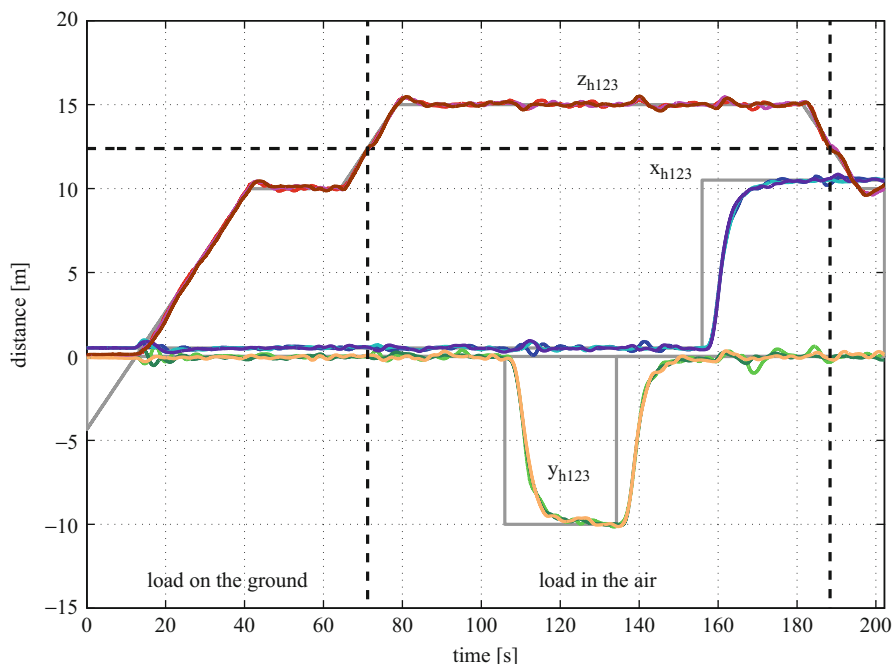


Fig. 114.16 Motion of three helicopters transporting one load of 4 kg. The load was lifted when the helicopters reached approximate 12.4 m. The coordinates of the helicopters are shown in different ground-fixed frames, which have the same orientation but different origins (take-off position of each helicopter); therefore, there are no offsets between the helicopter trajectories. The x_{h123} and y_{h123} coordinates describe the horizontal motion of the helicopters, and the z_{h123} coordinates describe the vertical motion

taking into account parameters of the system and of the environment like building altitude and the length of the ropes.

Figure 114.17 shows a 2D view of the trajectories followed by the three autonomous helicopters and the transported pan-tilt camera unit. The formation of the helicopters was similar to the formation used for the experiment conducted on December 2007: an equilateral triangle, with a distance of 8 m between the helicopters. The weight of the payload composed of camera, transportation housing, and batteries was approximately 5 kg, where the payload of each helicopter, due to additional electronic equipment, was about 2 kg. The trajectories of the helicopters are plotted in gray, and the trajectory of the camera is plotted in black. The estimated rope positions are plotted as gray lines, whereat a time step of 1 s elapsed between two successive estimations. The arrow indicates direction of the load transportation. It is noteworthy that despite the stormy weather conditions, the load stayed clearly within the extends of the roof during the deployment.

Figure 114.18 complements the flight data presented in Fig. 114.17, by depicting the values of the x , y and z coordinates of the helicopters and the load during the flight. The motion of the load is depicted from the moment the load was lifted

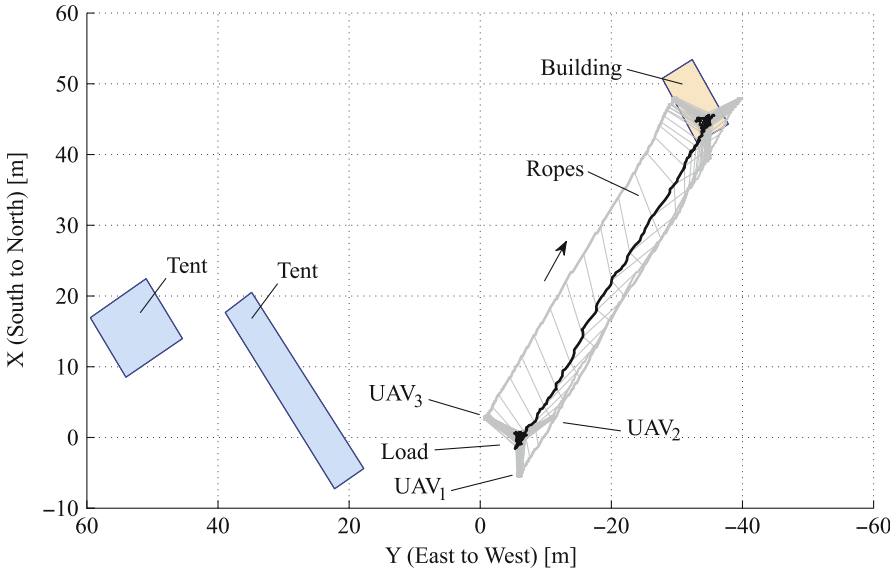


Fig. 114.17 Path followed by the three helicopters transporting the load in the $x - y$ plane. The trajectories of the load and the helicopters are in *black* and *gray*, respectively. The *arrow* indicates direction of the load transportation. The formation of the helicopters was a equilateral triangle, with a distance of 8 m between the helicopters

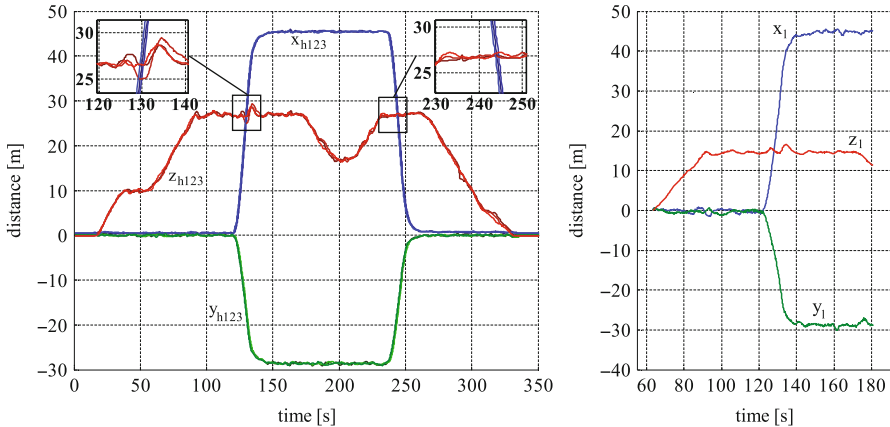


Fig. 114.18 Values of the x , y and z coordinates of the helicopters (*left*) and the load (*right*) during the flight. The coordinates of each helicopter and the load are given relative to one global Newtonian reference frame, but for each helicopter and for the load, the coordinates were plotted with respect to a relative origin (the take-off position of the respective helicopter or the lift-off position of the load). The trajectories of the helicopters show almost no deviation relative to each other

off the ground to the moment it was deployed on the roof. The coordinates of each helicopter and the load are given relative to one global Newtonian reference frame, but for each helicopter and for the load, the coordinates were plotted with respect to a relative origin (the take-off position of the respective helicopter or the lift-off position of the load). The mapping of helicopters and line colors is ambiguous, but similar to Fig. 114.16, the motions of the helicopters are only discussed together. The trajectories of the helicopters show almost no deviation relative to each other. Therefore, the helicopters preserved their relative formation during the whole flight. To demonstrate the robustness of the translation controller, the additional weight of the load was neglected during the experiment. Therefore, in the interval [120 s, 140 s], the strong acceleration of the helicopters caused a height deviation. However, beside this deviation, no additional side effects were noticeable. During the return flight [240 s, 260 s], after the load deployment, no disturbance of the height is visible. It should be mentioned that during the execution, wind gusts around 35 km/h were registered. These velocities are close to the maximum speed of the helicopters. Therefore, the performance of the coupled load transportation system is considered very good.

Finally, several photographs of the LTS during the execution of the mission are shown in Fig. 114.19. A fourth helicopter used to acquire airborne video footage of the mission can be seen in two pictures.

114.4.3 Load Transportation and Deployment by a Single Helicopter

As in the case of multiple helicopters, the rope connected to the fuselage imposes forces on it. The same control scheme for orientation controller (see Fig. 114.7) can be used in order to compensate the influence of the rope using direct force measurement. However, the position control for single helicopter transporting a load is more challenging compared to the case with three and more helicopters. The helicopter movement should actively compensate the movement of the load on a flexible rope. The task for a load transportation system is usually specified in terms of desired load movement. The required movement of the helicopter is in general more complicated than the movement of the load.

In addition to more complicated position control, the estimation of the load position could cause difficulties. In our experimental setup, the two encoders on cardan joint axes in LTD (see Fig. 114.15) were used to estimate the movement of the load. The direct calculation of the load position by multiplying angles with the rope length is problematic. Assuming normal flight conditions, the ropes are tautened between helicopter and the payload, similar to a string of a music instrument. Therefore, external disturbances (like wind gusts) or high accelerations can stimulate a natural oscillation of the ropes. This behavior has been observed in real flight experiments several times. The load motion calculated from the angle

Fig. 114.19 Three UAVs transporting a pan-tilt wireless camera to the roof of a building with a height of 12 m in Utrera (Spain) in May 2009. The images show the mission during the take-off (*top picture*) and shortly before (*bottom picture*) the load deployment. On two pictures, a fourth helicopter is visible, which was used to acquire airborne video footage of the mission



encoders during these oscillations does not correspond to the real movement of the load. The oscillating angles lead to strong responses of the controller and, in the worst case, to the further excitation of the oscillations. This can happen also in the case of load transportation with multiple helicopters. But the position control for multiple helicopters is more robust and can reject these disturbances. In case of a single helicopter, a special model-based observer should be used for motion estimation of the load. More details about this observer and positioning control for single helicopter transporting the load as well as experimental results can be found in Bernard et al. (2011).

114.5 Conclusions

The applications of Unmanned Aerial Systems (UAS) are typically related to information exchanges that are useful in the context of surveillance, mapping, target tracking, and similar tasks presented in other chapters. However, other applications of interest can involve physical interactions between several UAS and other objects. In this chapter, two branches of this novel research area have been presented: aerial manipulation and load transportation. Different types of physical interactions between UAVs as well as between an UAV and an environmental object can be represented with a general scheme as explained in Sect. 114.2. But for practical relevant cases, the models are complicated due to dynamic coupling between different systems. This was illustrated on particular case of autonomous helicopters used for aerial manipulation and load transportation. It was also shown that for analysis and control design, simplifications could be found which lead to practical relevant solutions. Preliminary results for aerial manipulation and more advanced achievements for load transportation, applied for sensor node deployment, were presented and discussed. The first results, which have been already achieved in the research community, demonstrate the big potential for UAS applications involving direct contact and force interaction.

References

- AIRobots, AIRobots European project. World Wide Web electronic publication (2010), <http://www.airobots.eu>
- A. Albers, S. Trautmann, T. Howard, T.A. Nguyen, M. Frietsch, C. Sauter, Semiautonomous flying robot for physical interaction with environment, in *Robotics, Automation and Mechatronics*, Singapore, 2010
- ARCAS, ARCAS European project (2011). World Wide Web electronic publication, <http://www.arcas-project.eu>
- M. Bernard, K. Kondak, Generic slung load transportation system using small size helicopters, in *ICRA*, Kobe (IEEE, 2009), pp. 3258–3264
- M. Bernard, K. Kondak, I. Maza, A. Ollero, Autonomous transportation and deployment with aerial robots for search and rescue missions. *J. Field Robot.* **28**(6), 914–931 (2011)
- J. Borenstein, The OmniMate: a guidewire- and beacon-free AGV for highly reconfigurable applications. *Int. J. Prod. Res.* **38**(9), 1993–2010 (2000)
- L. Chaimowicz, V. Kumar, M.F.M. Campos, A paradigm for dynamic coordination of multiple robots. *Auton. Robots* **17**(1), 7–21 (2004)
- L.S. Cicolani, G. Kanning, Equations of motion of slung-load systems, including multilift systems. Technical report, NASA Ames Research Center, 1992
- C.E. Doyle, J.J. Bird, T.A. Isom, C.J. Johnson, J.C. Kallman, J.A. Simpson, R.J. King, J.J. Abbott, M.A. Minor, Avian-inspired passive perching mechanism for robotic rotorcraft, in *IROS*, San Francisco (IEEE, 2011), pp. 4975–4980
- A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, A. Bicchi, Decentralized intrusion detection for secure cooperative multi-agent systems, in *Proceedings of the IEEE International Conference on Decision and Control*, New Orleans, 2007, pp. 1553–1558
- R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J.P. Ostrowski, G. Pappas, C.J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, B. Southall, A framework and architecture for multi-robot coordination. *Int. J. Robot. Res.* **21**(10–11), 977–995 (2002)

- T.L. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, P.S. Schenker, P. Pirjanian, H.D. Nayar, Distributed control of multi-robot systems engaged in tightly coupled tasks. *Auton. Robots* **17**(1), 79–92 (2004)
- K. Kondak, M. Bernard, N. Losse, G. Hommel, Elaborated modeling and control for autonomous small size helicopters, in *International ISR/Robotik Joint Conference*, Munich (VDI, 2006), pp. 207–208
- K. Kondak, M. Bernard, N. Meyer, G. Hommel, Autonomously flying VTOL-robots: modeling and control, in *ICRA*, Roma (IEEE, 2007), pp. 736–741
- S. Kook Yun, M. Schwager, D. Rus, Coordinating construction of truss structures using distributed equal-mass partitioning, in *IROS*, St. Louis (IEEE, 2009)
- C.M. Korpela, T.W. Danko, P.Y. Oh, MM-UAV: mobile manipulating unmanned aerial vehicle. *J. Intell. Robot. Syst.* **65**(1–4), 93–101 (2012)
- K. Kosuge, M. Sato, Transportation of a single object by multiple decentralized-controlled non-holonomic mobile robots, in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Kyongju, 1999, vol. 3, pp. 1681–1686
- H. Li, F. Karray, O. Basir, I. Song, A framework for coordinated control of multiagent systems and its applications. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **38**(3), 534–548 (2008)
- C. Lim, R.P. Metzger, A. Rodriguez, Interactive modeling, simulation, animation and real-time control (MoSART) twin lift helicopter system environment, in *Proceedings of the American Control Conference*, San Diego, 1999, vol. 4, pp. 2747–2751
- Q. Lindsey, D. Mellinger, V. Kumar, Construction of cubic structures with quadrotor teams, in *Proceedings of Robotics: Science and Systems*, Los Angeles, 2011
- M. Mammarella, G. Campa, M.R. Napolitano, M. Fravolini, R. Perhinschi, Machine vision/GPS integration using EKF for the UAV aerial refueling problem. *IEEE Trans. Syst. Man Cybern. Part C* **38**(6), 791–801 (2008)
- D. Mellinger, M. Shomin, N. Michael, V. Kumar, Cooperative grasping and transport using multiple quadrotors, in *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, Lausanne, 2010
- D. Mellinger, Q. Lindsey, M. Shomin, V. Kumar, Design, modeling, estimation and control for aerial grasping and manipulation, in *IROS*, San Francisco (IEEE, 2011), pp. 2668–2673
- N. Michael, J. Fink, V. Kumar, Cooperative manipulation and transportation with aerial robots. *Auton. Robots* **30**(1), 73–86 (2011)
- M. Mittal, J. Prasad, P. Sra, Nonlinear adaptive control of a twin lift helicopter system. *IEEE Control Syst.* **11**, 39–45 (1991)
- S.A.A. Moosavian, E. Papadopoulos, Free-flying robots in space: an overview of dynamics modeling, planning and control. *Robotica* **25**(5), 537–547 (2007)
- P. Pounds, A. Dollar, Hovering stability of helicopters with elastic constraints, in *Proceedings of the ASME Dynamic Systems and Control Conference*, Cambridge, 2010
- P.E.I. Pounds, D.R. Bersak, A.M. Dollar, Grasping from the air: hovering capture and load stability, in *ICRA*, Shanghai (IEEE, 2011), pp. 2491–2498
- H.K. Reynolds, A.A. Rodriguez, H-inf control of a twin lift helicopter system, in *Proceedings of the 31th Conference on Decision and Control*, Tucson, 1992
- T. Sugar, V. Kumar, Decentralized control of cooperating mobile manipulators, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, 1998, vol. 4, pp. 2916–2921
- W. Wagtendonk, *Principles of Helicopter Flight* (Aviation Supplies & Academics, Newcastle, 2006)

J. B. de Sousa, Philip McGuillivary, João Vicente,
Maria Nunes Bento, José A. P. Morgado, Maria Madruga Matos,
Ricardo Ayres Gomes Bencatel, and Paulo Mónica de Oliveira

Contents

| | | |
|---------|-------------------------|------|
| 115.1 | Introduction | 2788 |
| 115.2 | UAS Background | 2789 |
| 115.3 | Maritime Missions | 2795 |
| 115.3.1 | Taxonomy | 2795 |
| 115.3.2 | Current Practice | 2798 |

J.B. de Sousa (✉)

Department of Electrical and Computer Engineering, University of Porto, Porto, Portugal

e-mail: jtasso@fe.up.pt

P. McGuillivary

United States Coast Guard, PACAREA Science Liaison, Alameda, CA, USA

e-mail: Philip.A.McGillivary@USCG.MIL

J. Vicente

Portuguese Joint Command and Staff College/Security and Defense Research Center, Lisboa,
Portugal

e-mail: joao.vicente.6@gmail.com

M.N. Bento • J.A.P. Morgado • M.M. Matos

Centro de Investigação, Academia da Força Aérea, Sintra, Portugal

e-mail: mfbento@academiafa.edu.pt; jamorgado@academiafa.edu.pt; mlmadruga@academiafa.edu.pt

R.A.G. Bencatel

Aerospace Engineering Department, University of Michigan, MI, USA

e-mail: bencatel@umich.edu

P.M. de Oliveira

Instituto Português do Mar e da Atmosfera, Lisboa, Portugal

e-mail: paulo.monica@ipma.pt

115.4 Current Trends and How These Will Affect UAS for Maritime Operations 2801

115.4.1 Ocean Observation and Global Sustainability 2801

115.4.2 Technological Trends 2802

115.4.3 UAS Capabilities and Life Cycle Management 2802

115.5 Potential Future UAS Maritime Operations 2808

115.6 Conclusions 2809

References 2809

Abstract

Current practices and major trends in unmanned aircraft systems (UAS) for maritime operations are presented along with projections of future UAS maritime applications. First, various aspects of UAS program management, including organization, operator responsibilities, program operation, and overall program costs, are reviewed. Next, maritime aircraft missions and required capabilities are outlined, and current practices for both manned and unmanned aircraft are described. Technological trends relating to UAS are also discussed, focusing on developments relevant to maritime UAS operations. This information provides a background for assessing potential future maritime UAS operations. Trends including miniaturization of sensors and computer systems, high energy density of power sources, and increased subsystem standardization and modularity will have important effects in the future. However, another significant trend is towards increased system autonomy via new command and control frameworks that facilitate integration of UAS into higher level maritime observing systems through new concepts of operation for networked systems and new business models.

115.1 Introduction

Unmanned aircraft systems (UAS) are here to stay. The focus has been more on military UAS. This has to do with clear operational needs and established certification protocols. However, recent trends in UAS technology developments and certification procedures may pave the way for an expanding range of UAS uses for nonmilitary purposes.

Investment in UAS has been steadily increasing. The U.S. Department of Defense (USDOD) alone increased its expenditures on UAS from \$284 million in 2000 to a staggering U.S.\$3.3 bn in 2010 (Gertler 2012). The annual global world expenditures on these systems are expected to rise to double digits (U.S.\$ bn) by 2020. This rise has been enabled by technological developments in materials, computers, sensors, and command and control. Currently, the trend is towards onboard autonomy. Autonomy paves the way for operations in remote locations where data links to remote operators may not be available or when the cost of communications is prohibitive.

Onboard autonomy basically means that deliberation takes place onboard without human intervention. In other words, the typical sense-deliberate-act cycle is intrinsic

in contrast to what happens with automated vehicles, where sense and acting are mediated by scripted control procedures:

Autonomous means that a system has a choice to make free of outside influence... Automatic means that a system will do exactly as programmed, it has no choice (Clough 2002).

Maritime deployments present added challenges for UAS design and operations because of marine environmental conditions and more complicated communication requirements. Additionally, the lack of ground support and large areas of ocean imply either land-based access corridors with greater UAS endurance and range requirements or UAS ship or water launch and recovery mechanisms. The realities of routine maritime UAS operation can also mean additional certification issues for integration into ship systems in addition to UAS coordination with other maritime assets, including manned aircraft.

The paper discusses current practices and trends in UAS for maritime operations. This is not a roadmap and does not address issues specific to countries or organizations. It is a focused overview of maritime UAS systems complementing and summarizing information available from multiple sources including roadmaps, standards, regulations, and essays. By this means some concept of future UAS maritime operations can be better understood.

The conceptual framework for the discussion is presented in the next two sections. First, UAS background is reviewed. The conceptual framework stems from a notational work breakdown structure (WBS), which is a standard method for understanding systems. Operations and support, levels of automation, operator functions, interoperability, certification, cost structure, missions, and required capabilities are then discussed in this framework. Life cycle management concerns are also emphasized because UAS are rapidly evolving technologies, and obsolescence is a major concern. The discussion uses information available from disparate sources and provides a framework targeted at maritime operations. Second, maritime aircraft missions and required capabilities and current practices for both manned and unmanned aircraft (UA) are discussed in this framework.

Trends in maritime operations and in technological developments pertaining to UAS cost and capabilities are briefly discussed before enumerating how these may affect future maritime aircraft operations. This will entail a paradigm shift: from single vehicle to multi-vehicle networked operations.

115.2 UAS Background

An UAS is a pilotless aircraft. Piloting is done by onboard computers and remote pilots. This makes dull, dirty, and dangerous (3D) missions one of the main reasons for operational use of UAS, notably in military operations.

UAS come in several shapes, sizes, configurations, and categories. Extreme examples include the *QinetiQ Zephyr Solar Electric* UAS (QinetiQ 2009) and the *Micromechanical Flying Insect* (Fearing 2007).

Table 115.1 UAS categories (Adapted from DOD 2009)

| UAS category | Maximum gross takeoff weight (lbs) | Normal operating altitude (ft) | Speed (kts) | Representative UAS |
|--------------|------------------------------------|---------------------------------|--------------|--------------------------------|
| Group 1 | 0–20 | <1,200 Above ground level (AGL) | 100 | <i>WASP III, Pointe</i> |
| Group 2 | 21–55 | <3,500 AGL | <250 | <i>Silver Fox, ScanEagle</i> |
| Group 3 | <1,320 | <18,000 Mean sea level (MSL) | | <i>RQ-7B</i> |
| Group 4 | >1,320 | | Any airspeed | <i>MQ-5B, MQ-8B</i> |
| Group 5 | | >18,000 MSL | | <i>MQ-9A, RQ-4 Global Hawk</i> |

In spite of UAS diversity, it is important to have commonly accepted and understood UAS categories. The categorization methodology introduced by the USDOD (DOD 2009) is based on three enduring UA attributes: weight, altitude, and speed. The five resulting categories are reproduced in Table 115.1.

A different categorization has to do with roles of aircraft. The U.S. military has a *Tier* system to designate roles of individual aircraft in a usage plan for integrated operations. Different branches have their own tier systems. The U.S. Air Force tiers are the following:

Tier N/A: Small/micro unmanned aircraft vehicle (UAV)

Tier I: Low altitude, long endurance

Tier II: Medium altitude, long endurance (MALE)

Tier II+: High altitude, long endurance conventional UAV (or HALE UAV)

Tier III-: High altitude, long endurance low-observable UAV

A few definitions, taken from Hagan 2009, are needed at this stage:

Capability. The ability to achieve a desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks.

Mission. The objective or task, together with the purpose, which clearly indicates the action to be taken.

Process. The combination of people, equipment, materials, methods, and environment that produces output – a given product or service.

System. The organization of hardware, software, material, facilities, personnel, data, and services needed to perform a designated function with specified results, such as the gathering of specified data, its processing, and delivery to users.

Subsystem. A functional grouping of components that combine to perform a major function within an element such as electrical power, attitude control, and propulsion.

Work Breakdown Structure. An organized method to break down a project into logical subdivisions or subprojects at lower and lower levels of details.

Technological trends, capabilities, costs, operations, and missions are better understood in terms of UAS product structure and main life cycle processes – acquisition/development, operation, and support – affecting life cycle management

Table 115.2 UAS product structure (Adapted from DOD.a 2011)

| Level 1 | Level 2 |
|----------------|--|
| Air vehicle | Airframe (may include subsystem for taking off and landing in water) |
| | Propulsion (includes power/thrust to propel the vehicle; may include energy harvesting subsystems) |
| | Communications/identification |
| | Navigation/guidance |
| | Central computer |
| | Auxiliary equipment |
| | Air vehicle application software |
| Payload | Air vehicle system software |
| | Survivability (may include subsystems specific to maritime operations) |
| | Reconnaissance |
| | Electronic warfare |
| | Armament |
| | Weapons |
| | Remote Sensor System Delivery (may include subsystems for the delivery of sensors/actuators) |
| Ground segment | Payload application software |
| | Payload system software |
| | Ground Control Systems (may include stand-alone communication systems or networks; may include subsystems for mounting communication devices such as buoys or aerostats) |
| | Command and Control Subsystem |
| | Launch and recovery equipment (may include subsystems for launch and recovery from ships, land, or the ocean) |
| | Transport system components |
| | Ground segment application software |
| | Ground segment system software |

(LCM), a management process, applied throughout the life of a system, that bases all programmatic decisions on the anticipated mission-related and economic benefits derived over the life of the system (Hagan 2009).

The USDOD has a reference WBS handbook for the acquisition process (DOD.a 2011). The WBS provides a standard method for structuring systems, and their products, into logical subdivisions and levels. However, there is no standard WBS for UAS. A notational product structure for maritime UAS adapted from the reference WBS (DOD.a 2011) is presented in Table 115.2. The table concerns only levels 1–2 of the UAS product structure and does not include level 1 WBS items which are relevant, for example, to cost analysis. These level 1 items include system integration, systems engineering, system test, and evaluation; training; data; support equipment; operational/site activation; industrial facilities; and initial spares and repair parts.

Typically, the operation of a single UA requires at least one ground station, several operators, and a data link to the UA. In the simplest remotely piloted aircraft (RPA), the data link allows the pilot to operate the aircraft from the ground station. More advanced UAS have some degree of onboard automation (e.g., waypoint tracking and vision tracking of targets) to facilitate interactions with operators on the ground station and to reduce dependence on high-speed communication links.

Table 115.3 Levels of automation (Adapted from Clough 2002)

| Level | Level descriptor |
|-------|--|
| 10 | Fully autonomous |
| 9 | Battlespace swarm cognizance |
| 8 | Battlespace cognizance |
| 7 | Battlespace knowledge |
| 6 | Real-time multi-vehicle cooperation |
| 5 | Real-time multi-vehicle coordination |
| 4 | Fault/event adaptive vehicle |
| 3 | Robust response to real-time faults/events |
| 2 | Changeable mission |
| 1 | Execute preplanned mission |
| 0 | Remotely piloted vehicle |

Several taxonomies for levels of automation have been proposed; the one proposed in Clough (2002) is presented in Table 115.3. Currently, the level of onboard automation for most UAS is quite low (at most Level 4); higher level functions are currently performed by remote operators. Examples of these higher level functions, taken from C.E. Nehme et al. 2006, are the following: monitoring (weapon status, network communications, health and status of UA, sensor activity); negotiating with other stakeholders; notifying relevant stakeholders; optimal position supervision; path planning supervision; perceiving/interpreting images; perceiving/interpreting other sensor data; analyzing images; analyzing other sensor data; positive target identification; resource allocation; and scheduling and tracking target.

Several standards have been developed for interfacing with a UAS control system. These standards are targeted at facilitating interoperability among software components from multiple vendors. The North Atlantic Treaty Organization (NATO) Joint Capability Group UAV (JCGUAV) developed the Standardization Agreement (STANAG) 4586 via a standard set of messages between the air vehicles and the control stations and the control stations and external command, control, communications, computers, and intelligence (C4I) nodes (NATO 2007) and began the development of STANAG 4660 for an Interoperable Command and Control Data Link (IC2DL) to meet the need for a low bandwidth command and control link (Seagle 2011). The Joint Architecture for Unmanned Systems (JAUS) was originally an initiative promoted by the USDOD to develop an open architecture for unmanned systems (Joint Architecture for Unmanned Systems (JAUS n.d.)); several standards for a service-based framework evolved from it. Currently, it is unclear how the existing standards will address levels of automation beyond the current state of the art in UAS usage. Moreover, the role of experienced human operators in planning and execution control procedures, termed mixed-initiative planning and execution control, is still to be determined in future concepts of operation.

Open systems architecture concepts have been advocated in several unmanned air/surface/ground/underwater vehicle system (UXS) roadmaps to address the problem of interoperability (DOD.b 2011). This is because open architectures (OA) utilize a common set of interfaces and services, associated data models, robust,

standard data busses, and methods for sharing information to facilitate development. The expectations are that OA will reduce life cycle costs, enhance competition, and open new markets. Observe that the true goal of interoperability entails more than software interoperability:

The interoperability goal for Unmanned Systems is an ability to provide data, information, material, and services to and accept the same from other systems, units, or forces ... and to use the exchanged data, information, material, and services to enable them to operate effectively together. (DOD.b 2011).

Current legislation and standards only allow UAS to fly in segregated airspace. The integration of UAS into nonsegregated airspace is one of the main challenges facing UAS development and operations (Dalamagkidis et al. 2009).

In the USA the access to the National Airspace System (NAS) can only be obtained through a temporary Certificate of Authorization (COA) from the Federal Aviation Administration (FAA) or through a certification of the aircraft. A COA may take up to 60 days to obtain, thus precluding fast access to NAS through the so-called “file-and-fly” system used for piloted aircraft. Access to NAS will basically require an UA to be airworthy; to be operated by a qualified pilot/operator; and to comply with operating rules, standards, and procedures. Airworthiness is the basic requirement for any aircraft system, manned or unmanned, to enter the National Air Space (NAS). Airworthiness will require certification of the UAS.

The primary guidance for USDOD airworthiness certification is DOD (2005). The USDOD is expanding this guidance to include criteria that address UAS specific issues. UAS-unique standards are derived from NATO STANAG (2007, 2012). UAS operators will have to respond in a timely manner to commands from air traffic control, while the UAS should also have airborne collision avoidance system (ACAS) and contingency management capabilities. Regulatory compliance for UAS needs definition in national and international regulatory guidelines and standards. This is one of the reasons why some UAS manufacturers have developed hybrid manned UAS, which are basically manned aircraft with UAS capabilities, to allow rapid testing and demonstration (with a pilot onboard). The USDOD UAS Airspace Integration Plan (DOD.c 2011) outlines DOD’s approach, which is summarized by four overarching precepts: apply USA world-leading aviation expertise to UAS; conform where possible, create where needed; leverage DOD authorities and equities; and engage as one. This is an incremental approach to provide UAS critical access to a given operations profile prior to implementing a full dynamic operation solution. The end state is routine NAS access comparable to manned aircraft for all DOD UAS operational, training, and support missions. The National Aeronautics and Space Administration (NASA) *UAS Integration in the NAS Project* leverages current NASA investments to contribute capabilities that reduce technical barriers related to the safety and operational challenges associated with enabling routine UAS access to the NAS (Griner 2011). UAS access to international and foreign national airspace faces similar challenges. There are several initiatives underway worldwide to address the problem of integration in civilian airspace.

The Australian Civil Aviation Safety Authority has been very active in this area. The Civil Aviation Safety Regulations (CASR) – Part 101 Unmanned Aircraft

and Rocket Operations consolidates the rules governing all unmanned aeronautical activities into one body of legislation (Civil Aviation Safety Authority [n.d.](#)).

Europe has also been active in this field. The key civil aviation entities are the European Joint Aviation Authorities (JAA) with the European Organization for the Safety of Air Navigation (EUROCONTROL); the European Aviation Safety Agency (EASA), defined as the legal edifice for civil aviation in Europe, like its North American counterpart, the FAA; the European Organization for Civil Aviation Equipment (EUROCAE); and the Civil Aviation Authorities (CAAs) for each country. The European Defence Agency (EDA) and NATO are the most representative organizations concerning military aviation. In 2004, experts from JAA and EUROCONTROL submitted the report entitled *UAV Task Force Final Report* (UAV TASK-FORCE [2004](#)) about the regulation of civilian UAS. One year later, EASA issued an *Advance Notice for Proposed Amendment* of this document (EASA [2005](#)). In January 2007, the International Civil Aviation Organization (ICAO) started the UAS Study Group (UASSG) with the aim of developing Standards and Recommended Practices (SARPs) (ICAO [2006](#), [2007](#)). In February 2008, the United Kingdom (UK) presented the third edition of its Civil Aviation Publication CAP 722 entitled *Unmanned Aerial Vehicle Operations in UK Airspace – Guidance*. The fifth edition of CAP 722 was issued in 2012 and incorporates amendments to legislation introduced in the *Air Navigation Order* (ANO) (Civil Aviation Authority [2012](#)). In early 2011, ICAO published new recommendations to provide the fundamental international regulatory framework (ICAO [2011](#)). In 2004, the *European Military Aviation Authorities Group* (EMAAG) was established at the annual *International Military Aviation Authorities Conference*. In 2008, a forum for military airworthiness authorities – the *European Military Airworthiness Authorities* – was created under the aegis of the EDA. In January 2005, the *Délégation Générale de l'Armement* (DGA) presented version 3.0 of *UAS Systems Airworthiness Requirements* (USAR) (FMOD/DGA [2005](#)). The document was later adopted by NATO as the basis for STANAG 4671. In 2005, NATO established the JCGUAV working group which is organized into several subgroups and activities. *Flight in Nonsegregated Airspace* (FINAS) is one of those subgroups which concerns the definition of airworthiness requirements for UAV. This group approved the first draft of STANAG 4671 (NATO STANAG [2007](#)). More recently, the STANAG 4703 (for light UAS) has been developed, being also under ratification (NATO STANAG [2012](#)). In the UK, a working group created under the purview of the *Society of British Aerospace Companies* – involving industry members and representatives from CAA air traffic control (ATC) – issued, in November 1991, the document entitled *A Guide to the Procurement Design and Operation of UAV Systems*. This document provided guidelines for the development of UAS in the UK and has been used by the UK Ministry Of Defence (MOD) as the basis for the *Defence Standard 00-970 part 9*. EDA is responsible for the workgroup *Mid Air Collision Avoidance System* (MIDCAS) that has an ongoing project namesake. It has been active in what concerns integration of UAS into nonsegregated airspace, being more dedicated to the issues of detect, sense, and avoid (DS&A) technology.

Life cycle cost has been another major challenge for UAS development, operations, and support. There are several reasons for this. First, the UAS field is relatively new but it is evolving quite fast, especially in some UAS categories. Systems, capabilities, and associated cost have been changing significantly over time; and there is no standard way to estimate and report costs. Second, the distribution of UAS cost across acquisition, operation, and support categories differs significantly from that of manned aircraft making comparison between UAS and manned aircraft complicated. In addition, some UAS, notably those in the small UAS categories, enable new applications for which there is no experience in manned aviation. Moreover, existing cost comparisons between manned and unmanned aircraft have not been necessarily favorable to UAS. Third, the diversity of UAS, ranging in size from a few millimeters to tens of meters, precludes uniform cost metrics. Common cost metrics, such as weight and endurance or pound-hour per thousand dollars (pound-hour/\$k) (DOD.a 2005) fail to capture the highly nonlinear dependence on UAS diversity (Valerdi et al. 2005). Additionally, cost-per-hour quotes for UA use are often misleading in that they address only the recurring costs of actually flying the vehicle. However, nonrecurring costs, which typically include engineering, fabrication, test and integration, payload integration, vehicle transport, support team travel, and aircraft acquisition costs, must also be considered. This state of affairs makes it hard to determine cost estimation relationships (CERS) for UAS. Fourth, the unprecedented pace of technological development of key components, such as sensors and communications devices, requires unprecedented models of development with special emphasis on upgrades with short life spans. Moreover, developmental UAS have demonstrated their worth in both military (e.g., the *Global Hawk*) and scientific operations (NASA.a 2006; NASA.b 2006). In fact, UA technologies have been transitioned to operations, especially to the battlefield, at an unprecedented rate making it hard to draw the line between development and operations. New acquisition and business models may be needed. Finally, UAS technology enables new concepts of operation, notably those concerning the higher levels of automation from Table 115.3. However, the economics of coordination and cooperation for heterogeneous systems of systems is still poorly understood.

115.3 Maritime Missions

115.3.1 Taxonomy

A general taxonomy of current and future airborne maritime missions is presented in Tables 115.4 and 115.5. In spite of the large number of different missions, there is one segment that is the cornerstone of current airborne operations in the maritime domain: air maritime surveillance. Other types of missions also present new opportunities for future UAS operations.

Maritime surveillance encompasses the systematic and continuous observation of the maritime domain to achieve effective situational awareness

Table 115.4 Civilian maritime missions (Adapted from NASA.b 2006)

| Earth sciences missions | Coastal and oceanic missions | Security missions |
|---|---|--|
| Repeat pass interferometry for surface deformation | Coastal management | Search and rescue (SAR) |
| Cloud and aerosol measurements | Coastal water quality | Marine safety |
| Stratospheric ozone chemistry | Identification and tracking of marine species | Alien migrant interdiction operations |
| Tropospheric pollution and air quality | Shallow water benthic ecosystem | Ports, waterways, and coastal security |
| Water vapor and total water measurements | Wildlife habitat change | Marine environmental protection (e.g., oil spill response, marine debris) |
| Coastal ocean observations | Carbon dioxide flux | Aids to navigation |
| O2 and CO2 flux measurements | Disaster: real-time comm. predict, measure, reducing risk to responder, pre- and post-event monitoring and assessment | Drug interdiction |
| Aerosol, cloud, and precipitation distribution | | Defense readiness |
| Glacier and ice sheet dynamics | | Living marine resources (e.g., fisheries, endangered species & whale/marine mammal protection) |
| Vertical profiles of shortwave atmospheric heating rates | | Other law enforcement |
| Ice sheet thickness and surface deformation | | Ice operations |
| Imaging spectroscopy | | |
| Gravitational acceleration measurements | | |
| Antarctic exploration surveyor | | |
| Magnetic fields measurements | | |
| Cloud properties | | |
| River discharge | | |
| Cloud microphysics/properties | | |
| Focused observations – extreme weather | | |
| Forecast initialization | | |
| Hurricane genesis, evolution, and landfall | | |
| Physical oceanography, meteorology, and atmospheric chemistry | | |
| Tracking long distance transport and evolution of pollution | | |
| Cloud systems: clouds/aerosol/gas/radiation interactions | | |
| Long time scale vertical profiling of atmosphere | | |
| Global three dimensions continuous measurement of environmentally important species for assimilation in global models | | |
| Transport and chemical evolution in the troposphere | | |

Table 115.5 Military maritime missions (Adapted from Nehme et al. 2006)

| Military |
|--|
| Intelligence/reconnaissance (mapping, battle damage assessment, target acquisition, target designation) |
| Drones (target practice, decoy) |
| Transport |
| Surveillance (listening; geospatial surveillance; nuclear, biological, and chemical (NBC) sensing) |
| Communications |
| Extraction: payload extraction from a specified target. In the military, insertions typically involve cargo and/or personnel (search and rescue would fall here) |
| Insertion (payload delivery, electronic warfare) |
| Maritime security |
| Surface warfare |
| Special operations forces support |
| Maritime interdiction operations (MIO) support |

(del Pozo et al. 2010). It is one of the main applications of aerial vehicles for intelligence, surveillance, and reconnaissance (ISR). This is because maritime spaces are immense, with no physical or functional boundaries. Moreover, the average speed of any type of maritime vessel is quite low, and regions of interest extend well beyond the typical land-based electromagnetic horizon. This means that for proper surveillance and reconnaissance, one has necessarily to resort to the speed and endurance provided by airborne platforms. Also, satellite-based ISR has its own limitations. For example, detection of typical objects/activities may be difficult, if not impossible, from high orbit satellites and, notably, from geostationary assets.

Air maritime surveillance permeates military and civilian roles, covering a wide range of missions that are performed by a comprehensive set of systems operated by military forces, law enforcement agencies, and civilian organizations. Even though the requirements for long-range antisubmarine/surface aircraft have somehow diminished after the Cold War, there is an increasing growth in the number of smaller multi-mission capable platforms. This is in mainly due to the increasing demand for persistent wide area surveillance in the exclusive economic zones (EEZ), in the coastal areas, and in the planned extensions of the continental shelves of coastal countries. The relevance of air maritime surveillance has, therefore, increased over recent years. Its main roles are indicated in Table 115.6.

The importance of some of these roles is better understood by considering the fact that the economic value of the marine ecosystem has been estimated in the vicinity of U.S.\$21 tn (Costanza et al. 1997); that the global opiate and cocaine markets are valued above U.S.\$140 bn (UN 2011) – a considerable part of which is trafficked by sea; or that maritime piracy has an annual cost impact of U.S.\$7 bn to U.S.\$12 bn (Bowden et al. 2010).

Table 115.6 Maritime patrol/surveillance roles

| Maritime patrol/surveillance roles | |
|---|--|
| Military | Civilian/military |
| Intelligence, surveillance and reconnaissance | Search and rescue |
| communications, command, and control | Exclusive economic zone |
| Antisubmarine warfare | Outer continental shelf patrol |
| Anti-surface warfare | Maritime patrol and assistance |
| | Vessel monitoring and tracking |
| | Shipping lane protection and security |
| | Law enforcement |
| | Port security |
| | Environment preservation (pollution control) |
| | Customs and border control |
| | Immigration control |
| | Fisheries control |
| | Natural disaster management |

115.3.2 Current Practice

115.3.2.1 Manned Aircraft

Air maritime surveillance is traditionally performed by manned maritime patrol/surveillance aircraft (MPA/MSA). These platforms, capable of deploying an impressive array of sensors, enable wide area surveillance and thus provide an operational advantage over surface vessels.

Adding to the surveillance capabilities, MPA also provide an attack option when equipped to deliver bombs, missiles, mines, depth charges, and torpedoes. The antisubmarine warfare (ASW) and anti-surface warfare (ASuW) missions concern high-end combat roles performed by specialized, complex, and more expensive military aircraft that require extended loiter time and specialized sensor suites and weapons to engage surface and subsurface targets. Even though these aircraft can also perform the less demanding maritime surveillance missions (safety and security), their employment is considerable more expensive than that of short-/medium-range platforms.

According to the ISR Special Report 2009 (Flight Insight 2009), the role of MPA/MSA accounted for 39 % of the global share, totalling 787 aircraft worldwide, as shown in Table 115.7.

The layered maritime surveillance roles involve a tiered organization of aircraft capabilities. The first tier includes the long-range MPA, capable of weapon delivery, only available to the military forces of a selected group of countries. Aircrafts in this category include the *Boeing P-8 Poseidon*, the *Lockheed P-3 Orion*, the *Dassault Atlantique*, and the *CASA C-295MP*. The second-tier platforms include medium-range MSA, such as the *ATR-42* or the *CASA CN-235*, which can perform a variety of roles and are widely used by armed forces and coast guards. The third tier involves short range, light aircraft, designed for coastal surveillance, such as the *Dornier Do228*, the *Beechcraft King Air 200*, and the *CASA C-212*.

Table 115.7 Worldwide
ISR active fleet per region
(Adapted from Flight Insight
2009)

| Region | MPA/MSA | ISR |
|------------------------|------------|--------------|
| North America | 214 | 857 |
| South America | 55 | 112 |
| Europe | 242 | 549 |
| Africa | 9 | 42 |
| Middle East | 8 | 84 |
| Asia-Pacific | 259 | 377 |
| Total aircrafts | 787 | 2,021 |

In this decade, the USA is expected to remain the global leader in the procurement of MPA/MSA aircraft, although countries in Asia and Middle East are expected to increase their procurement for MPA/MSA as well. However, the high-end capabilities of these new multi-mission maritime aircraft come with a high price tag, not affordable to the majority of the users. Even the modernization of legacy systems like the P-3 can be quite expensive. For instance, the P-8 cost per platform is eight times higher than that of the CN-235 that equips the U.S. Coast Guard (Robertson 2012).

Even though this is an era of budget cuts, the demand for ISR aircraft seems likely to increase. *Visiongain* estimates the global airborne ISR market, including new acquisitions and upgrades, to be worth U.S.\$17.3 bn in 2011 (Visiongain 2011). The segment of maritime patrol alone is estimated to have a robust growth of more than U.S.\$6.5 bn per year, totalling U.S.\$78 bn throughout the 10-year period. Therefore, opportunity and necessity are some of the drivers that indicate significant growth opportunities in the maritime patrol/surveillance segment.

115.3.2.2 Unmanned Vehicle Systems

The proliferation of maritime missions opens new opportunities for short-/medium-range platforms dedicated to coastal/exclusive economic zone/Outer Continental Shelf (OCS) patrol. In addition, countries with ambitions to maintain some degree of sovereignty over their maritime domains may resort to short-/medium-range platforms instead of costly high-end MPAs. Considering the technological advances in the unmanned arena, namely, modularity and miniaturization of sensors and mission management information systems, there is a growing interest in the development and employment of UAS for maritime applications. This is discussed next.

Tier II MALE/HALE UA require airfields for takeoff/landing and are therefore operated from land. Those systems have increased range, loiter time, and larger payload when compared to smaller shipborne organic UAS. In addition, the employment of such platforms avoids challenging launch and recovery operations from ships in high seas. However land-based UA have some limitations for maritime operations. One of their main limitations is the compromise between the necessary

detection capability and the required capability for area coverage. Small, low-contrast (even for multispectral cameras) targets (e.g., the head of a survivor at sea) impose zoom/altitude requirements which are quite different from those pertaining to the wide area coverage requirements arising in broader maritime surveillance.

The maritime MALE segment includes systems as the *Guardian* UAS, a maritime-configured *Predator B* operated by the U.S. Department of Homeland Security, which has proven to be effective in supporting operations against several maritime threats, including counter-narcotics efforts against fast boats and submersibles. Other maritime capable MALE platforms include the Israeli-produced *Heron* UAS and the *Hermes 900* UAS equipped with a variety of sensors, with extended range, and loiter time of up to 40 h.

The most important, complex, and expensive UAS program in the HALE segment is the U.S. Navy *MQ-4C Broad Area Maritime Surveillance* (BAMS). The BAMS UAS is based upon the *Global Hawk* UAS and will provide persistent maritime and littoral ISR to naval forces up to a range of 2,000 nautical miles, complementing manned platforms such as the *P-8 Poseidon*. It is expected to achieve initial operational capability by 2016.

On the lower end of the maritime missions, there are small tactical UAS typically used as organic assets of surface vessels, which can easily be deployed to investigate contacts (typically over-the-horizon-targeting (OTHT) distance of 20/30 miles). Due to the low altitude of flight, the problem of autonomous flight in nonsegregated airspace does not arise in this tier. The main technical challenges on this tier are related to the small size of these UAS (reduced payload capabilities) and the need for launch and recovery systems. Automatic (or manual, for that purpose) recovery of these UAS from a medium-sized ship operating in rough seas is not a trivial task. Typically, this requires small, lightweight, fast, and inexpensive vehicles. This is also one of the reasons behind the development of shipboard rotary-wing UAS capable of vertical takeoff and landing (VTOL) operations.

The *ScanEagle* UAS, launched from a pneumatic catapult, is one of the most successful tactical systems used in maritime missions such as ISR, sea-lane escort and convoy protection, search and rescue, border security, communications, video, and data relay. After accomplishing the mission, it is autonomously recovered using a folding “skyhook” and catch line. In service with the U.S. Navy since 2005, the *ScanEagle* has flown nearly 250,000 h under the Naval Air Systems Command’s ISR services contract (U.S. Navy 2012).

Finally, there is a growing interest in the high-end maritime capabilities of unmanned combat air vehicles. The U.S. Navy requirement for persistent carrier-based ISR coverage to be performed by UAS capable of operating in contested airspace and performing deep penetrating strike has led to the development of an Unmanned Carrier-Launched Surveillance and Strike Aircraft (UCLASS) demonstrator program. One of the contenders, the *Northrop Grumman X-47B* Unmanned Combat Air System (UCAS) demonstrator, successfully completed its first land-based catapult launch in November 29, 2012, marking the dawn of a new era for naval aviation (Naval Air Systems Command Public Affairs 2012). In 2013, it is expected to perform autonomous carrier-based launches and recoveries.

115.4 Current Trends and How These Will Affect UAS for Maritime Operations

115.4.1 Ocean Observation and Global Sustainability

Currently, there is a pressing need for a sustained, persistent, and affordable presence in the oceans that will help us to understand and monitor how key issues such as climate change, ocean acidification, unsustainable fishing, pollution, waste, loss of habitats and biodiversity, shipping, security, and mining are affecting global ocean sustainability and stewardship (IOC/UNESCO 2011). This is not an easy endeavor. First, the oceans cover 71 % of the Earth and contain 96 % of the Earth's living space thus making ocean observation a problem at the planetary scale. Second, the oceans are still largely inaccessible, not only to humans but also to man-made devices. Third, the oceans are a communication-challenged environment: land-based communications have limited range and satellite communications are quite expensive. Fourth, although ships have been the mainstay of seagoing ocean sciences, a ship can only be at one place at a time, can only carry a small number of scientists, and can only stay at sea for limited time. In addition, ship time is very expensive. Fifth, the interior of the ocean changes faster than it can be measured with traditional sampling devices, such as shipborne sensors and drifters (Bellingham and Rajan 2007).

A sustained, persistent, and affordable presence in the oceans require innovative approaches to systems development, operations, and management. This can only be achieved with an incremental and multidimensional approach. First, there is a need to increase the number of systems (buoys, drifters, floats, etc.) in operation in the oceans and to develop and deploy new fleets of robotic vehicles for ocean observation with unprecedented spatial and temporal resolution. Second, it is necessary to network existing systems and new robotic vehicle systems for coordinated adaptation to observational needs. This entails being able to command and control networks of manned and unmanned vessels which, in turn, may form ad hoc communication networks allowing extended and cost-effective communications coverage. Observe that presently most systems at sea lack basic networking capabilities. These capabilities would allow, for example, commercial ships to act as opportunistic sensor and communication platforms, as well as "data mules" ferrying data collected by other systems that they may encounter at sea. Third, there is a need to develop unmanned vehicles capable of long duration/range missions for cost-effective spatial and temporal coverage. Fourth, there is a need for new organizational frameworks to manage and coordinate the system(s) of systems that will result from these networking trends and associated cost benefits. This poses unprecedented technological and organizational challenges to countries and international organizations. Some efforts are underway to address these challenges. For example, a network of national observatories is being coordinated to provide ocean data for the Global Ocean Observing System (GOOS, <http://www.ioc.goos.org>). Many observatories are surface or seafloor moorings with sensor arrays. Where moorings are cabled to shore for power, a few observatories

Table 115.8 Payload trends

| | |
|---|--|
| Payload | |
| Modular/standard payload mechanical, electrical, and signal interfaces will facilitate the integration of new payloads and will reduce overall costs. | |
| Survivability | Stand-alone localization components will facilitate finding downed UA at sea and may lead to new concepts of operation. Design for survivability will improve reliability and reduce costs. |
| Reconnaissance/electronic warfare | Advances in miniaturization and sensor development will lead to new sensor capabilities for small UA. |
| Remote sensor system delivery/retrieval | Developments in stand-alone ocean sensor and communication nodes will enable delivery of nodes in remote locations where ad hoc networks may be formed. In addition, UAS may revisit these locations to collect stored data and ferry it to base stations. This may reduce the cost of communications. Retrieval of devices (e.g., sensors or containers with water samples) from the ocean will lead to unprecedented fast response to episodic events (e.g., underwater volcanic eruption). |
| Payload application and system software | Developments in low-power and high-performance computer systems will enable advanced onboard intelligent data handling and processing and advanced sensor fusion algorithms. This will reduce the need for high-speed data links (for remote processing of sensor data) and also for operator's intervention. Open architecture development paradigms will enable rapid testing and prototyping and will reduce development costs. |

include buoyancy gliders as observing system components. Discussions have been underway to further develop Integrated Ocean Observing Systems (IOOS) which also include propeller-driven autonomous underwater vehicles (AUV), autonomous surface vessels (ASV), and UAS. These networking concepts are applicable to other domains such as surveillance and maritime security, where the role of UAS becomes even more relevant (McGillivray et al. 2012) (Tables 115.9 and 115.10).

115.4.2 Technological Trends

Technological trends relevant to maritime UAS in groups 1–3 from Table 115.1 are discussed next (trends for groups 4–5 are expected to follow closely those for manned aircraft and are not discussed here). This is done with respect to the UAS product structure presented in Table 115.2.

115.4.3 UAS Capabilities and Life Cycle Management

The trends discussed so far in this section will lead to new UAS capabilities and will significantly affect certification and interoperability, operations and support, and cost structures. This poses significant challenges to LCM processes.

Table 115.9 Trends affecting components of the air vehicle

| | |
|---|---|
| Air vehicle | |
| Modular/standard mechanical, electrical, and computational subsystems will facilitate systems integration and will reduce overall costs | |
| Airframe | <p>Developments in composite materials, nanotechnologies, and intelligent structures will decrease weight and improve aerodynamic performance.</p> <p>Ruggedized airframes will allow launch and recovery from the ocean, from maritime assets (e.g., ships, submarines, offshore platforms, and autonomous surface crafts), and from remote locations (e.g., islands or ice). Operations from remote locations will also be enabled by autonomous recharging and/or refuelling.</p> <p>Specialized airframe designs will allow energy harvesting from atmospheric phenomena, such as wind, and/or from the sun.</p> <p>Air-to-air refuelling subsystems will enable long duration missions and persistence.</p> <p>Design for survivability will make a major impact on overall reliability. This will be done in conjunction with advanced fault identification and handling algorithms.</p> <p>Developments in mechanical standardization will allow interoperability of launch and recovery systems.</p> <p>Developments in anti-icing and related technologies will enable UAS to operate in remote regions, especially in the high latitudes.</p> |
| Propulsion/energy storage | <p>VTOL and short takeoff and landing (STOL) technologies will facilitate deployment from maritime assets and remote areas.</p> <p>Developments in high energy density power sources, namely, hydrogen and fuel cell technologies, are expected to increase flight time, even for UA equipped with combustion engines.</p> <p>Laser and microwave technologies are expected to allow recharging of batteries on the fly for small UAS.</p> |
| Communications/identification | <p>Advances in communication standards will reduce development and operational costs.</p> <p>Trends in network-centric communication will facilitate integration in system(s) of systems frameworks.</p> <p>Advances in cognitive radio technologies and in heterogeneous communication devices will enable “smart” communications with heterogeneous assets (e.g., manned and unmanned aircraft, offshore platforms, buoys, tagged animals) through interoperated networks.</p> <p>Advanced relaying technologies, data compression, and network coding techniques are expected to increase effective communication data rates over extended ranges thus alleviating dependence on satellite communication (SATCOM) and enabling over-the-horizon communication. Aerostats and other aerial assets will allow sustained beyond line-of-sight (BLOS) communications. In addition, multiple UAS and other maritime assets may be required to form communication networks to provide communications coverage in remote environments.</p> <p>Disruptive tolerant networking (DTN) protocols will enable the delivery of data and commands in communications challenged environments. This allows for priority messaging when several transports, including SATCOM, are available. Data mules will transport low priority messages between nodes which may not be in direct communications.</p> |

(continued)

Table 115.9 (continued)

| | |
|---|--|
| Air vehicle | <p>Secure protocols and encryption will allow secure access control and control handover for networked operations.</p> <p>Identification standards will enable operations over wide geographic areas, including international waters, where assets from different countries and organizations may be present.</p> |
| Navigation/ guidance | <p>New satellite services for global positioning will complement and improve existing services through better integration and coverage.</p> <p>Developments in microelectromechanical systems (MEMS) technologies will enable the development of tactical grade low-cost inertial measurement units.</p> <p>Navigation aids mounted on naval assets will assist navigation in remote locations, in high latitudes, or in Global Positioning System (GPS)-denied environments.</p> <p>Guidance systems for energy harvesting from atmospheric phenomena will allow persistent operations.</p> |
| Central computer | <p>Miniaturization, multi-core computer architectures, and dedicated computing units will enable demanding onboard computations, such as those related to data fusion, deliberative planning, and higher levels of autonomy.</p> |
| Air vehicle application and system software | <p>Interoperability standards and open architecture frameworks will reduce development costs and facilitate true interoperability with heterogeneous assets. Interoperability will be extended to interactions with ship-based systems and with other maritime assets. This will reduce integration and certification costs.</p> <p>Developments in cyber-physical systems technologies and frameworks will enable the development of application software targeted at system of systems integration.</p> <p>Improvements in software development techniques and tight integration with onboard computer systems, with special emphasis on redundancy, will lead to more reliable flight systems.</p> <p>Advances in control design will lead to standardized maneuvers with special emphasis on maneuvers with performance and safety guarantees. Libraries of maneuvers are expected to grow with time to deliver new capabilities, namely, those concerning interactions with other assets (e.g., refuelling, forming a communications network, tracking a vessel, interacting with manned aircraft). This will allow developers to continuously extend the command set for existing UA platforms. In addition, advances in networked controls will allow tight motion coordination of multiple UAS over unreliable communication channels. All of these developments will pose new challenges to certification.</p> <p>Advances in collision avoidance technologies and concepts, together with advances in collision detection capabilities, will also facilitate certification and air worthiness.</p> <p>Advances in fault identification, intelligent system health monitoring, contingency management, and fault recovery and in controls for degraded modes of operation will improve reliability and reduce associated costs. This will lead to more reliable flight systems.</p> |

(continued)

Table 115.9 (continued)

| | |
|-------------|---|
| Air vehicle | <p>Advances in onboard automation, namely, in deliberative planning, will enable higher levels of automation (levels 5–6 from Table 115.3) and advanced forms of mixed-initiative interactions with remote operators operating from heterogeneous ground and maritime assets, including manned aircraft. This will also allow high level control over networks with intermittent connectivity.</p> <p>Code mobility technologies will facilitate continuous code upgrades for remote systems.</p> |
|-------------|---|

Table 115.10 Trends affecting the off-board components

| | |
|---|---|
| Ground/ocean segment | |
| The ground/ocean segment will expand dramatically allowing multiple ground and maritime systems to interact with UAS. This will be facilitated by advances in interoperability and standardization. | |
| Ground control systems | Trends towards networking and system of systems will enable UAS to interact with heterogeneous ground/ocean control systems ranging from more “traditional” control stations to hand-held devices, to unattended buoys in the ocean, or to other unmanned vessels. This will allow either direct real-time control or the anticipation of future interactions (e.g., new UAS mission plans may be uploaded to an ocean buoy for downloads by UAS operating in the region). |
| Command and control subsystem | <p>Advances in onboard autonomy and interoperability will allow UAS to become “more operable” thus facilitating the task of human operators and reducing training cost. Moreover, these advances are expected to enable one operator to control UAS teams, which may involve other types of maritime assets. This will be done through the specification of high level objectives and extended UAS command sets.</p> <p>Interoperability standards and open architecture frameworks will reduce development costs and facilitate interoperability with heterogeneous assets.</p> <p>Advances in command and control frameworks for system of systems frameworks will facilitate integration of UAS into higher level maritime observing systems. This will enable new concepts of operation and new business models.</p> <p>Developments in human factors technologies will allow the optimization of mixed-initiative interactions by providing better integration of human operators in system of systems frameworks.</p> |
| Launch and recovery equipment | <p>Standards for modular launch and recovery systems will facilitate operations from heterogeneous maritime assets and remote locations and will enable new concepts of operation taking advantage of geographically distributed launch and recovery locations. These may entail launch, or at least recovery, from the ocean.</p> <p>Standards and certification procedures will be developed to enable operations from maritime assets.</p> |

(continued)

Table 115.10 (continued)

| | |
|--|--|
| Ground/ocean segment | |
| Transport system components | Standardized components will facilitate transportation and minimize the cost of spare parts. |
| Ground segment application and system software | Advances in archival of data in a searchable manner, in the semantic web, and in intelligent data handling and processing techniques for high volumes of data will enable unprecedented real-time data analysis capabilities. Cloud computing technologies will provide unprecedented computational power for intensive data handling and processing techniques Advances in data fusion and visualization techniques will significantly improve situational awareness capabilities (e.g., overlay of video on top of satellite imagery from the same spot if not at the same time) Advances in interoperability will enable interactions with external systems required for the operation of UAS. |

The challenges come from the rapid pace of technological developments and from a paradigm shift in maritime operations: from single vehicle to multi-vehicle networked operations. The fast pace of technological developments, especially those affecting the software subsystems from Table 115.2, is bound to significantly affect LCM for UAS in groups 1–3 (Table 115.1) because frequent upgrades will be required to prevent obsolescence. Moreover, frequent upgrades will be facilitated by simpler certification requirements for UAS in groups 1–3. The paradigm shift will mean that UAS will become nodes in system of system networks. This will significantly change UAS maritime operations.

Several countries and organizations are developing technological roadmaps to address capabilities development in a systematic fashion and within an appropriate organizational framework. The technology roadmap (DOD.b 2011) is quite comprehensive and presents key capabilities, organized around seven themes, to be developed for the continued integration of unmanned systems into the USDOD. The projected evolution of key capabilities for each of these themes is briefly summarized next (Table 115.11).

Capabilities specific to UAS maritime operations will include ship-based and ocean-based launch and recovery (McGillivray 2010) and also retrieval of systems from the ocean. These will pose additional legal challenges, namely, when it concerns operations in international waters and classification of maritime assets (Kraska 2010).

The problem of reducing UAS life cycle costs is addressed in UAS assessment reports (Subcommittee on Unmanned Systems 2011) and roadmaps (DOD.b 2011). Several steps towards reducing the cost associated to the UAS life cycle are briefly discussed next with respect to the traditional decomposition of cost into recurring and nonrecurring costs (NASA.a 2006).

Payload integration, which is one of the largest nonrecurring costs for UAS, can be decreased through standardized interfaces. The cost of support personnel

Table 115.11 Capability roadmaps (Adapted from DOD.b 2011)

| | |
|-------------------------|---|
| Interoperability | Software reuse (2014–2015) |
| | Common ground station architecture (2014–2015) |
| | Common ground stations (2015) |
| | Integrated manned-unmanned teaming (2018) |
| | Integrated common operational picture (2019–2020) |
| Autonomy | Common autonomy capabilities across platforms (2020–2025) |
| | Environmental understanding and adaptation (2014–2015) |
| Airspace integration | Autonomous collaboration (2018) |
| | Safe operations for DOD UAS (2014–2106) |
| | Missions in low-density airspace (2014–2016) |
| | Dynamic operations for large UAS (2017–2018) |
| Communications | Dynamic operations for large- and medium-sized UAS (2018–2020) |
| | Lower size, weight, and power (SWaP) system (2013) |
| | Improved (100x) receiver design (2016) |
| | Dynamic reconfiguration (2016–2017) |
| | Multifunctional antennas (2018–2019) |
| Training | Environmental-based adaptive dynamic system changes (2019–2025+) |
| | Pre-deployment training/lessons learned before arriving the area of operations (2014) |
| | Surrogate UAS support/training improvement plan (2016–2017) |
| | Universal pilots and operators (2018–2019) |
| Propulsion and power | HP/LP power extraction (2016–2017) |
| | Extreme endurance (2018–2019) |
| Manned-unmanned teaming | Payload control (2013–2014) |
| | Control of UAV from other aircraft (2015) |
| | Unmanned wingman (2016) |
| | Manned-unmanned scout/ATK teaming (2017–2018) |
| | Unmanned scout/ATK teaming (2018–2019) |

that must accompany the UAS can be reduced by making UAS more “operable.” This requires access to NAS on a “file-and-fly” basis, simple deploy and launch mechanisms and procedures, minimal preflight and post-flight procedures, high system reliability, and increased levels of automation for flight plan development. The cost of acquisition, especially for the larger UAS, may be prohibitive for most users and countries. This opens a market for UAS service providers. On the other hand, organizations that have a constantly recurring mission (e.g., U.S. Coast Guard) would need to include acquisition costs in the mission cost. Modularity and standardization will contribute to the reduction of development costs. This will also reduce the cost of UAS upgrades. In addition, UAS will benefit from developments in autonomy, interoperability, and sensor integration for other unmanned systems.

Recurring costs are those that are directly proportional to the number of hours the UAV actually flies and include direct costs, insurance, communication support, mission planning, and data analysis. The cost of ground operators will be significantly reduced with the expected increase in levels of onboard and off-board automation. Interoperability standards will have the same impact. Interagency

coordination, asset pools, and consolidated operations centers should also contribute to cost savings (Subcommittee on Unmanned Systems 2011). New concepts of operation involving networked vehicles and launch and recovery from ships and remote locations will contribute to the reduction of transportation costs. The cost of insurance has been one of the major components of the cost per flight hour. Improvements in reliability and fault tolerance, competition, and specialized insurance policies may bring these costs down. The cost of communications is significant, especially for long-range missions where SATCOM seem to be the only available solution. Increased onboard autonomy reduces bandwidth requirements for command and control. But this does not solve the problem of the increasing bandwidth requirements arising from more sophisticated payloads. Disruptive tolerant networking (DTN) communication protocols and networked vehicles will help solve the problem of beyond line-of-sight (BLOS) communications. The cost of data analysis may still increase in the near future as sophisticated payloads become available. Automation is the answer. Developments in archival of data in a searchable manner and in algorithms for analyzing high bandwidth data streams are expected to reduce this cost.

115.5 Potential Future UAS Maritime Operations

Future UAS maritime operations will differ significantly from what is done today. This is because technological trends and projected developments of capabilities will entail a paradigm shift from single UAS operations to UAS integration into system(s) of systems. New concepts of operation will revolve around interactions, teaming, persistence, services, network behavior, and dynamic reconfiguration.

Tier II maritime MALE/HALE UAS will be operated from land, eventually from remote locations located closer to areas of interest. In the first stage of development, UAS will provide direct observation and communication services to manned and unmanned sea-based and airborne assets through encrypted communication channels. Later, teamed operations involving HALE and MALE UAS, and possibly smaller UAS, equipped with complementary sensors and communication devices and operating at different heights will provide services with adaptation of spatial and temporal resolution to observational needs, thus bridging the gap between wide area coverage requirements arising in broader maritime surveillance and detailed inspection of features of interest. In addition, the inclusion of UAS into operations including manned ships, as well as ocean observation surface buoys and underwater observation networks, enables a wider range of data collection relevant to ocean monitoring than previously possible. At a later development stage, dynamic reconfiguration of interactions and of UAS teams will allow dynamic missions in persistent operations with heterogeneous UAS.

Small tactical maritime UAS will be used as organic assets of surface vessels and submarines. Extreme endurance will allow UAS to be launched and recovered from different assets at different locations. This will enable transportation, delivery, and recovery (from the ocean) services. Additional sensing and communications

services will be provided to manned aircraft and ships. This will allow loyal wingman operations in manned-manned teaming for SAR missions, as well as in MIO support.

115.6 Conclusions

Current practices and trends in UAS for maritime operations are presented along with projections of future UAS maritime operations. This is discussed in the conceptual framework of a notational work breakdown structure (WBS), which is a standard method for understanding systems. First, a taxonomy of maritime aircraft missions and required capabilities is presented along with current practices for both manned and unmanned aircraft. Second, trends in ocean observation and in technological developments pertaining to UAS cost and capabilities are presented before discussing how these may affect future UAS capabilities and life cycle management. Third, projections of potential future UAS maritime operations are discussed, with special emphasis on new concepts of operation.

Technological trends and projected developments of capabilities will entail a paradigm shift from single UAS operations to UAS integration into system(s) of systems. The new UAS capabilities will allow concepts of operation which could have not been imagined before. These, in turn, will pose new organizational and legal challenges. For example, interoperable UAS will be capable of launch and recovery from assets from different countries. Networking is the future. The question is how?

References

- J. Bellingham, K. Rajan, Robotics in remote and hostile environments. *Science* **318**, 1098–1102 (2007)
- A. Bowden, K. Hurlburt, E. Aloyo, C. Marts, A. Lee, The economic cost of maritime piracy. One Earth Future Working Paper, 2010
- Civil Aviation Authority, CAP 722 unmanned aircraft system operations in UK Airspace – Guidance, 2012
- Civil Aviation Safety Authority. CASR Part 101. n.d. <http://www.casa.gov.au/scripts/nc.dll?WCMS:STANDARD::pc=PC.91039>. Accessed 20 Dec 2012
- B.T. Clough, Metrics, schmetrics! how the heck do you determine a UAV's, in *Performance Metrics for Intelligent Systems (PerMIS) Conference*, Gaithersburg, Maryland, USA, 2002
- R. Costanza et al., The value of the world's ecosystem services and natural capital. *Nature* **387**, 253–260 (1997)
- K. Dalamagkidis, K.P. Valavanis, L.A. Piegsl, *On Integrating Unmanned Aircraft Systems into the National Airspace System: Issues, Challenges, Operational Restrictions, Certification, and Recommendations* (Springer, New York, 2009)
- F. del Pozo, A. Dymock, L. Feldt, P. Hebrard, F. Sanfelice di Monteforte, Maritime surveillance in support of CSDP – the wise pen team final report to eda steering board. European Defence Agency, WISE PEN TAM, 2010
- DOD, *MIL-HDBK-516B Department of Defense Handbook Airworthiness Certification Criteria*, Department of Defense, 2005

- DOD, *FY2009–2034 Unmanned Systems Integrated Roadmap*, Office of the Secretary of Defense US DOD, 2009
- DOD.a, *Unmanned Aircraft Systems Roadmap: 2005–2030*, Office of the Secretary of Defense USA DOD, 2005
- DOD.a, *MIL-HDBK-881 Department of Defense Handbook Workbreakdown Structure*, Department of Defense, 2011
- DOD.b, *The Unmanned Systems Integrated Roadmap FY2011–2036*, Office of the Secretary of Defense USA DOD, 2011
- DOD.c, *DoD UAS Airspace Integration Plan*, DOD, 2011
- C.E. Nehme, M.L. Cummings, J.W. Crandall, *A UAV Mission Hierarchy*, Human and Automation Laboratory, Massachusetts Institute of Technology, Cambridge, 2006
- EASA, *Policy for Unmanned Aerial Vehicle (UAV) Certification, A-NPA, No. 16/2005*, Rulemaking Directorate, European Aviation Safety Agency, Köln, 2005
- R. Fearing, *Micromechanical Flying Insect* (2007), <http://robotics.eecs.berkeley.edu/~ronf/mfi.html/>. Accessed 2012
- Flight Insight, *ISR Special Report 2009*, Flight Insight, 2009
- FMOD/DGA, *Unmanned Air Vehicle System Airworthiness Requirement (USAR). Version 3.0*, FMOD/DGA, 2005
- J. Gertler, *U.S. Unmanned Aerial Systems*, Congressional Research Service, 2012
- J. Griner, UAS integration in the NAS project: Project overview. *Integrated Communications, Navigation and Surveillance Conference (ICNS)*, p. 1–22, Herndon, Virginia, USA, 2011
- G. Hagan, *Glossary Defense Acquisition Acronyms and Terms* (Virginia 22060–5565: Defense Acquisition University Press, Fort Belvoir, 2009)
- ICAO, ICAO exploratory meeting on UAVs. Working Paper 2, the Sixteenth Meeting of the APANPIRG ATM/AIS/SAR Sub-Group (ATM/AIS/SAR/SG/16). ICAO, 2006
- ICAO, Progress report on unmanned aerial vehicle (UAV) work. ICAO AFI Planning and implementation regional group sixteenth meeting (APIRG/16), 2007
- ICAO, *Unmanned Aircraft Systems (UAS)*, International Civil Aviation Organization, 2011
- IOC/UNESCO, *A Blueprint for Ocean and Coastal Sustainability*, Paris, 2011
- JAUS. *Joint Architecture for Unmanned Systems (JAUS)*. n.d. <http://en.wikipedia.org/wiki/JAUS>. Accessed 2012
- J. Kraska, The law of unmanned naval systems in war and peace. *J. Ocean Technol.* **5**(3), 43–68 2010
- P. McGillivray, J. Borges de Sousa, R. Martins, Connecting the dots: networking maritime fleets of autonomous vehicles for science and surveillance, in *Marine Technology Reporter*, 2012, pp. 33–38
- P. McGillivray, Design considerations for launch and recovery of autonomous systems from ships, including Coast Guard icebreakers, in *Launch & Recovery*, Arlington, 2010
- NASA.a, *Earth Observations and the Role of UAVs: A Capabilities Assessment*, Civil UAV Assessment Team NASA, 2006
- NASA.b, *Earth Observations and the Role of UAVs: Volume 2*, Civil UAV Assessment Team NASA, 2006
- NATO, *STANAG 4586 Second Edition, Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability*, NATO Standardization, NATO, 2007
- NATO STANAG, *NATO STANAG 4671 Unmanned Aerial Vehicle Systems Air Worthiness Requirements*, NATO, 2007
- NATO STANAG, *STANAG 4703 LIGHT UAS AIRWORTHINESS REQUIREMENTS*, NATO, 2012
- Naval Air Systems Command Public Affairs, First X-47B UCAS Catapult Launch Makes Naval Aviation History. 29 Nov 2012 http://www.navy.mil/submit/display.asp?story_id=70864. Accessed 12 Dec 2012
- QinetiQ, 2009, <http://www.qinetiq.com/what/products/Documents/Zephyr-UAV.pdf>. Accessed 2012
- S.Robertson, *Future Maritime Surveillance*. House of Commons, Defence Committee, UK Parliament, 2 Apr 2012

- D. Seagle, NATO Joint Capability Group on UAVs, in *2011–2012 UAS Yearbook – UAS: The Global Perspective*, 9th edn. 46/216, 2011
- Subcommittee on Unmanned Systems, Federal Unmanned Systems: Status, Issues, and Recommendations, 2011
- U.S. Navy, *Scan Eagle UAV Conducts First Flight On LPD-Class Ship*. <http://www.defencetalk.com/scan-eagle-uav-conducts-first-flight-on-lpd-class-ship-45890/>. Accessed 12 Dec 2012
- UAV TASK-FORCE, *UAV TASK-FORCE Final Report a Concept for European Regulations for Civil Unmanned Air Vehicles*. The Joint JAA/EUROCONTROL initiative on UAVs, 2004
- UN, *World Drug Report 2011*, United Nations Office on Drugs and Crime, United Nations, New York 2011
- R. Valerdi, J. Merril, P. Maloney, Cost metrics for unmanned aerial vehicles, in *16th Lighter-Than-Air Systems Technology Conference and Robotic Balloon Systems Conference*, American Institute of Aeronautics and Astronautics, 2005
- Visiongain, *The Airborne ISR Market 2011–2021*, Visiongain, 2011

Autonomous Remote Sensing of Invasive Species from Robotic Aircraft 116

Ali Haydar Göktoğan and Salah Sukkarieh

Contents

| | | |
|------------|---|------|
| 116.1 | Introduction | 2814 |
| 116.2 | Weed Surveillance and Management Life Cycle | 2816 |
| 116.3 | Avionics Architecture | 2817 |
| 116.4 | Aquatic Weeds Surveillance and Management Using Helicopter UAV | 2819 |
| 116.4.1 | Helicopter UAV | 2821 |
| 116.4.2 | Weed Detection and Classification | 2824 |
| 116.5 | Woody Weed Surveillance and Management Using Fixed-Wing and Helicopter UAVs | 2825 |
| 116.5.1 | Fixed-Wing UAV | 2827 |
| 116.5.2 | Woody Weed Surveillance Field Experiments | 2829 |
| 116.6 | Cacti and Sleeper Weeds | 2830 |
| 116.7 | Conclusion | 2831 |
| References | | 2832 |

Abstract

This chapter presents an Unmanned Aircraft System (UAS) which consists of a helicopter, a fixed wing Unmanned Aerial Vehicles (UAVs), and their supporting infrastructure. This UAS, which was used in a number of environmental research experiments focused on autonomous remote sensing, detection, classification, and management of invasive species, weeds, in Australia.

The annual cost of weeds to the Australian economy is estimated at A\$4 billion. Over the last few years, experiments have been performed at three geographically distant regions of Australia to evaluate if the UAS can be used as a cost-effective tool in management of invasive species. In these experiments

A.H. Göktoğan (✉) • S. Sukkarieh

Australian Centre for Field Robotics (ACFR), School of Aerospace, Mechanical & Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW, Australia

e-mail: agoktogan@acfr.usyd.edu.au; salah@acfr.usyd.edu.au

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,

2813

DOI 10.1007/978-90-481-9707-1_72,

© Springer Science+Business Media Dordrecht 2015

three distinct families of weeds, growing on three different types of terrain were investigated.

In the first group of experiments, a helicopter UAV equipped with a high-resolution imaging payload was flown over difficult to reach water channels and wetlands for detection of aquatic weeds. The second set of experiments was performed in large, relatively flat rangelands to map woody weed infestations. A low-flying fixed-wing UAV platform was also operated over a designated cattle grazing area and collected high-resolution aerial imagery. Weed infestation maps were produced from aerial imagery using machine learning techniques. By incorporating the human subject matter experts and farmers into the decision-making process, weed management plans produced. An autonomous helicopter was then tasked with spraying herbicides on aquatic weeds and dispensing granular herbicides on top of the selected woody weeds. The third set of experiments was focused on the airborne detection of wheel cacti on remote mountainous terrain using fixed-wing aircraft. Cacti infestation maps were generated and compared with data collected by weed experts on the ground. Successful results of these experiments are encouraging and suggest that robotic aircrafts in a properly designed UAS can play an important role in environmental robotic science.

116.1 Introduction

Unmanned Aerial Vehicles (UAVs) have been used in the military domain since the First World War (Cook 2007). Wide use of UAVs and other robotic platforms in conflict zones has resulted in a substantial military robotics industry (Singer 2009). On the other hand, the commercial and civilian applications of UAVs are rapidly evolving. Market forecasts indicate that currently limited market share of civilian UAVs is promising to have a sustainable growth for foreseeable future (Cox et al. 2004). Precision agriculture (Herwitz et al. 2002), environmental monitoring (Dunbabin and Marques 2012), wildlife surveying (Körner et al. 2010; Watts et al. 2010), Antarctic ecology research (Lucieer et al. 2010), rangeland management (Rango et al. 2006), agronomical research and crop health assessment (Nebiker et al. 2008), and weed detection, classification, and management are some of the civilian areas in which UAVs have proven to have potential uses (Bryson et al. 2011b; Göktoğan et al. 2010).

Introduction of non-native plants to Australia has threatened the ecological environment, rangeland and primary production industries. Invasive species have major impact on the environment; they destruct the native habitats, hinder the growth of the native species, cause land degradation, and significantly reduce the agricultural, horticultural, and forestry productivity. Australia has a fragmented landscape, thus the indigenous ecosystems are at a higher risk of being overcome by the invasive species or weeds (Nuberg et al. 2009). Weeds are rapidly spreading invasive plants that have an adverse effect on the environment and can have serious economic impacts. Although some native plants can also become weeds if the right conditions arise, the overwhelming majority of weeds in Australia are those which are introduced into Australia from other continents, mainly due to human activities.

The annual cost of invasive species or weeds to the Australian economy as a loss in agricultural revenue is in the order of A\$4 billion (Pash 2007; Sinden et al. 2004). Besides the major economic effect, natural fauna, flora, biodiversity, native landscape, and irrigation also suffer from rapidly spreading invasive species. This is why management of invasive species, weeds, has national significance (NRMMC 2007). Work presented in this chapter is focused on surveillance and management of three different families of weeds: (a) aquatic weeds, (b) woody weeds, and (c) cacti.

In parallel with an increasing public environmental awareness and education, there is an increasing trend toward development and utilization of new technologies for better environmental monitoring and management (Dunbabin and Marques 2012). Management strategies of invasive species can be grouped into three categories (a) prevention, (b) early detection and rapid assessment, and (c) control, containment, or eradication (Virtue et al. 2001).

In recent years, commercial satellite imagery for civilian use has become more accessible and is being used in assessing natural resources. Typical satellite imagery available to the civilian domain has approximately 2–45 m resolution per pixel. These images are useful for a high-level overview of large landscapes. It is even possible to detect large weed infestations from using satellite imagery (Everitt et al. 2008). However, weed management strategies aim for early detection of weed patches before they turn into problematic weed infestations. Current satellite imagery cannot offer the needed spatial and temporal resolution for early detection and rapid assessment of small weed clusters.

On the other hand, aerial imagery from human piloted aircrafts can provide better than 1 m spatial resolution imagery, which can be useful for providing ground truth for the satellite data as well as investigation of relatively large rangelands. Using higher-resolution cameras and flying at lower altitudes can yield even higher-resolution imagery. However, it is not safe to operate human-piloted aircrafts in narrow corridors around natural obstacles such as valleys, trees, or bushes, even if ultralight aircraft are used (Grigg et al. 1997). High-resolution aerial imagery from human-piloted aircraft is being successfully used for weed detection and assessment, particularly at late weed phenological stages (Yang and Everitt 2010; López-Granados 2011). Furthermore, human piloted agricultural crop-dusting airplanes have a long history of being used in pest insect and weed management since the end of the First World War (Eldon and Lemmer 1965). However, the high operational cost of human piloted aircraft is an important limiting factor for weed management practices.

Therefore, when a centimetre or better resolution is desired in aerial imagery, UAVs can be tasked with intelligent and adaptive sampling for an effective *close sensing* mission without putting human operators into risk (Watts et al. 2010; Göktoğan et al. 2010). As it is addressed in the following sections, robotic aircraft as a part of an appropriately designed Unmanned Aircraft System (UAS), can play an important roles in all three categories of weed management strategies. In the context of this chapter, the term UAS refers to not only the UAV platform but also all entities utilized in UAS operations such as all onboard avionics, communication subsystems, the ground control station (GCS), data analysis tools, as well as human operators.

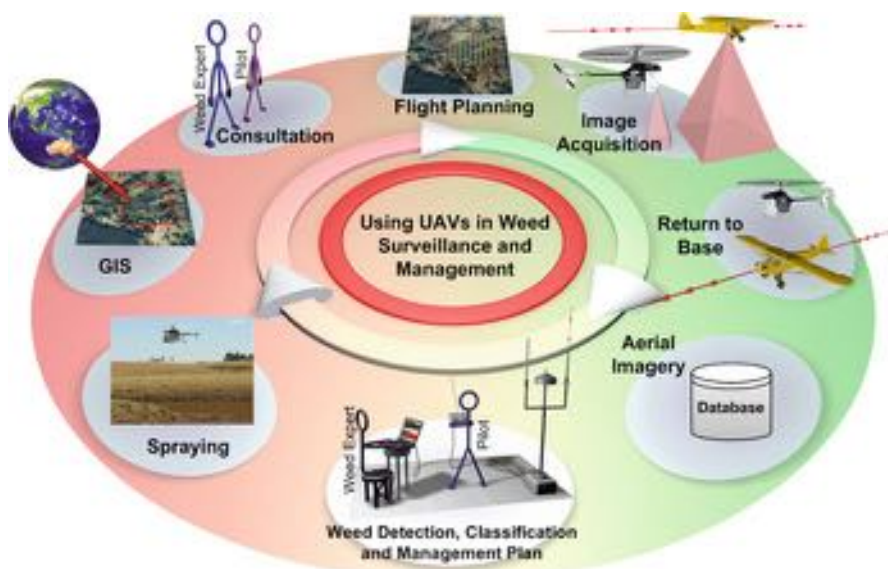


Fig. 116.1 System life cycle of using UAVs in weed surveillance and management

This chapter addresses a set of novel applications of autonomous remote sensing of invasive species from robotic aircrafts and presents the overall system view. The emphasis is on the system perspective and the focus on the connections, interactions and dependencies between components, rather than the UAV platforms and data processing algorithms as a disconnected set (Fig. 116.1).

116.2 Weed Surveillance and Management Life Cycle

As illustrated in Fig. 116.1, the lifecycle of a typical weed surveillance mission starts with some a-priori estimate of the weed location provided by land owners, farmers and weed experts. Weed experts generally keep this information in their geographical information system (GIS).

Prior to UAV flights, weed experts and UAV operators identify the regions of interest. Initial planning is based on the data from GIS, environmental conditions, weed sightings, and/or data from satellite imagery, aerial imagery obtained from manned flights, but above all, it is based on the tacit knowledge of the subject matter experts.

UAVs must be operated in accordance with the rules and regulations applicable to the particular type of UAV and the operation region. Hence, during the mission and flight planning phase, UAV operators must work closely with the Civil Aviation Safety Authority (CASA). Based on the draft situation model and environmental constraints, such as topography of the region, obstacles, wind directions, and so forth, flight plans are prepared on the ground and uploaded to the UAV autopilot for

autonomous surveillance. Alternatively, for small line-of-sight operations, a human pilot may remotely control the UAV to fly over the suspected regions for data collection. In a typical image acquisition mission, system attributes such as flight paths, flight height, flight speed, focal length of the camera, and image acquisition rate, desired image overlap between image frames of geographically close regions etc. are all closely related with each other. Mission planning is therefore a multi criteria decision making process optimising the flight plan and system configuration. It aims to maximise the return from the UAS, while minimizing the fuel, flight time and risk associated with the flight operation. Once each UAV image acquisition flight over a specified region is finished, and the UAVs have returned to the base, data is downloaded to aerial imagery databases at the ground station for processing.

The initial phase of the data processing pipeline is image registration/geo-referencing. This process identifies the relation between image pixels and geographical coordinates. It can utilise accurately surveyed control points on the ground and/or the position and orientation estimates of the airborne camera at the time of image capture. Ground control points may confer higher accuracy compared to a direct method of geo-referencing airborne images using logged navigation data. However, as it is mentioned in the following sections, manual ground surveying of weed-infested aquatic habitats and steep cliffs of cacti infestations on hilly terrains would not always be practically possible. Compared to larger manned aircraft, small UAVs are limited in options of carrying highly accurate sensors. These limitations can be reconciled in UAV flight planning by aiming to achieve relatively smooth and repeatable flights and acquiring more overlapped aerial imagery.

Current civil aviation rules and regulations imposed by the CASA have tight restrictions on the spraying of chemicals or dispensing of herbicide granules from aircrafts. The relevant acts provide details about the endorsement of the aircraft and onboard human pilot for the spraying operations. Although these rules may not be directly applicable to unmanned operations, it is difficult to reach a conclusion in relation to UAV operations for spraying or dispensing herbicides. Thus, in order to demonstrate the proof-of-concept, a water-soluble non-toxic dye is used for spraying operation and non-toxic granules dispensed instead of real herbicides.

Other regulatory and operational challenges are mainly related to the integration of new unmanned systems into existing regulated airspace. Currently, sharing the same civilian airspace between manned and unmanned aircrafts is not routine. Therefore, rules and regulations in relation to UAS operations are conservative to safeguard public safety. Although it may take many years to come, it is envisaged that prolonged safety records of new UASs will help to change existing regulations to allow regular UAS operations without compromising overall aviation and public safety.

116.3 Avionics Architecture

Avionics systems used both in the helicopter and fixed-wing UAV platforms have the same high-level system architecture. Figure 116.2 illustrates the major architectural building blocks. UAVs are fitted with an autopilot module, a navigation computer,

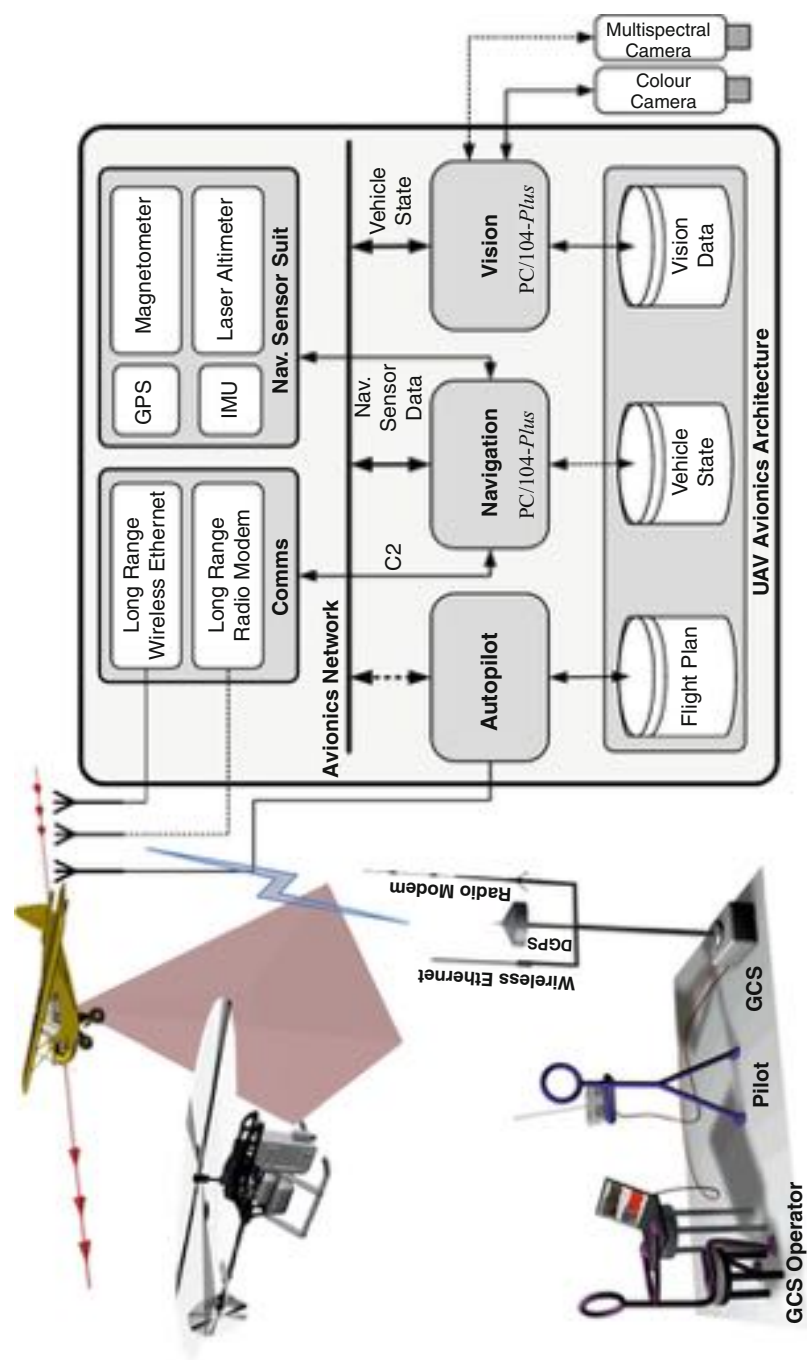


Fig. 116.2 Typical high-level avionics system architecture for UAVs used in weed surveillance and management experiments

and a vision computer. The navigation and vision computers are both in PC/104-*Plus* form factor, and the autopilot module is built around an embedded microcontroller board. They are linked together via an onboard network, and they are also able to communicate to the GCS via the communication subsystem.

In order to accurately localise and geo-reference the weeds and other feature of interests (FoI) in the aerial imagery, an accurate estimate of the UAV position and orientation, i.e., attitude are needed. The navigation computer provides the position and attitude estimate at a rate of up to 20 Hz by fusing sensor data from the navigation sensor suite. Optionally, a downward pointed laser range finder can be used as a laser altimeter, and its readings can be incorporated into the navigation solution to better estimate the flight height above ground. Both the raw sensor data and the navigation solution are logged on onboard solid-state disks. All the logged data is accurately time stamped for further analysis. The navigation solution is published on the avionics network for other subsystems. The navigation data is also available to the GCS. The GCS uses the navigation solution to update the GCS screen in real time to indicate the position and attitude of the RUAV to maximise the situation awareness of the GCS operator and the pilot (Göktoğan and Sukkarieh 2009).

The vision subsystem is tasked to control vision cameras, acquire image frames, time stamp each frame, and save them into solid-state disk for further analysis. The UAVs vision subsystem consists of a PC/104-*Plus* computer and a digital video camera interfaced via IEEE 1394 (FireWire). Optionally, an additional multispectral camera can be added to the vision subsystem.

Presented UAVs are equipped with two separate wireless communication subsystems. The long-range radio modem establishes a full-duplex link between UAV and the GCS. It is mainly used for the transfer of telemetry, command, and control (C2) packages. If this communication link is lost for a predefined period of time, then the autopilot switches its mode and initiates a communication recovery procedure. This procedure consists of loitering over a predefined place with a predefined path so that it can try to re-establish communications. If communication cannot be established after a communications time-out period, either the flight abort procedures or auto-landing procedures can be executed.

The wireless Ethernet link between a UAV and the GCS provides wider bandwidth compared to the radio modem link. However, this link has shorter communication range, and it is subject to frequent dropouts. Hence, flight and mission critical data are not transmitted via wireless Ethernet, but via a long-range radio modem unit. Wireless Ethernet link generally utilised for the transmission of high bandwidth sensor data such as sample image frames.

116.4 Aquatic Weeds Surveillance and Management Using Helicopter UAV

Aquatic weeds, as the name implies, often spread in aquatic habitats. However, they can also survive in land habitats for long duration of time and thrive in the right conditions. Two particular aquatic weeds, namely the alligator weed (*Alternanthera*

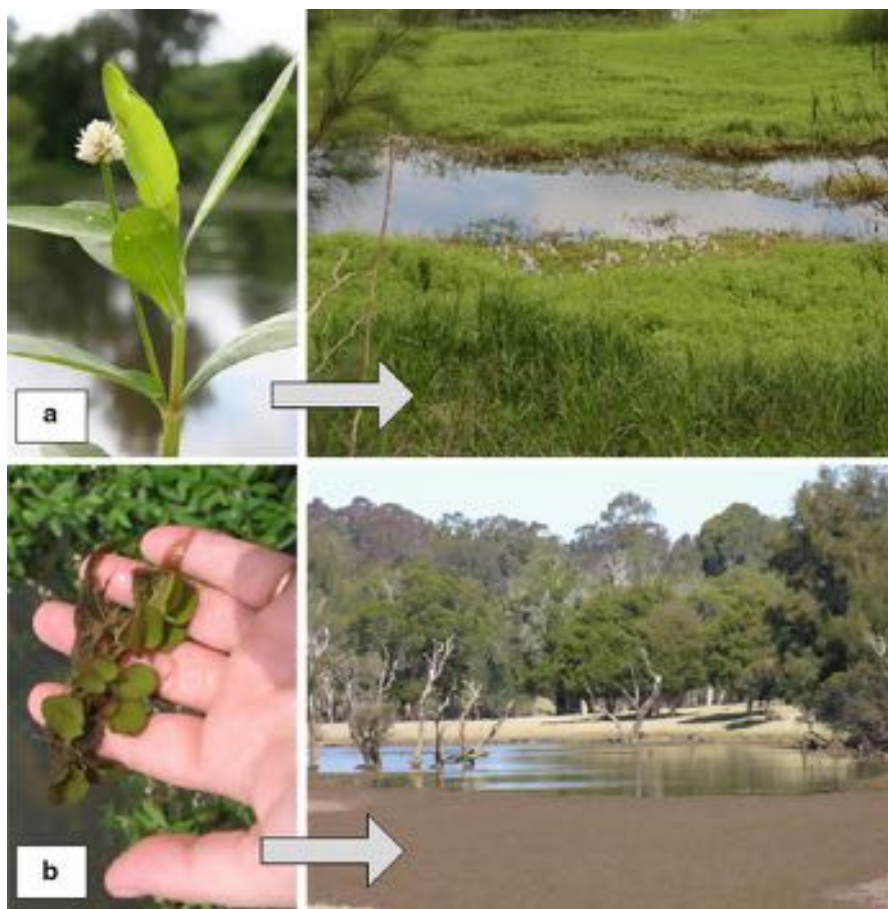


Fig. 116.3 An irrigation channel blocked by aquatic weeds (a) alligator weed (*Alternanthera philoxeroides*) and (b) salvinia (*Salvinia molesta*)

philoxeroides) and salvinia (*Salvinia molesta*) (Sainty and Jacobs 2003), are studied. As shown in Fig. 116.3, by creating thick blankets of vegetation across the water surface, these weeds decrease oxygen concentrations in water to a level that is ultimately be detrimental to bottom-growing native vegetation and fish populations.

Alligator weed is extremely invasive, and has a high potential for rapid spreading over large areas. Furthermore, it can also invade both aquatic and land habitats. Salvinia can cause devastating environmental effects and severely affect riparian ecosystems, water quality, wildlife, and the surrounding primary industries. Given the right conditions, salvinia exhibits an extraordinary growth rate; it can double its dry weight in two and a half days. Holm and East-West Center (Holm and Center 1977) describes salvinia as one of the world's worst weed.

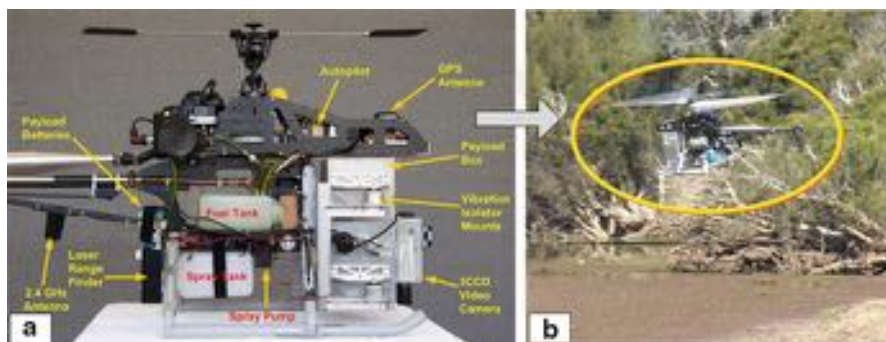


Fig. 116.4 (a) Close-up view of the UAV frame with the payload box and onboard electronics, (b) G18 helicopter UAV flying over the salvinia-infested aquatic habitat

Compared to fixed-wing UAV platforms, helicopters provide better manoeuvrability over irregularly shaped aquatic habitats (Australia Law 2009). A modified G18 helicopter UAV platform was used in surveillance of aquatic weed infestations in the Killarney Chain of Ponds in Pitt Town, New South Wales. The same platform was later equipped with a spraying mechanism to spray water-soluble non-toxic dye to demonstrate the capability of spraying herbicides on selected regions (Sukkarieh et al. 2009; Göktoğan et al. 2010; Sukkarieh 2008, 2009, b).

116.4.1 Helicopter UAV

The helicopter UAV used in the presented applications is based on a locally manufactured, commercially off the shelf (COTS), G18 industrial helicopter platform (Fig. 116.4). In order to increase its robustness for field experiments in remote locations, a range of major modifications have been made to increase its reliability, endurance, and payload capacity (Göktoğan et al. 2010).

The basic specifications of a modified G18 based helicopter UAV is tabulated in Table 116.1. With its 1.8 m main rotor diameter, and 15 kg maximum take-off weight (MTOW), this platform can be classified as small-to-midsize helicopter. Depending on the vision payload and/or the herbicide carried onboard, the endurance varies between 20–120 minutes. This flight time is sufficient and well suited to existing operational practices of weed management personnel.

As shown in Fig. 116.4, the front section of the platform is modified to accommodate the mission payload which consists of a navigation computer, autonomous flight control system, image processing equipment, vision camera, and the radio communication units. The entire payload is placed in a custom-made enclosure and mounted to the main body using vibration isolation mounts. The isolators had to be symmetrically located around the centre of gravity of the payload to ensure the rolling and pitching motions were not introduced by the vibration.

Table 116.1 Basic specifications of modified G18 helicopter UAV

| Helicopter UAV | |
|---------------------|-----------------------------------|
| Overall length | 2.0 m |
| Overall height | 0.63 m |
| Main rotor diameter | 1.8 m |
| Max takeoff weight | 15 kg |
| Engine | 26cc Zenoah (modified) |
| Speed | 0–100 km/h |
| Endurance | 20 min to 2 h (payload dependant) |

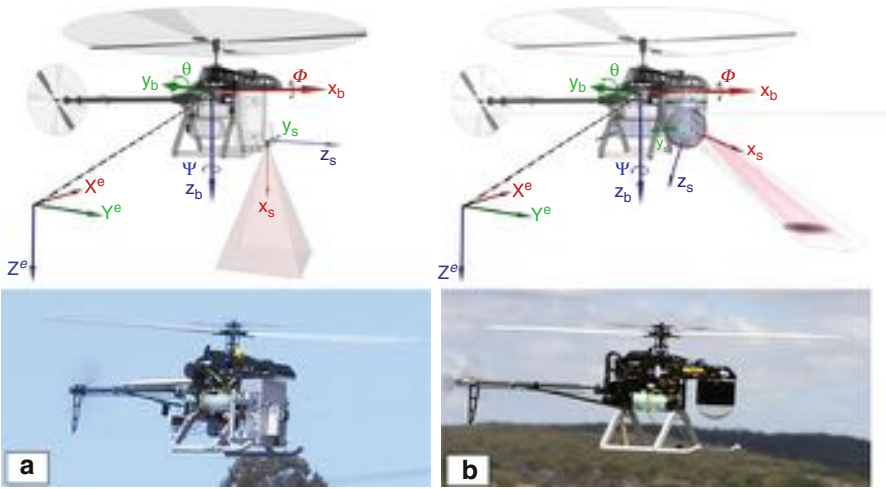


Fig. 116.5 Helicopter UAV in two different configurations. (a) UAV carrying the payload box and a fixed, downward-looking monocular camera. (b) UAV without payload box, with a gimbaled camera. Upper illustrations show the Earth, body, and sensor coordinate reference frames and how camera frustum and image footprint are related

Additional mounting points were also installed for the Global Positioning System (GPS) antenna, the magnetometer, and the communications antenna.

The G18 helicopter has a large frontal volume that can be utilised to carry different vision payloads (Fig. 116.4). For aerial image acquisition operations, two different types of camera attachments have been tested. Figure 116.5 illustrates that, in the first configuration, a downward-looking monocular vision camera is mounted in front of the payload box. This is the most commonly used configuration as it provides images with minimum slant angle. Alternatively a gimbaled camera can be mounted to the payload bay for more flexible camera-pointing applications, in which the feature of interest can be observed without hovering above the target. Coordinate reference frames used to geo-reference the acquired imagery are illustrated in Fig. 116.5.

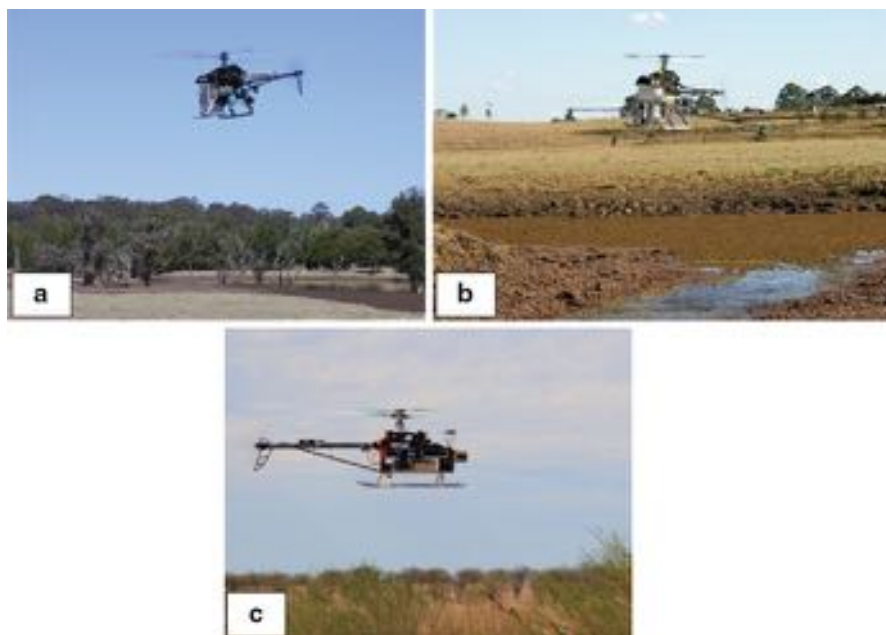


Fig. 116.6 (a) Helicopter UAV equipped with vision payload and downward-looking camera, flying over aquatic habitat for an aerial image acquisition mission, (b) the same helicopter with spraying booms attached, flying low over salvinia-infested water channel spraying a water-soluble non-toxic dye, and (c) helicopter without vision payload, reconfigured with granule dispenser, flying over woody weeds

In addition to two alternative vision payload configurations, the presented helicopter UAV can be easily configured for different missions. As shown in Fig. 116.6, it can be configured to carry a high-resolution, downward-looking monocular camera along with a high-accuracy navigation sensor for aerial image acquisition missions. Logged navigation data can later be used for geo-referencing the captured images. In this configuration, spray booms can also be attached to each side of the helicopter.

The booms are designed to reduce vibration caused by the rotor down draft. Ball joints were installed and the boom hinges to allow for movement of the spray booms when in contact with the ground during takeoff and landing. As shown in Fig. 116.4, the spray liquid tank, spray pump, control system, and power supply are positioned located under the centre of gravity of the helicopter. Hence, changing weight due to spraying operation does not have adverse effects on flight stability. For the proof-of-concept demonstrations, a dye was added to water to enable visualisation of the spray patterns and area covered.

For granule-dispensing missions, the vision payload box can be replaced with a controlled droplet applicator (CDA). The CDA unit consists of a plastic granule flask and an electromechanically driven, custom-designed dosage controller valve and dispensing outlet. In a woody weed management mission, the helicopter can

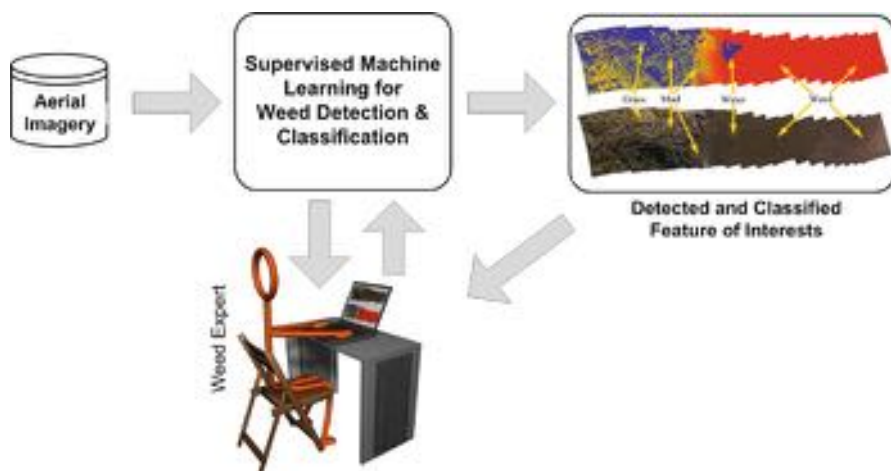


Fig. 116.7 Supervised learning process for weed detection and classification processes

be tasked to hover over the targeted weed cluster and the onboard computer can activate the CDA unit for dispensing predetermined doses of granule herbicides. Weed experts often prescribe just a few grams of weed killer, granular herbicide for a reasonable size woody weed. Thus, the helicopter UAV can be used to target large number of woody weeds in each flight. As has been mentioned in previous sections, due to existing civil aviation rules and regulations in relation to UAV operations, non-toxic granules were dispensed instead of real herbicides.

116.4.2 Weed Detection and Classification

As it is highlighted in Fig. 116.1, keeping a human in the loop, particularly in decision-making tasks, is an important feature of the weed surveillance and management life cycle. Utilisation of weed experts' input into the weed detection, classification, and mapping process is an important requirement for the presented works. In order to utilize the tacit knowledge of weed experts, supervised machine learning techniques are used. In supervised learning the user provides input and the corresponding output data to the machine learning system. The aim of the system is to establish (i.e., learn) a model for the relationships between input and output data sets. The learnt model can then be used to predict the output for any new input data.

Figure 116.7 illustrates the role of the human weed expert in the supervision of the weed detection and classification process. In this process, using a small set of aerial imagery, the weed expert interactively marks a number of regions as FoI and labels them as alligator weed, salvinia, water, mud, etc. The supervised machine learning algorithm takes these labelled learning samples and processes the rest of the aerial imagery to detect the same or similar FoIs.

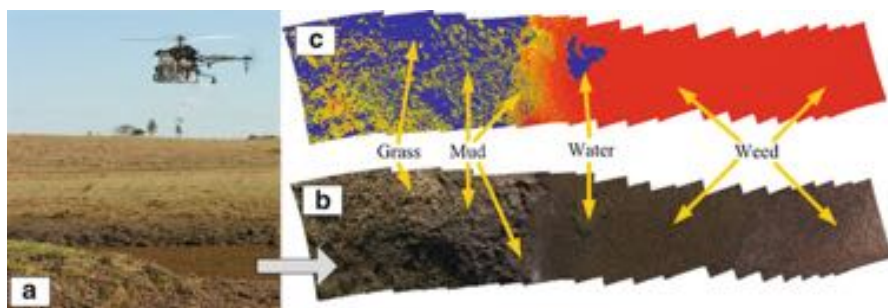


Fig. 116.8 (a) The helicopter UAV over a weed-infested water channel acquiring aerial imagery. (b) Stitched mosaic of images. (c) The result of the offline weed classification process performed on the same stitched image mosaic. Colors represent the weed probability; *red* indicates high probability; *yellow* is medium probability; and *blue* is low probability

For the aquatic weed detection and classification mission, Support Vector Machine (SVM) technique has been applied to the geo-referenced and stitched aerial image mosaic. The SVM has also been used in a number of different ecological studies (Trebar and Steele 2008; Huang et al. 2002). The aim of the SVM is to map input vector into a multidimensional feature space and locate optimal hyperplane boundaries between classes (Göktoğan et al. 2010; Christopher 1998; Vapnik 2000). Classification results are presented to weed experts in the form of color-coded the probability distributions maps (Fig. 116.8). Weed management plans are made based on these infestation maps. Comparison of the ground truth with the output of classification results proved that the SVM can be used as an effective method for aquatic weed detection. However, further experiments should be performed to develop more effective ways for human-machine interfaces and tuning of the machine learning system.

116.5 Woody Weed Surveillance and Management Using Fixed-Wing and Helicopter UAVs

Woody weed infestations that cover large open rangeland areas are difficult to control, and their management is very costly and time-consuming (Bryson and Sukkarieh 2011b; Reid et al. 2011; Hung et al. 2010, 2012). In the second case study presented in this chapter, mesquite (*mesquite prosopis*), parkinsonia (*Parkinsonia aculeata*), and prickly acacia (*Acacia nilotica*) infestations around Williams Outstation and Carrum in the Julia Creek region of Northwest Queensland are targeted (Fig. 116.9).

Mesquites have a major impact on pastoral and grazing rangelands. They generally grow on floodplains, along watercourses, and aggressively replace grasslands.

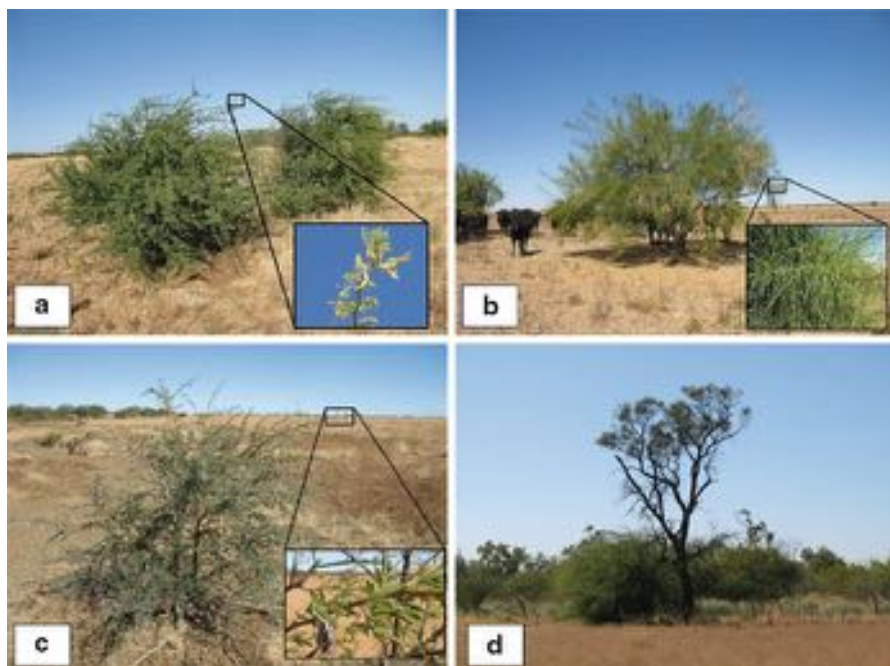


Fig. 116.9 Woody weeds: (a) mesquite (*mesquite prosopis*), (b) parkinsonia (*parkinsonia aculeata*), (c) prickly acacia (*acacia nilotica*), and (d) a native tree around Julia Creek, in Northwest Queensland

They can form impenetrable, thorny, dense bushes around watercourses and restrict native animals' and stock access to water. Feral animals can easily spread mesquite seeds. Parkinsonia is also a thorny woody weed, and its floating seeds can easily spread over large distances in wet seasons. Prickly acacia is introduced to Australia from India and Pakistan as an ornamental. It is dispersed by stock and human activities. These woody weeds outcompete native vegetation which is a major concern to the grazing industry.

In this work, 3D maps of the woody weed infestations are created. The 3D map of the terrain is generated by triangulating a set of dense feature points between subsequent image frames based on the optimal pose data. The pose estimation done by processing IMU and GPS data using an Extended Kalman Filter (EKF) for each image frame captured. Image frames were processed for to extract scale-invariant feature transform (SIFT) features. The extracted SIFT features were matched between consecutive image frames. Identified species are labelled in the infestation maps. Both a fixed-wing and a helicopter UAV platform are used for the surveillance and management missions. A fixed-wing UAV performed broad acre surveillance, and a helicopter UAV performed precision targeting around the weed (Bryson et al. 2011a).



Fig. 116.10 The fixed-wing, 1:3 scale J3 Cup UAV platform used in woody weed and wheel cacti surveillance missions. (a) Typical GCS setup in a remote country site in Northwest Queensland. (b) The UAV taking off, under a human pilot control, from an unsealed road in the bush. (c) The UAV in autonomous flight. (d) The UAV landing onto an unsealed road

116.5.1 Fixed-Wing UAV

A majority of weed surveillance and management UAV missions are performed in remote locations in the Australian bush without dedicated runway. Thus the selected UAV platform must be able to take-off from and land on a dirt surface. Figure 116.10 shows a typical rangeland in Northwest Queensland in which woody weed surveillance and management UAV missions are performed.

With its conventional tractor configuration, the J3 Cub aircraft is a suitable platform for short field operations on rough surfaces such as unsealed roads in unimproved areas. Furthermore, the J3 Cub has a large-size and weight-carrying capability which were required for the payload. Hence, a COTS, 1:3 scale J3 Cub airframe was chosen for use in the performed weed surveillance missions. It then underwent a series of extensive modifications. The main sensor and navigation payload is mounted in the upper fuselage (Fig. 116.11). To isolate it from the engine vibration, isolator mounts were selected based on the vibration frequency, payload mass, and desired level of isolation. This was necessary to isolate the payload from the vibration to increase the reliability of the electronics, improve the quality of the inertial measurement data, and permit the video camera to use slower shutter speeds without motion blur. The fuselage was widened with custom-made fibreglass fairings to incorporate the new mounts and to allow the required movement of the payload.

The internal payload camera required a section of the fuselage bottom to be removed. This was replaced with a Perspex window. A special mounting was made so this could be installed exactly perpendicular to the camera's optical axis to reduce



Fig. 116.11 (a) The fixed-wing 1:3 scale J3 Cub UAV platform with the onboard avionics and downward-pointing color camera, (b) The UAV flying over woody weed infestation

Table 116.2 Basic specifications of modified 1:3 scale J3 Cub fixed-wing UAV

| Fixed-wing UAV | |
|--------------------|---------------------------------------|
| Wing span | 3.58 m |
| Max takeoff weight | 27 kg |
| Engine | Desert aircraft 100cc (twin cylinder) |
| Cruise speed | 90 km/h |
| Max speed | 145 km/h |
| Endurance | 2 h |

reflections. It was also designed to be easily removable for cleaning or to enable the camera lens to be changed. The entire fuselage bay surrounding the camera was painted matt black and sealed off with black tape to prevent the entry of ambient light which may have caused glare on the Perspex window, compromising the quality of the camera images.

To improve the ground handling, a more durable, semi-scale undercarriage was manufactured. This features a bungee shock absorber-type mechanism as per the full scale J3 Cub. For the main wheels, 6 in. radius, low-pressurized soft tires are used. The tail wheel and its steering mechanism were also replaced with a heavier duty design. The final weight of the aircraft was nearly double its design weight, increasing the stall speed and thus the takeoff and landing distance. The aircraft wings were modified to incorporate flaps. Multiple settings allow additional lift at takeoff and additional lift and drag upon landing. A larger engine than standard was also selected to reduce the takeoff distance. The engine-mounting box was designed to incorporate engine isolator mounts to reduce the effects of vibration on the airframe and payload. The final specifications of the aircraft are listed in Table 116.2.

The modular nature of the payload installation in the J3 Cub allows for different payload types to be carried in the future. Its ability to fly for several hours and operate from short, unimproved areas makes it eminently suitable for continuing research of this type.

116.5.2 Woody Weed Surveillance Field Experiments

The main objectives of this experiment were to produce accurate, geo-referenced 3D maps of the terrain of the operational area and to detect and classify the different species of vegetation present in the map. As illustrated in Figs. 116.2 and 116.11, the fixed-wing UAV platform was equipped with a sensor payload box consisting of an IMU, a GPS receiver, and a downward-pointed color camera. Fused IMU and GPS data allowed for the estimation of the UAV's position and orientation at any given time. This information was later used to geo-reference of the detected woody weeds. The same fixed-wing UAV platform configuration was later is used in mapping of cacti infestations over inaccessible semiarid rangelands.

The fixed-wing UAV operated on an approximately 24 km² of test area. In order to survey differences in coverage and spatial resolution, flight plans were prepared in such a way that the fixed-wing UAV operated at two different altitudes: 100 m above ground level (AGL) and 500 m AGL. Based on the camera resolution of 1,024 × 768 pixels and the use of a 12.5 mm camera lens with a field of view of approximately 28 × 22°, this provided an expected spatial resolution of approximately 3.7 cm/pixel and a ground footprint of 38 m at 100 m altitude and, a spatial resolution of approximately 18.6 cm/pixel and a ground footprint of 190 m at 500 m altitude. In this configuration, a flight time of 1 h at a speed of approximately 90 km/h was appropriate. Based on the calculated footprints, a flight pattern of successive swaths along the flight area with approximately 20 % lateral overlap was performed in each flight. In a single flight, the system could cover an area of 4,000 × 600 m at 100 m altitude and an area of 6,000 × 1,500 m at 500 m (MLA 2010).

Once the fixed-wing UAV had completed its mission, the acquired imagery, logged IMU, and GPS data were downloaded at the test site. SIFT features were extracted from overlapping frames and an EKF estimated the UAV pose at the time of each image frame capture. The pose data was used for the calculation of 3D coordinates of the extracted image features. A non-linear least squares bundle adjustment phase was then used to optimize both the trajectory and 3D feature point estimates using all of the raw IMU, GPS, and extracted vision feature data. The 3D feature points and corresponding vision frames were finally used to build a surface model of the terrain on which the collected imagery was used to texture the surface (Bryson et al. 2011b).

In order to detect woody weed crowns, the colour and texture, and the shadows were used. This is a three-phase process involving image segmentation, object detection, and classification. The image segmentation phase divides the image into three different classes based on the color and texture information. Extracted color and texture features are grouped into a single feature vector consisting of 3 color channels and 30 texture channels. Using SVM as the classifier, the feature vectors are then labelled to identify their classes as object, shadow, and background (Hung et al. 2011, 2012). The detected objects are further classified as mesquite, parkinsonia, and prickly acacia and native trees (Fig. 116.12) (Bryson and Sukkarieh 2011b).

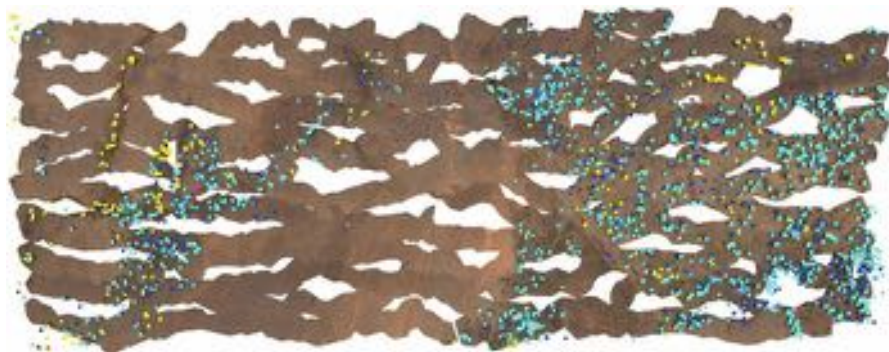


Fig. 116.12 A stitched image mosaic of a woody weed infestation. Colored dots indicate classified tree crowns of woody weed and native vegetation. (Blue, parkinsonia; cyan, mesquite; red, mimosa bush; yellow, native tree)

116.6 Cacti and Sleeper Weeds

Wheel cactus (*Opuntia robusta*), which is native to northern Mexico, was introduced to Australia as a hedge and garden plant. It is well adapted to the semiarid environment of southern Australia. As with other weeds, high density of wheel cacti has adverse impact on the native flora and fauna (Patrick et al. 2010). Trial flights for surveillance of the wheel cactus infestations were performed near Oraparinna, in the Flinders Ranges of South Australia (Fig. 116.13). In conjunction with the flight trails, a ground-based survey of cacti in the flight area was also performed. Coordinates of cacti in the test area were recorded into a database as the ground truth. This data was later used for the validation of the detection and mapping algorithms (Bryson and Sukkarieh 2011a).

The operation region has hilly terrain with steep cliffs causing prolonged strong turbulent airflow at low altitudes. Undulated terrain also limits the line-of-sight operations for low-flying UAV. Hence, the fixed-wing 1:3 scale J3 UAV platform was used at an altitude approximately 100 m above the highest point of the terrain.

Unlike previously mentioned aquatic weed and woody weed surveillance and management missions, in the cacti detection mission UAV helicopter was not used for the dispensing of herbicides. The mission was limited to classification and mapping wheel cactus infestations. The main objective of this mission was to demonstrate that UAV platforms with low-cost vision sensor payloads can be used for identification and mapping of cacti in remote locations.

The success of the detection and classification was limited. Although cacti were successfully detected, the classification results also contain a fairly large number of false detections. This can be attributed to the shape of the cacti. As shown in Fig. 116.13, the wheel-like structure of cacti has a narrow horizontal cross section compared to its side view. This makes it difficult to see from the downward-pointed



Fig. 116.13 Wheel cacti (*Opuntia robusta*) on a hilly terrain near Oraparinna, in the Flinders Ranges of South Australia

camera on the fixed-wing UAV platform. Hence, it is envisaged that increasing the spatial resolution of aerial imagery would help improve the classification results.

Wheel cacti are more easily distinguishable from the horizontal perspective. The round, wheel-like appearance of the cacti can be utilized in the future, if aerial imagery can be obtained from a more horizontal perspective using a gimbaled camera on low-flying helicopter UAV platform as shown in [Fig. 116.5b](#). However, strong turbulent airflow due to the undulated terrain is a limiting factor for small-scale helicopter operations (Bryson and Sukkarieh [2011a](#)).

116.7 Conclusion

This chapter presented an Unmanned Aircraft System (UAS) developed as a cost-effective tool for the surveillance and management of weeds. The UAS incorporates a G18 helicopter and a 1:3 scale J3 Cub fixed-wing UAV platform, onboard autopilot, navigation and vision payloads, ground control station (GCS), post-mission image processing and machine learning systems, and operational personnel.

There is a significant scope for this UAS to be actively used in weed surveillance. The UAV platforms provide a means of traversing large distances, accessing difficult

ground and aquatic operation areas. Weed detection, classification, and mapping techniques using aerial imagery continue to improve the efficiency of the weed management life cycle. However, the future holds many new regulatory, operational, and scientific challenges. These form part of the ongoing operational and scientific research in the area.

In addition to specific weeds mentioned in this chapter, there are also another group of weeds called *sleeper weeds* which are invasive plants that become established in a relatively small region and have not spread widely (Brinkley and Bomford 2002). Their limited spread may be due to limited pollinators, or the exotic natural enemies introduced along with the weed or unsuitable environmental and climate conditions (Scott et al. 2009). Early identification, eradication or containment of sleeper weeds before they start spreading rapidly and destroying the native biodiversity is both important environmentally and economically (Cunningham et al. 2004). Successful trials performed on aquatic weed, woody weed, and cacti infestations suggest that, a UAS can have potential to be used in the assessment and identification of risk categories of sleeper weeds.

The UAS has been operated over weed a variety of weed-infested land and aquatic habitats in different regions of Australia. Successful results of these experiments are encouraging and suggest that robotic aircrafts in a properly designed UAS can play an important role in environmental robotics science.

Acknowledgments This work is supported in part by the Australian Centre for Field Robotics (ACFR) funded by the New South Wales State Government and Land & Water Australia (LWA) as a part of the “Defeating the Weed Menace” (DWM) program; by Meat and Livestock Australia (MLA) under project code B.NBP.0474; “UAV Surveillance Systems for the Management of Woody Weeds,” the Australian Weeds Research Council (AWRC) under project code AWRC08-04; the ARC Centre of Excellence program and Linkage Project LP0989291, funded by the Australian Research Council (ARC); and the New South Wales State Government. Authors express their appreciation to Judy Lambert, DWM program Coordinator; Andrew Petroeschovsky, National Aquatic Weeds Coordinator, NSW Department of Primary Industries, Grafton Agricultural Research & Advisory Station; Luke Joseph, Farm & Dam Control Pty Ltd; and SunWater for their invaluable advice. This project would not be possible without the dedicated work of Mitch Bryson, Calvin Hung, Alistair Reid, Nick Lawrence and Zhe Xu, and the support of ACFR’s Aerospace Group members, particularly without the state-of-the-art engineering support from the team of Jeremy Randle, Steve Keep, and Muhammad Esa Attia.

References

- T.R. Brinkley, M. Bomford, *Agricultural Sleeper Weeds in Australia What Is the Potential Threat?* (Bureau of Rural Sciences, Kingston, 2002)
- M. Bryson, S. Sukkarieh, Detecting Cacti – by using unmanned aerial vehicles and innovative classification algorithms, vol. RIRDC publication no 11/018, RIRDC Project No AWRC 08-04, 2011a
- M. Bryson, S. Sukkarieh, *UAV Surveillance Systems for the Management of Woody Weeds* (Meat & Livestock Australia Limited, Locked Bag 991, North Sydney, 2011b)
- M. Bryson, A. Reid, C. Hung, T. Abuhashim, S. Sukkarieh, Using unmanned aerial vehicles for mapping, classification and monitoring of invasive weeds, in *International Symposium on Remote Sensing of the Environment (ISRSE)*, Sydney, 2011a

- M. Bryson, S. Sukkarieh, A.H. Göktoğan, J. Randle, M.E. Attia, S. Keep, A. Reid, C. Hung, T. Abubashim, Z. Xu, N. Lawrance, T. Lupton, A. Kassir, J.J. Chung, G. Brooker, N. Tahir, Robotic aircraft for remote sensing of the environment. Paper presented at the workshop on robotics for environmental monitoring at the IEEE/RSJ international conference on intelligent robots and systems (IROS), San Francisco, 2011b
- J.C.B. Christopher, A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **2**(2), 121–167 (1998). doi:10.1023/a:1009715923555
- K.L.B. Cook, The silent force multiplier: the history and role of UAVs in warfare, in *Aerospace Conference*, Big Sky (IEEE, 2007), pp. 1–7
- T.H. Cox, C.J. Nagy, M.A. Skoog, I.A. Somers, Civil UAV capability assessment (Draft Version). NASA, 2004
- D.C. Cunningham, C.B. Simon, G. Woldendorp, M.B. Burgess (2004) A framework for prioritizing sleeper weeds for eradication. *Weed Technol.* **18**(ArticleType: research-article/Issue Title: Invasive Weed Symposium/Full publication date: 2004/Copyright © 2004 Weed Science Society of America):1189–1193
- M. Dunbabin, L. Marques, Robots for environmental monitoring: significant advancements and applications. *IEEE Robot. Autom. Mag.* **19**(1), 24–39 (2012). doi:10.1109/mra.2011.2181683
- W.D. Eldon, G.F. Lemmer, Origins of aerial crop dusting. *Agric. Hist.* **39**(3), 123–135 (1965)
- J. Everitt, R. Fletcher, H. Elder, C. Yang, Mapping giant salvinia with satellite imagery and image analysis. *Environ. Monit. Assess.* **139**(1), 35–40 (2008). doi:10.1007/s10661-007-9807-y
- A.H. Göktoğan, S. Sukkarieh, Distributed simulation and middleware for networked UAS. *J. Intell. Robot. Syst.* **54**(1–3), 331–357 (2009). doi:<http://dx.doi.org/10.1007/s10846-008-9269-7>
- A.H. Göktoğan, S. Sukkarieh, M. Bryson, J. Randle, T. Lupton, C. Hung, A rotary-wing unmanned air vehicle for aquatic weed surveillance and management. *J. Intell. Robot. Syst.* **57**(1), 18 (2010). doi:10.1007/s10846-009-9371-5
- G.C. Grigg, A.R. Pople, L.A. Beard, Application of an ultralight aircraft to aerial surveys of kangaroos on grazing properties. *Wildl. Res.* **24**, 359–372 (1997)
- S.R. Herwitz, L.F. Johnson, J.C. Arvesen, R.G. Higgins, J.G. Leung, S.E. Dunagan, Precision agriculture as a commercial application for solar-powered unmanned air vehicles, in *AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles*, Portsmouth, 20–23 May 2002, pp. 2002–3404
- Holm LGDLPea, Center E-W, *The World's Worst Weeds: Distribution and Biology*, vol. xii (Published for the East-West Center by the University Press of Hawaii, Honolulu, 1977)
- C. Huang, L.S. Davis, J.R.G. Townshend, An assessment of support vector machines for land cover classification. *Int. J. Remote Sens.* **23**(4), 725–749 (2002). doi:10.1080/01431160110040323
- C. Hung, M. Bryson, S. Sukkarieh, A novel vision-based tree crown and shadow detection algorithm using imagery from an unmanned airborne vehicle, in *15th Australian Remote Sensing & Photogrammetry Conference (ARSPC)*, Alice Springs, 2010, p. 12
- C. Hung, M. Bryson, S. Sukkarieh, Multi-class predictive template for tree crown detection. *ISPRS J. Photogramm. Remote Sens.* **68**(0), 170–183 (2012). doi:10.1016/j.isprsjprs.2012.01.009
- C. Hung, M. Bryson, S. Sukkarieh, Vision-based shadow-aided tree crown detection and classification algorithm using imagery from an unmanned airborne vehicle, in *34th International Symposium on Remote Sensing of Environment (ISRSE)*, Sydney, 2011
- F. Körner, R. Speck, A.H. Göktoğan, S. Sukkarieh, Autonomous airborne wildlife tracking using radio signal strength, in *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, 18–22 Oct 2010, pp. 107–112
- F. López-Granados, Weed detection for site-specific weed management: mapping and real-time approaches. *Weed Res.* **51**(1), 1–11 (2011). doi:10.1111/j.1365-3180.2010.00829.x
- A. Lucieer, S. Robinson, D. Turner, Using an unmanned aerial vehicle (UAV) for ultrahigh resolution mapping of antarctic moss beds, in *15th Australian Remote Sensing & Photogrammetry Conference (ARSPC)*, Alice Springs, 2010, p. 12
- MLA, Weed control takes to the skies (Episode 12). MLA (2010), <http://www.youtube.com/watch?v=LhBGyURY3do>. Accessed 12 Mar 2012

- Nebiker, S. Annen, A. Scherrer, M. Oesch, D. A light-weight multispectral sensor for micro UAV - opportunities for very high resolution airborne remote sensing. In proceedings of XXI ISPRS congress, *In International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*; Beijing, China, 2008; vol. 37, pp. 1193–1200
- NRMCC, *The Australian weeds strategy: a national strategy for weed management in Australia* (Natural Resource Management Ministerial Council, Canberra, 2007). Accessed from <http://nla.gov.au/nla.cat-vn4230723>
- I. Nuberg, B. George, R. Rei, *Agroforestry for Natural Resource Management* (CSIRO Publishing, Collingwood, 2009)
- R. Pash, Chopper weed. Contours Australian Government, Department of Agriculture, Fisheries and Forestry (2007)
- G.P.J. Patrick, D.C. Agnew, D. Cooke, A model for state wide co-ordinated management of invasive cacti, in *17th Australasian Weeds Conference*, Christchurch, 2010, pp. 287–290
- A. Rango, A. Laliberte, C. Steele, J.E. Herrick, B. Bestelmeyer, T. Schmutge, A. Roanhorse, V. Jenkins, Research article: using unmanned aerial vehicles for rangelands: current applications and future potentials. *Environ. Pract.* **8**(03), 159–168 (2006). doi:10.1017/S1466046606060224
- A. Reid, F. Ramos, S. Sukkarieh, Multi-class classification of vegetation in natural environments using an unmanned aerial system, in *IEEE International Conference on Robotics and Automation Shanghai International Conference Center*, Shanghai, 9–13 May 2011
- G.R. Sainty, S.W.L. Jacobs, *Waterpants in Australia*, 4 edn. (Sainty and Associates Pty. Ltd., Sydney, 2003)
- J. K. Scott, K. Batchelor, N. Ota and P. Yeoh, *Sleeper and alert weeds: where will they awaken as climate changes*, Land & Water Australia, 2009, p. 4
- J. Sinden, R. Jones, S. Hester, D. Odom, C. Kalisch, R. James, O. Cacho, The economic impact of weeds in Australia: report to the CRC for Australian Weed Management/ by Jack Sinden ... [et al.]. CRC for Australian Weed Management technical series no. 8 (CRC for Australian Weed Management, Glen Osmond, 2004). Accessed from <http://nla.gov.au/nla.cat-vn3107287>
- P.W. Singer, *Wired for War: The Robotics Revolution and Conflict in the Twenty-First Century* (Penguin, New York, 2009)
- S. Sukkarieh, *Cost-Effective Surveillance of Emerging Aquatic Weeds Using Robotic Aircraft* (Lands & Water Australia, Australian Centre for Field Robotics, Sydney, 2008)
- S. Sukkarieh, *Aquatic weed surveillance using robotic aircraft*, Land & Water Australia, 2009, p. 8
- S. Sukkarieh, *Thinking bush: what do you get when you strap a rotor to a weed controller?* Land and Water Australia, 8, 40–41, 2009
- S. Sukkarieh, Terrestrial and aquatic weed detection using robotic aerial systems. Paper presented at the 10th Queensland Weed Symposium, Capricorn Resort, Yeppoon, 26–29 July 2009b
- S. Sukkarieh, M. Bryson, A.H. Göktoğan, M.E. Attia, S. Keep, T. Lupton, J. Randle, Cost effective surveillance of emerging aquatic weeds using robotic aircraft. *ACFR* (2009), <http://www.acfr.usyd.edu.au/research/projects/aerospace/FundRes/Weed-Helicopter-2008.wmv>. Accessed 12 Mar 2012
- M. Trebar, N. Steele, Application of distributed SVM architectures in classifying forest data cover types. *Comput. Electron. Agric.* **63**(2), 119–130 (2008)
- V.N. Vapnik, *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science, 2nd edn. (Springer, New York, 2000)
- J.G. Virtue, R.H. Groves, F.D. Panetta, *Weed Risk Assessment* (CSIRO Publishing, Collingwood, 2001)
- A.C. Watts, J.H. Perry, S.E. Smith, M.A. Burgess, B.E. Wilkinson, Z. Szantoi, P.G. Ifju, H.F. Percival, Small unmanned aircraft systems for low-altitude aerial surveys. *J. Wildl. Manage.* **74**(7), 1614–1619 (2010)
- C. Yang, J.H. Everitt, Comparison of hyperspectral imagery with aerial photography and multispectral imagery for mapping broom snakeweed. *Int. J. Remote Sens.* **31**(20), 5423–5438 (2010). doi:10.1080/01431160903369626

Calvin Coopmans, Brandon Stark, Austin Jensen, YangQuan Chen,
and Mac McKee

Contents

| | | |
|-------|---|------|
| 117.1 | Introduction | 2836 |
| 117.2 | Cyber-Physical Systems (CPS) | 2838 |
| 117.3 | Personal Remote Sensing (PRS) | 2839 |
| 117.4 | CPS Enabled by SUAS: UAS-Based Real-Time Irrigation Control (WaterWatch) | 2845 |
| 117.5 | CPS Enabled by SUAS: Algal Bloom Tracking for Alternative Energy | 2847 |
| 117.6 | CPS Enabled by SUAS: Invasive Species Management | 2849 |
| 117.7 | CPS Enabled by SUAS: Tagged Fish Tracking and Wildlife Management | 2852 |
| 117.8 | Other Ideas for UAS Sensing-Enabled CPSs | 2853 |
| 117.9 | Conclusions and UAS-Enabled CPS Outlook | 2856 |
| | References | 2858 |

C. Coopmans (✉)

The Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State University, Logan,
UT, USA

e-mail: c.r.coopmans@ieee.org

A. Jensen

The Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State University, Logan,
UT, USA

Utah Water Research Laboratory, Utah State University, Logan, UT, USA

e-mail: austin.m.jensen@ieee.org

B. Stark • Y. Chen

Mechatronics, Embedded Systems and Automation (MESA) Lab, School of Engineering,
University of California, Merced, CA, USA

e-mail: brandon.stark@gmail.com; yqchen@ieee.org; ychen53@ucmerced.edu

M. McKee

Utah Water Research Laboratory, Utah State University, Logan, UT, USA

e-mail: mac.mckee@usu.edu

Abstract

Cyber-physical systems (CPS) are tightly coupled systems of hardware and software providing large-scale, closed-loop control or management of high-level, complex dynamical systems. Effective cyber-physical systems require data for operation. Unmanned aircraft systems (UAS), including small UAS (SUAS), can acquire unique data about physical processes, allowing a CPS to efficiently accomplish real-time monitoring, management, or control and actuation tasks. In this chapter, several examples of SUAS sensing-enabled CPS scenarios are presented, enabling adaptive management and effective control of complex physical systems such as water distribution based on measurement of soil moisture and crop evapotranspiration, radio tag-based tracking of fish, alternative energy harvesting, mapping of invasive plant species, and airborne plume (pollution) tracking. This chapter serves as a motivator for potential wide-spreading UAS applications where UAS are used as mobile sensors and/or mobile actuators in large-scale, closed-loop CPS.

117.1 Introduction

As unmanned aircraft systems offer enhanced functionality and utility and improved levels of safety, applications for these versatile platforms will rapidly become abundant in the coming years. Civilian applications for UAS are an emerging field, one that has great potential and the possibility of explosive growth, as aerial vehicles are fully utilized for such applications. The U.S. Congress is aware of these possibilities and has given the Federal Aviation Administration (FAA) a deadline of 2015 for the creation of rules allowing the inclusion and integration of UAS into the greater U.S. National Airspace System (NAS). This integration will not be determined by policy alone; the challenges and opportunities of addressing ethics and the public perception of UAS (Oh 2012) will forever be a part of aviation and robotics as levels of integration and technology progress. The FAA's UAS access rules for the NAS may eventually be in the familiar form of "file and fly," and domestic UAS operations will be commonplace. With the proper certification and standards, government and commercial operations will include regular UAS use, including integration into large-scale operations such as cyber-physical systems.

The possibilities of civilian UAS use in areas ranging from science exploration to high-valued precision agriculture are abundant. Applications in law enforcement, search and rescue operations, intelligence gathering, and fire-fighting applications have been championed as the future of UAS, but such applications are only the beginning: remote sensing and many other technical activities will benefit greatly from the utilization of UAS. In fact, real remote sensing with small SUAS is not only feasible but has significant advantages. Much like the personal computing revolution of the past 40 years, advancements in SUAS will bring remote sensing down to the personal level, allowing for unprecedented levels of knowledge and empowering change for the greater good at smaller scales of operation than ever before.



Fig. 117.1 UAS-sized sensors for water management: thermal and shortwave infrared imagers (Infrared Cameras Incorporated [2012](#); Goodrich ISR Systems [2012](#))



Fig. 117.2 A 10 cm-resolution mosaic of thermal imagery collected by the AggieAir™ small unmanned aerial vehicle system (Coopmans et al. [2012](#))

The rise of SUAS has auspiciously coincided with significant advances in sensors. Currently there are sensors capable of scientific data collection that is perfectly suited to fly in small, low-cost unmanned aerial vehicles (UAVs). As in Fig. 117.1, shortwave infrared (SWIR), thermal infrared (TIR), as well as near-visible infrared (NIR) and visible light cameras can be used to improve agricultural, civil, and scientific data collection in new and exciting ways. Thermal imagery (Fig. 117.2) can be utilized for vegetation monitoring (Berni et al. [2009](#)), and multispectral imagery can be utilized effectively in crop monitoring (Hunt et al. [2010](#)) as well as many wetlands applications (Jensen et al. [2011](#)).

Cyber-physical systems are the future of large-scale sensing and actuation. A key factor in CPS success is data that is detailed enough temporal and spatial resolution for effective operation, and the use of SUAS as mobile sensing platforms will deliver data more frequently and with higher quality than ever before. This chapter attempts to illustrate some perspective UAS-enabled CPS applications given the

current and proximal capabilities of SUAS and to inspire further research toward the goal of CPS. This chapter is organized as follows: Sect. 117.2, cyber-physical systems, introduces the concepts of a CPS and how an SUAS can be utilized within it. The following section, personal remote sensing, focuses on the underlying concepts of remote sensing at the personal level and the challenges associated with it. Sections 117.4–117.8 describe examples of CPS enabled by SUAS. Finally, concluding remarks and discussion on the future of UAS are presented in Sect. 117.9.

117.2 Cyber-Physical Systems (CPS)

Control and management of the real world are challenging, complex tasks. Processes interact at many spatial and temporal scales but when dealt with properly can be controlled just as a simple traditional process. With appropriate choices of sensors and actuators, closed-loop control can be implemented around any controllable process, limited only by the desired outcomes of the controlled system and the complexity of the controller.

A cyber-physical controller (Fig. 117.3) is a general structure that includes a real-world process (the “plant” in controls terminology), estimations of the dynamics of a process under control (an “observer”), and a controller. This allows closed-loop feedback to be instantiated for control of processes that might be outside of traditional definitions of control, such as the moisture in a farm field or the population of fish at an important aquatic site.

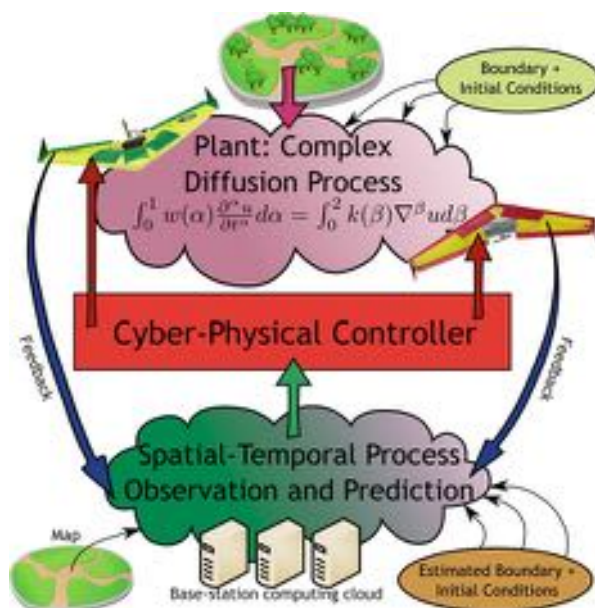
Models of physical processes can be used to help predict the behavior of a system in a particular scenario or the reaction of a process to a given input. Cyber-physical control is enabled by modeling (usually digitally) a target process and utilizing that model to make decisions and exert control over the process by way of an actuator or a set of actuators. Since cognitive computer modeling and control of complex processes require higher computational resources for higher accuracy, a computational cloud may be employed, distributing modeling tasks spatially and increasing reliability. Using advanced techniques such as fractional calculus (Jiao et al. 2012), sophisticated models of physical processes can be constructed with proper boundary conditions and, when refined by quality data, can be reliably and robustly controlled in real time.

Control loops cannot be closed without feedback data. Due to the nature of cyber-physical control, the process under control can be multivariate, and therefore, cyber-physical controllers (CPCs) potentially have very broad requirements for sensing. Since in any closed-loop system better sensing may equate to better control, providing CPCs with the most useful “best” data is of primary importance.

This is one requirement UAS can effectively fulfill in CPS: they can generate data with high spatial and temporal resolutions about complex processes, allowing for CPC loops to operate on new and heretofore hard-to-control plants.

UAS can also act as actuators in closed-loop scenarios. Chemicals, such as herbicides, can be applied with high precision to areas determined to be in need,

Fig. 117.3 The big picture of closed-loop cyber-physical system control



or a fire-fighting agent can be precisely dropped in advance of a fire's predicted path. A heterogeneous configuration of fixed- and rotary-wing aircraft in teaming scenarios can be beneficial as well; fixed-wing flying sensors can determine areas of interest, and rotary-wing craft can take more specific, detailed data or drop payloads such as ground sensor pods or radio relays for extending network communication. Ground or even underwater vehicles can act as part of this team, with robotic systems of many different kinds working together to become a cohesive cyber-physical system.

Networked systems, or “systems of systems,” will pervade the physical world in the coming years. These networks are themselves CPSs and must be engineered with the goal of long-term functionality and integration. This means that CPSs of many scales will eventually be integrated, and as in Fig. 117.4, these systems will work together for even more functionality and resilience.

117.3 Personal Remote Sensing (PRS)

Remote sensing is simply detection at a distance. Humans have a given set of perceptions granted by a suite of sensors (eyes, ears, nerves, etc.), but with science, through engineering, it is possible to significantly extend these perceptions. While information can come from many sources, the best way to collect specific information is by the use of a sensor – a device specifically constructed for the purpose of data collection. Multiple sensors may work in concert providing many sources of useful information at once; depending on the information demanded,

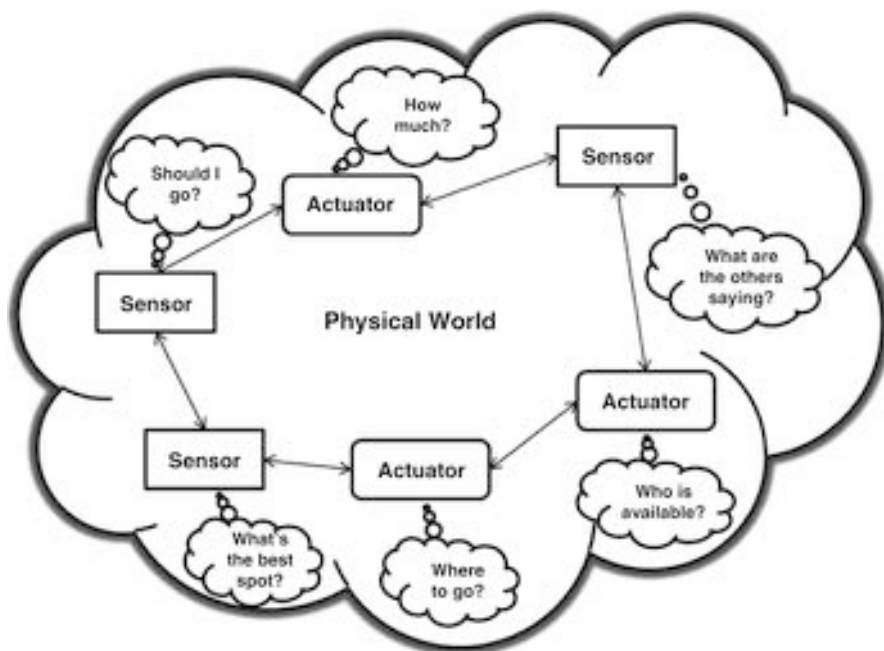


Fig. 117.4 Systems of systems: how many CPS interact to perform complex tasks

these sensors can change modes via communication to gather better data as time passes and requirements or environments change. Remote sensing allows us to gather data at a physical distance, including data for which there is no other way, or no safe way, to collect in person. In difficult-to-access or inhospitable environments, remote sensing is the only feasible method for accurate and reliable data collection.

Remote sensing, depicted in Fig. 117.5, usually connotes large-scale instruments requiring large-scale support operations (such as the Hubble telescope) and financial resources far beyond even the most serious everyday user's budget. Humanity has been attempting to extend the senses to gain advantages or additional knowledge for as long as tools have existed. Devices such as binoculars allow eyes to see further, and the hearing aid extends the range of an ear. These are examples of personal remote sensing: the detection of data useful to a small number of people – perhaps only a single person – on short timescales and within small budgets.

However, to be convenient, sensing requires aspects of autonomy: traditional systems require a large amount of user interaction for useful function. Flying, driving, or otherwise controlling a remote sensing system can be time consuming and in some cases demands high levels of skill or training for consistent quality data. An autonomous system can be preprogrammed to patrol for changes in a scene or “follow” a data trail, such as an airborne pollutant, to collect data automatically. In the case of the use of normalized difference vegetation index (NDVI) in agriculture, a farmer could use a UAS for remotely sensing the plant

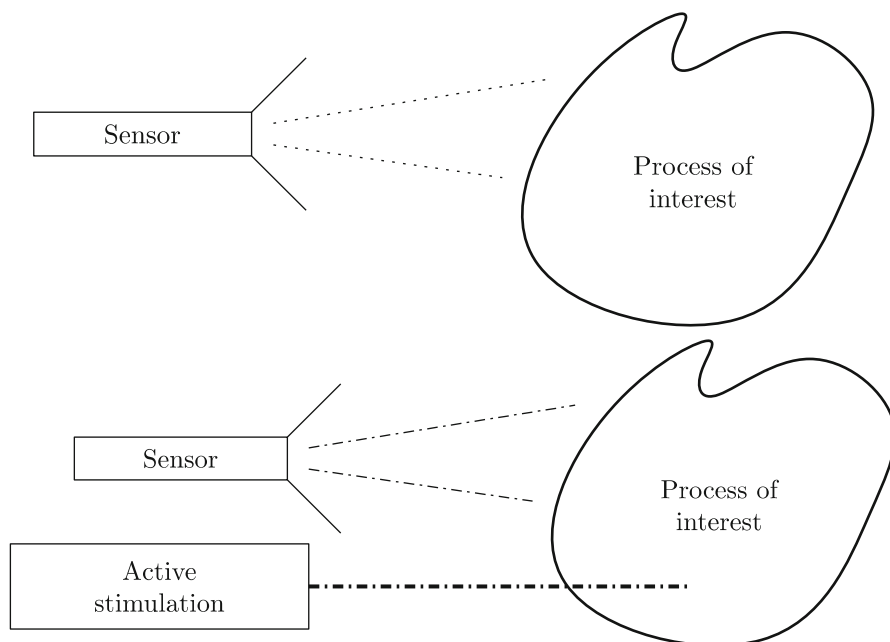


Fig. 117.5 Passive remote sensing vs. active remote sensing, which requires some kind of stimulation to reveal the desired data

growth and vigor, vegetation cover, and biomass in her fields and for determining where water, fertilizer, etc., are needed most. Such a PRS could reduce the costs of the inputs to agricultural production through greater efficiency in application and increase sustainability, yields, and profits.

The key driver for success with PRS systems is cost. Personal remote sensing can only be personal if individuals and small groups can afford to own and operate systems capable of delivering the data they need. Progression of technology (Microelectromechanical (MEMS) devices, embedded processing power, battery power density, etc.) coupled with acceptance of PRS systems by industry, regulators, and legislators, will allow PRS to become commonplace. Low-cost small UAS platforms, constructed for a little as 500 USD, have been demonstrated to be usable for remote sensing (Di and Chen 2011).

Personal remote sensing systems are poised to provide the next generation of useful data about our world. While large-scale remote sensing (like large-scale computing) will always have applications, such as weather prediction, PRS allows individuals or small groups to collect valuable data about situations or processes that concern them specifically. Personal remote sensing is an upcoming and perhaps disruptive technology which will show efficacy when combined with UASs and other autonomous systems. With PRS, new ways of interaction and feedback can be used – with scientific accuracy – to make better, more efficient decisions about what concerns us most.

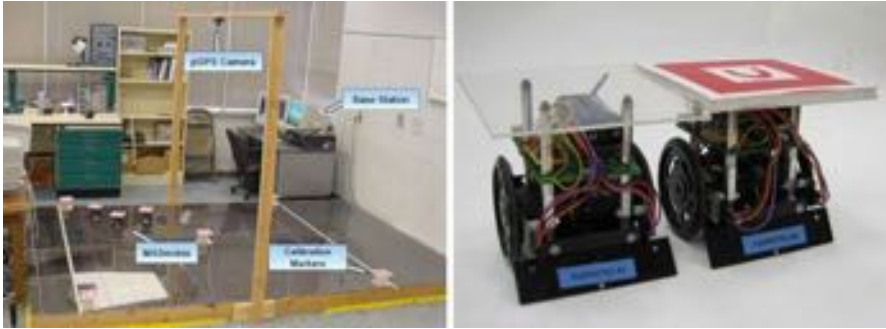


Fig. 117.6 The MASnet platform with control station and CSOIS MASmote robotic platforms

Although sensing is fundamental, once a decision about managing the world around has been made, action must be taken. Even a nonaction can be considered an active response if properly considered, and therefore, sensing is naturally coupled with actuation. Sensing and actuation cycles can be coupled in a multitude of configurations, and the use of robotic mobile platforms in addition to traditional static sensors can lead to new scenarios which have never before been conceived or implemented.

The Mobile Actuator and Sensor Network (MASnet) research platform in the Center for Self-Organizing and Intelligent Systems (CSOIS) at Utah State University combines wireless sensor networks with mobility (Bourgeois et al. 2006). In other words, the robots can serve both as actuators and sensors. Although each robot has limited sensing, computation, and communication ability, they can coordinate with each other as a team to achieve challenging cooperative control tasks such as formation keeping and environmental monitoring.

The MASnet platform is comprised of MASmotes, an overhead USB camera, and a base station PC, shown in Fig. 117.6. MASmotes are actually two-wheel differentially steered robots that can carry sensors and actuators with MicaZ from Crossbow serving as the processor. Thus, MASmotes support inter-mote communication, sensor data collecting, and pulse-width-modulated (PWM) signal generation. The overhead camera, called pseudo-GPS, is used to determine each robot's position and orientation. Images are processed on the base PC, which also serves for programming board communication and control algorithm computation. The base station communicates with a gateway mote mounted on a programming board by serial link, then the gateway mote then communicates with the MASmotes over a 2.4 GHz wireless mesh network. The base station can send commands and pseudo-GPS information to each MASmote, and all the MASmotes communicate with each other over the 2.4 GHz wireless mesh network.

Using such a hardware platform allows for various scenarios to be studied and proven beyond theory, including acquisition of data with mobile platforms such as UAS, which can be used as mobile actuators to complete the sensing

and actuation cycle. Once this basic framework has been established and proven, many applications for CPS can be implemented in a laboratory environment and experimentally verified before being taken to a much larger scale in the outside world.

Current sources of remote sensed aerial imagery (e.g., manned aircraft and satellites) are either expensive, have low spatial and/or temporal resolution, or have long turnover times. These shortcomings make it difficult to use remote sensing effectively for many applications, yet the market for these services is growing tremendously as costs fall. There are many potential applications for low-cost, effective remote sensing platforms including emergency response and disaster management, land management, resource monitoring, and civil engineering. For example, in emergency response, lives and money can be saved with the ability to send out small UAS to autonomously fly grid patterns over areas of interest for search and rescue operations or disaster zones. Additionally, in many watershed science applications, it is important to have up-to-date, aerial images of a river multiple times throughout the year at different flow levels. However, due to the price of aerial imagery from conventional sources, researchers performing work on rivers are currently limited in their ability to obtain multiple maps over their area of interest within the time frames required. In addition, since a river system can be constantly changing, if the processing turnaround time is too long, the imagery used by field crews will not be up-to-date and will be almost impossible to use. Satellite imagery can be especially difficult to work with because the time when a satellite passes overhead and local weather conditions cannot be controlled.

The AggieAirTM UAS was developed by the Center for Self-Organizing and Intelligent Systems (CSOIS), and the Utah Water Research Laboratory (UWRL) at Utah State University was designed to be the perfect SUAS remote sensing platform in response to a market demand requiring high-quality aerial imagery for agricultural, riparian, wetland, and civil engineering applications.

The AggieAir UAS platform (Fig. 117.7) is an autonomous, multispectral remote sensing platform. AggieAir makes aerial imagery more accessible and convenient by allowing the user to choose when and where the imagery is acquired by giving the user full control of the platform and through the AggieAir platform's independence of a runway. The aircraft launches using a bungee and glides to the ground for a skid landing (Fig. 117.8).

In order to assure mission success for a PRS SUAS, namely, the acquisition of data, extensive testing must take place. Payload handling and fault detection is a very important part of the AggieAir system. Since consumer-level cameras and other sensors are not constructed to a relatively high-quality level, allowances must be made in payload control software architecture. Figure 117.9 shows this software in detail and provides justification for many critical choices in payload design for scientific data collection (Coopmans et al. 2012). The performance of this system can be verified through extensive testing; especially useful is a hardware-in-the-loop simulation testing capability, providing simulated flight data into the payload system for confirmation of performance and reliability over any specific mission.



Fig. 117.7 AggieAir SUAS system diagram (Haiyang Chao 2009)



Fig. 117.8 AggieAir bungee launch and skid landing

During flight, images are captured every 4 s along with data describing the GPS position and aircraft orientation of the UAS at time of exposure of each image. After the UAS has completed a data collection mission and successfully landed, camera imagery and flight information are downloaded from the payload system. The images and their corresponding position and orientation information can be quickly, approximately, georeferenced to generate a map within 30 min after flight (Jensen 2009). However, due to the errors in position and orientation estimation by the aircraft sensors, the images are not perfectly georeferenced. Figure 117.10a shows a set of images that were taken over a river and directly georeferenced. Notice that they appear approximately where they should, but do not align well with each other.

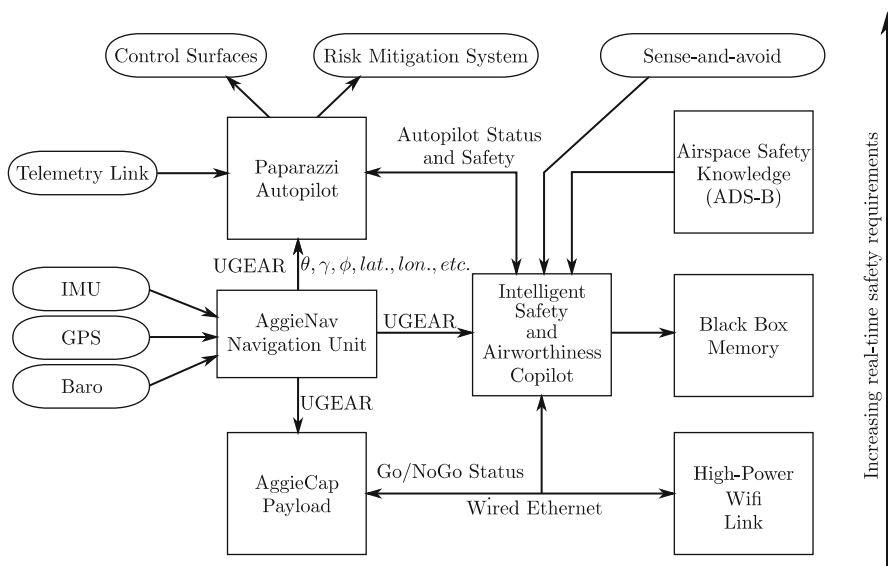


Fig. 117.9 The AggieCap software architecture

While directly georeferenced images might be sufficient for some applications, others need more accurate results. For this EnsoMOSAIC UAV is used. EnsoMosaic UAV, developed by Mosaic Mill of Finland, is an image processing software that reads aerial images captured with compact digital cameras and stitches them into seamless orthorectified image mosaics (MosaicMill EnsoMOSAIC 2012). Figure 117.10b shows an orthorectified mosaic made from the imagery displayed in Fig. 117.10a (generated by post-processing) was prepared and available only 72 h after the data collection flight.

117.4 CPS Enabled by SUAS: UAS-Based Real-Time Irrigation Control (WaterWatch)

Aerial remote sensing can be used to better regulate and control water use by optimally distributing irrigation supplies on the basis of predicted demand.

As seen in Fig. 117.11, a water domain is depicted as a closed-loop control system. Starting with weather and climate, the water becomes available. Combined with water rights, an optimal irrigation policy can be established (Tricaud and Chen 2011; Chao et al. 2008) that will allow the gates and flumes which control water flow to direct the water to its best use. While some water is lost to evaporation and leakage, most will reach the geo-domain of interest, allowing use of the water.

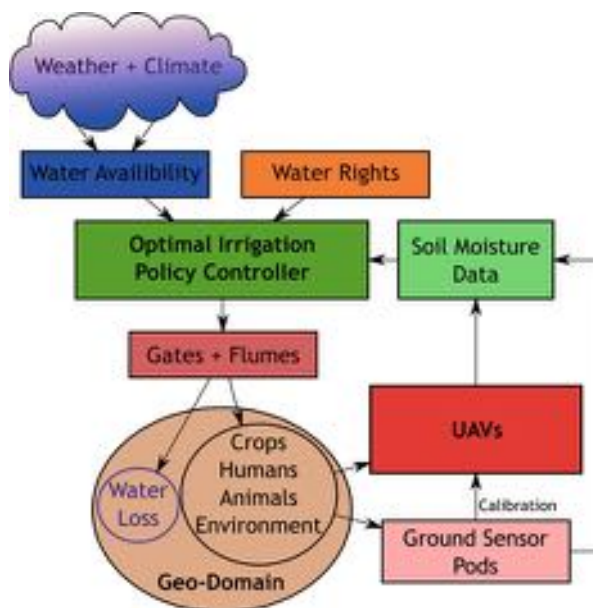


Fig. 117.10 Data collected from AggieAir before (a) and after post-processing (b)

During an example use of the water (a farmer's field), traditional ground sensor pods will determine the moisture level in the application areas, as well as UAS-borne sensors flying above the field. Aerial imagery collected in the visible, near-infrared, and thermal bands can be combined to determine the water content of the vegetation and the surface soils in the scene (Bajwa and Vories 2006), and data fusion can combine the ground and aerial sensor data into the controller, thereby closing the water loop on the particular field in question.

Use of multispectral sensors can provide deep insight into moisture content as seen in Fig. 117.12. Thermal infrared, shortwave infrared, and near-visible infrared, as well as visible light sensors, are needed to complete the picture and provide detailed moisture maps, allowing more data about the distribution of water and enabling greater control or management.

Fig. 117.11 Water distribution and use as a closed-loop control problem



117.5 CPS Enabled by SUAS: Algal Bloom Tracking for Alternative Energy

According to the NSF, “The United States faces a critical challenge to transform our current fossil fuel based energy economy to a stable and sustainable energy economy” (NSF 2012).

Sustainable Energy Pathways, SEP, is a new research program by the NSF, to encourage and “support interdisciplinary efforts by teams of researchers to address the challenges of developing efficient pathways towards a sustainable energy future. *The overarching theme of the solicitation is “sustainability” in all of its facets*” (NSF 2011).

In one example of sustainability, algae grow in wastewater lagoons (such as Logan, Utah City, sewage lagoons, shown in Fig. 117.13) and feed on nutrients from detergents, etc. These algae can be processed and transformed into energy: both biofuels and methane can be produced from algae, allowing small communities to transform local waste into power. Once the algae have been harvested, it is consumed in a bioreactor such as microbial fuel cells as detailed in Dye et al. (2010).

The algae harvesting cycle:

1. Algae grow in a controlled environment (lagoon) fed by wastewater.
2. Algae “bloom,” or rapidly increase, its biomass.
3. When sufficient biomass is available, the algae are harvested, providing raw materials for energy production.
4. The algae lagoon begins growing another batch of algae.



Fig. 117.12 Multispectral data collected from an agricultural scene: (a) Visible and NIR data before flood. (b) Visible and NIR data after flood

To maximize biomass in Step 3 above, feedback is needed to determine when to harvest the algae to maximize production and minimize production times. This feedback can be provided by a UAS in a PRS configuration, flying at an appropriate frequency to estimate and predict the peak algae biomass.

One perspective configuration for using a PRS UAS as a CPS to establish a SEP is seen above in Fig. 117.14. Stations like this one allow production of algae-based power by drawing on wastewater from a local community, keeping the energy pathway short and allowing the highest efficiency. When coupled with other sustainable energy sources such as solar power, the efficiency and autonomy is much higher, allowing for better localization of energy resources.

While algae grow in the phosphorus-rich wastewater of the lagoon, the algae's natural predator, *Daphnia*, presents a threat to the water treatment and energy production processes. When *Daphnia* are present in the water, the algae are only able to decrease the phosphorus by 1/40th the amount if they were alone



Fig. 117.13 A view of the Logan Lagoons from Google Earth (2012)

(Chea et al. 2010). This presents a compelling problem: harvest the algae *before* the *Daphnia* do, when the algae are at peak saturation.

UASs are used to survey the algae and detect the optimum harvesting time for the algae crop. Once this is determined, the algae are harvested by some automated means, and the reactor uses the algae as fuel to produce power. The solar grid shown is mainly for operation of the algae farm; however, if there is a power surplus, it can be directed out as product energy.

As seen in Fig. 117.13, the Logan Lagoons support 8 local districts and at 460 acres and are ideally sized for small UAS data collection. Relative to traditional static (i.e., non-mobile) pod sensors embedded in the ponds themselves, UASs cost less.

117.6 CPS Enabled by SUAS: Invasive Species Management

Accurately assessing changes in wetland vegetation over time is important for both natural resource management and ecological research. A fundamental question in ecology is what drives the distribution of plant species; addressing this question implies that we are able to accurately determine where certain species occur and how their occurrence changes over time. At the same time, natural resource managers need to be able to determine where desirable native plant species occur and how

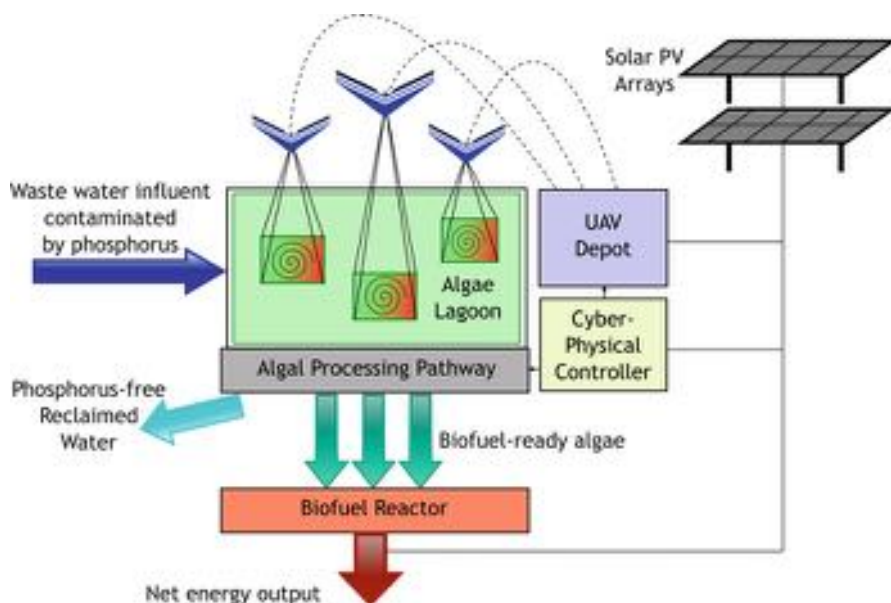


Fig. 117.14 Algae land: A UAS-enabled CPS for SEP. A water treatment lagoon as a UAS-enabled CPS to create biofuel from wastewater

management activities drive changes in vegetation. One of the biggest challenges for natural resource managers is whether they can accurately track changes in invasive plant species, either their expansion or their retraction, in response to control efforts. Many currently available remote sensing strategies do not operate at fine enough resolutions in space or time to be useful for these ecological and management purposes. Satellite imagery lacks sufficient resolution in both space and time to provide decision-relevant information to wetlands managers. Imagery obtained from the use of conventional aircraft platforms is too expensive for many wetlands management applications. In contrast, high-resolution imagery can be obtained from the application of unmanned aerial vehicles at very low cost in several different spectral bands.

Utah's wetlands are among the most valuable habitats in the state, yet the expansion of invasive species presents an ongoing threat to their productivity in terms of their value for habitat to birds, fish, and wildlife.

Research has been conducted at the Bear River Migratory Bird Refuge (BRMBR) which is located on the northeast shore of the Great Salt Lake, Utah, at the terminus of the Bear River. The Refuge is managed by the U.S. Fish and Wildlife Service as part of the National Wildlife Refuge System; comprises over 115 square miles of marsh, open water, uplands, and alkali mudflats; and is one of the largest wetland complexes along the Great Salt Lake. With this location and size, the Refuge provides critical wetlands wildlife habitat and resting grounds for migratory

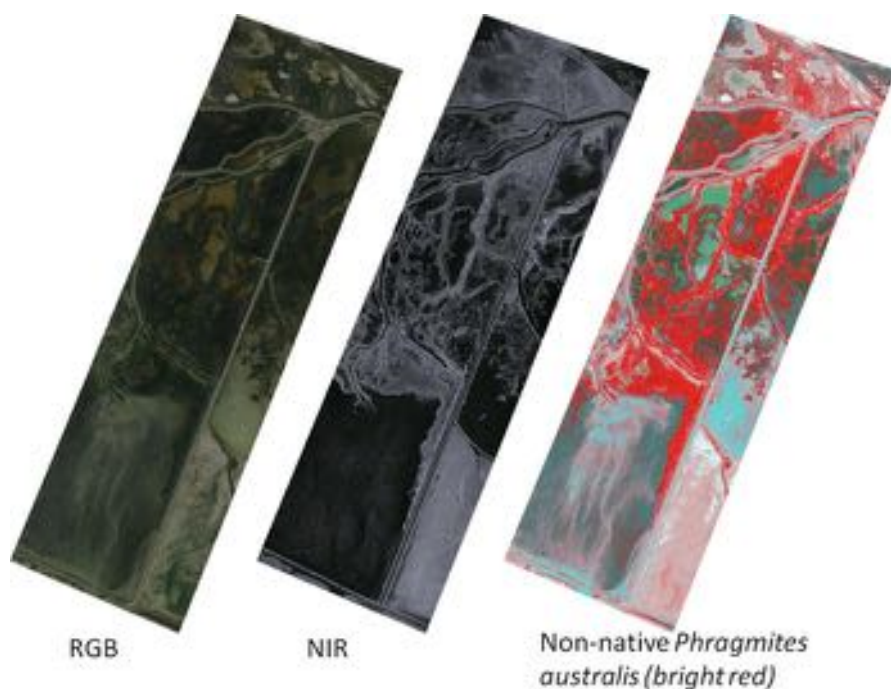


Fig. 117.15 Processed and classified UAS imagery for a 4-square-mile area at the BRMBR, Utah

birds along the Pacific Flyway. It is one of the most important habitat areas for migratory birds in North America.

In the work done by researchers at the Utah Water Research Laboratory (UWRL), aerial imagery in the red, green, blue, and near-infrared spectra was obtained for approximately 50 square miles of the BRMBR and used to produce a high-resolution base map of the entire area wherein *Phragmites* is a problem. These data were used in combination with on-ground field inspections to identify specific areas or patches, totaling approximately 12 square miles, where AggieAir UASs were used to conduct aerial sampling at three other points in time. The imagery that was acquired was used to identify the distribution of *Phragmites* within that 12-square-mile area and to measure the spread of *Phragmites* throughout the year. Each of these four high-elevation UAS flights (i.e., one 50-square-mile flight and three flights of 12 square miles each) produced mosaicked and ortho rectified images that have a resolution of approximately 25 cm (see Fig. 117.15a, b).

Recognition of wetland vegetation using remote sensing is very difficult (Yamagata and Yasuoka 1993). A machine learning strategy which uses remotely sensed reflectance data to classify the wetland vegetation into different categories was used to identify *Phragmites* at high accuracy. The approach is based on a multiclass relevance vector machine (MCRVM) that has been successfully used for multiclass land cover classification and crop identification with good results

(Zaman and McKee 2012). The machine has been shown to have good generalization capability. A MCRVM model was trained with the remotely sensed vegetation reflectance data acquired over the BRMBR and the data obtained from on-ground sampling. The MCRVM model was capable of classifying previously unseen data into different categories, including *Phragmites* and other vegetation types growing in the BRMBR. An example of the results of this classification application is shown in Fig. 117.15c, where the location of *Phragmites* (in bright red) has been determined with an accuracy of approximately 95 % (Zaman et al. 2011). After the classification results are obtained, GIS and ERDAS Imagine software is used to estimate the area occupied by different species, or mixes of species, in the wetland.

117.7 CPS Enabled by SUAS: Tagged Fish Tracking and Wildlife Management

In order to preserve our native fish species, managing, monitoring, and studying fish habitat is very important. Extensive fish habitat studies are conducted to minimize environmental impact before building a dam, while restoring natural habitat in a river, or to ensure economic strength of a fisheries market.

An important part of these habitat studies is tracking where these fish are migrating, moving, and living. This is commonly done by tagging a fish with a radio transmitter as seen in Fig. 117.16 and using it to locate the fish after some period of time to see to where it has moved (Hillyard et al. 2009; Gray and Haynes 1979). Locating a tagged fish is done using a receiver and a directional antenna. As the antenna is pointed in the direction of the fish, the beacon sent from the transmitter becomes stronger, and the operator navigates toward the fish. While this method has proved effective, it is time consuming and costly.

An alternative method to track radio-tagged fish is to use the AggieAir SUAS, for which a fish-tracking payload has been developed. With this payload, AggieAir can be used to fly over a body of water and receive the beacon transmitted by a tagged fish, measuring the signal strength and logging other data including position, orientation, and speed of the SUAS. For fish-tracking applications in rivers, one SUAS could be sent along the river to collect data. After the flight, the data can be post-processed into a map that shows the estimated locations of all tagged fish and the variance of the estimations. While this simple solution could work well for a one-dimensional feature like a river, it may not work as well for two-dimensional features such as lakes. A better solution for tracking fish or other animals in a two-dimensional scenario would be using multiple SUASs in formation. Multiple SUASs could better estimate where a tagged fish is and update their flight plans to get better approximations.

In a general radio tracking problem, Frew et al. (2005) proposed a radio localization technique cast as a distributed estimation problem. Consensus within a group of SUASs was reached with a decentralized method; however, the algorithms used are computationally complex and only considered communication with one neighboring SUAS. Korner (Korner et al. 2010) introduced a method using particle

Fig. 117.16 Implantable radiotransponder tags, allowing remote detection of fish of various sizes (From Lotek corporation (2012))



filters to track multiple targets with radio tags, but only one UAS was used. While these methods work well, a simple, decentralized method is needed to work in practice.

117.8 Other Ideas for UAS Sensing-Enabled CPSs

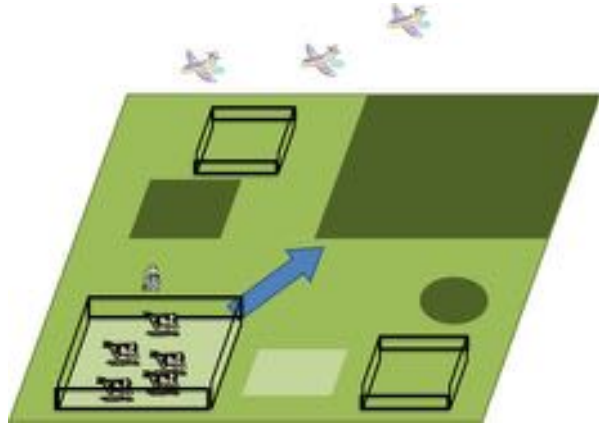
Since the markets for UAS-based remote sensing, personal remote sensing, and civilian unmanned systems in general are under developed, many more ideas, which are ultimately limited by our imagination, can be realized with the AggieAir personal remote sensing platform. Some of these stimulating ideas are presented briefly in this section.

(a) *Optimal Animal Grazing and Herding*

In his book *The Omnivore's Dilemma* (Pollan 2006), Micheal Pollan presents some thought-provoking dilemmas found in our food industry. Further along this line of thinking (relating to scarcity of resources), how will we survive if chemical fertilizer production is no longer possible? One chemical fertilizer-free solution is to revert to more ancient ways of food production, by using the manure produced by animal production to fertilize farmland in a more optimal way.

Figure 117.17 shows a CPS where grazing animals are moved from plot to plot during farming. Simultaneously, UASs are used as mobile sensors to sense water and nitrogen content of the soil. The goal is to make the grassland uniformly grow in a self-sustainable manner, which is only possible if UASs are

Fig. 117.17 Movement of animals during grazing can optimize nutrients in a field



employed as remote sensors in such a large area. Many research questions arise: How many animals per acre? How often are animals' locations changed? How often is aerial data collected? Most importantly, what is the optimal location for the animals at a given time? Perhaps the health of the animals can also be considered in the CPS, allowing for more healthy livestock as well as better land use.

(b) *Multi-UASs for Wind Field Mapping, Pollution, and Plume Characterization*

A UAS could serve as a new measurement platform for the surface wind field because UASs could be easily commanded to any 3D point in the Atmospheric Boundary Layer, which is a difficult mission for meteorological stations. Common microsensors such as temperature, humidity, pressure, and wind speed are now very easily mounted on UASs. Recently, it has become possible to employ multiple small UASs to form a team to measure the horizontal or vertical profile simultaneously. Such a configuration of UASs working in coordination on a common data collection assignment is called a "coven." This capability can offer significant advantages for detection of wind gradients.

The prevailing wind is also called longitudinal wind, which is the wind prediction estimated from larger-scale synoptic data. It is the prevailing direction at the predicted height without considering the surface frictions and complex terrains (see Fig. 117.18). The technique suggested is to send the UAS to measure the wind speed and estimate the noise level at the same time.

The formation shape (Fig. 117.19) could also be changed based on different resolution requirements. For example, a closer formation is needed for areas with turbulence while a formation of larger separation is chosen for prevailing wind dominant areas. The UAS distance could also be determined by the resolution requirements.

(c) *Wildfire and Disaster Emergency Response*

Wildfires are common in much of the Western USA. Many brave men and women give their lives to fight these fires and many millions of dollars are spent controlling the spread of fire and protecting the lives and property of the public.

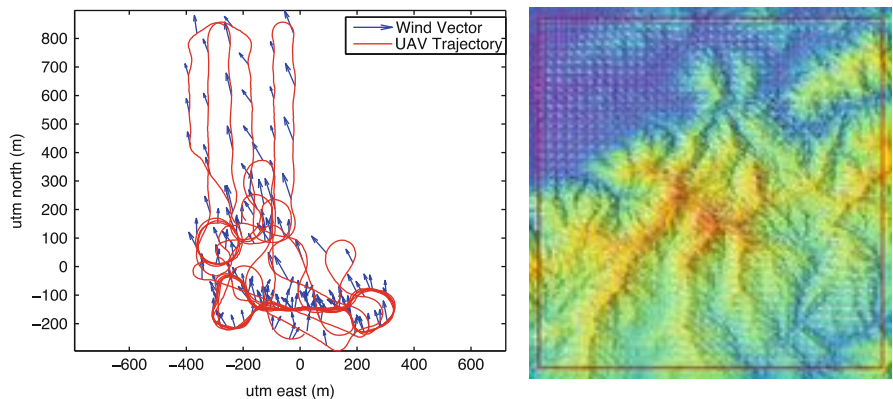


Fig. 117.18 Measured and estimated wind data from UAS flight (Chao and Chen 2012)

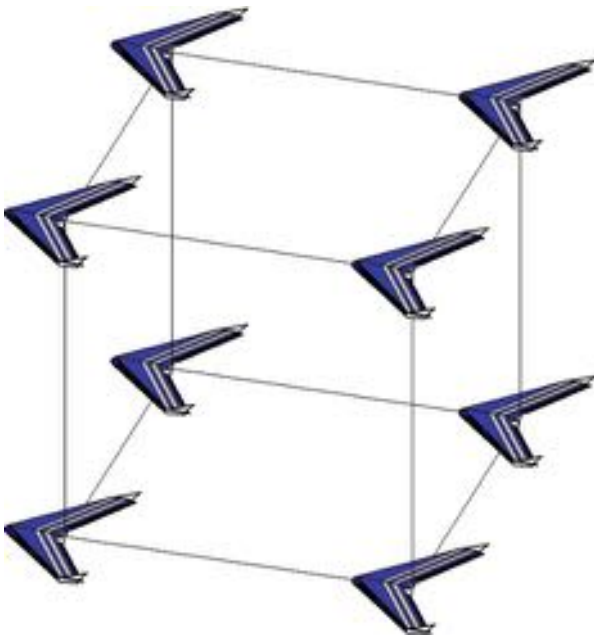


Fig. 117.19 A coven of UASs in a cube formation for better characterization of wind field

Figure 117.20 shows a CPS setup to control a wildfire. The UASs pictured on the left includes wind measurement sensors, as well as thermal mapping sensors. While the UASs provide real-time data about the fire, a base station computes the most probable path of the fire and automatically dispatches a larger UAS (to the right of Fig. 117.20) with fire retardant (water, etc.) to preemptively lessen the fire’s destructive power by dropping payload in the most useful location.

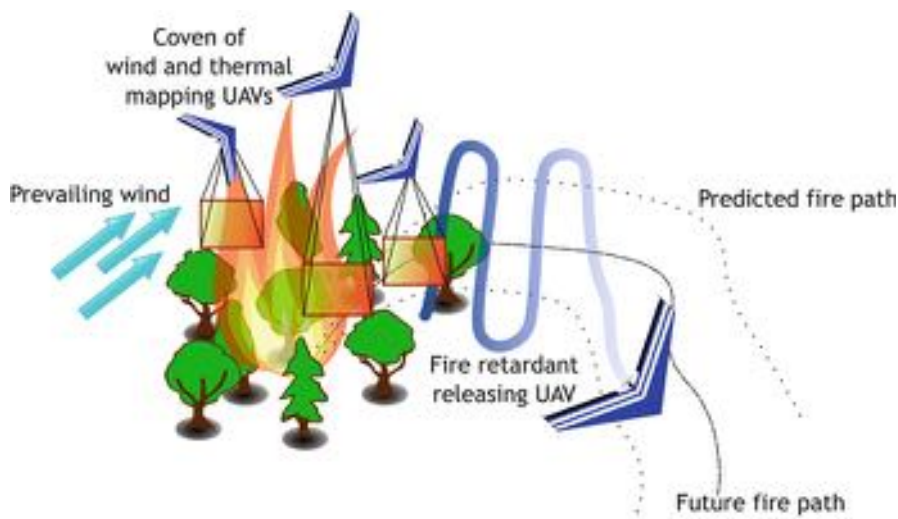


Fig. 117.20 Several UASs work together to control a wildfire with sensors and retardant

The prediction of the fire's path can also be used to warn the surrounding populace about the impending arrival of the fire automatically, allowing the residents the most possible time to evacuate before the fire arrives.

(d) *Multi-UAS and Nuclear Radiation Level Characterization and Tracking*

Multiple UASs can work together in a nuclear disaster scenario to profile and help predict the distribution of radiation after an unintended release of radioactive materials. This will help the surrounding populace react to the leak by avoiding the worst of the radiation and allow rescue efforts to be prioritized by suggesting safe zones, while helping manage the radiation exposure of the rescue crews.

The UASs involved can be equipped with total radiation dose (TRD) sensors, allowing a UAS to cognitively determine if it has been exposed to too much radiation and whether it must be decommissioned at a predetermined safe disposal site. In this way, aircraft are used to their fullest potential and any humans involved in a nuclear disaster response scenario are optimally kept from harm's way. Figure 117.21 shows AggieAir small UASs performing cognitive cooperative tasks such as formation flight for sensing.

117.9 Conclusions and UAS-Enabled CPS Outlook

The future of UASs is undoubtedly bright. While the current public perception of UASs is one of espionage and warfare, they will become more accepted into domestic use as their potential value becomes apparent and as the airspace rules change to include them. While current regulations of UASs are restrictive and

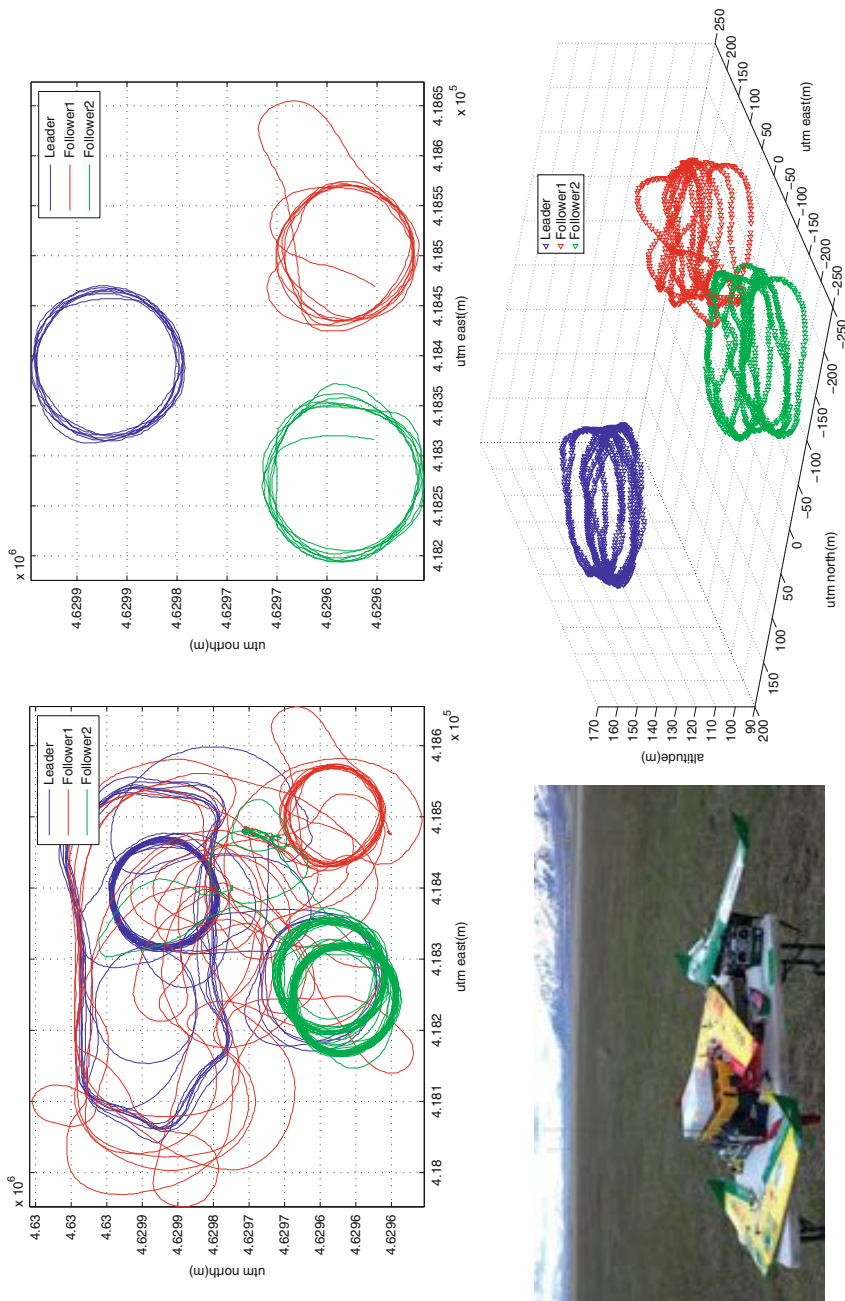


Fig. 117.21 Three UAS formation flight under cognitive autopilots

limited in the USA, soon UASs will become available for regular use as standards for certification and airworthiness are developed.

UAS applications such as search and rescue and precision agriculture are now well known, and more applications appear nearly every day. Small UASs in particular are expected to play a major role in domestic UAS operations. Versatile, inexpensive, and easily maintainable, these smaller systems can be utilized effectively in many applications where fast responses are needed with precision information. Further applications of these systems will lead to their utilization as part of larger cyber-physical systems as sensors or even actuators, providing integration and control on larger scales and higher complexities.

Cyber-physical systems are the future of solving real-world problems such as water management and alternative energy production. By using SUASs as sensors, better data can be collected and used to make informed control decisions, transforming difficult, complex, or abstract problems into closed-loop systems that can be analyzed and controlled.

In the next 15 years, the development of unmanned technology will enable UASs to be mobile actuators, actively exerting control in optimal locations for precision in both sensing and actuation and cognitive control systems for large-scale complex systems that were previously uncontrollable or impractical to control. Management of crisis situations such as food and water shortages, energy production, floods, and nuclear disasters can be assisted by SUASs. Difficult problems can be solved with cyber-physical systems. This chapter serves as a motivator to position small, inexpensive UASs as flying sensors and actuators within the closed loops of cyber-physical control.

References

- S.G. Bajwa, E.D. Vories, Spectral response of cotton canopy to water stress, in *Proceedings of the ASAE Annual Meeting*, Portland, 2006. Retrieved from <http://asae.frymulti.com/abstract.asp?aid=21090&t=1>
- J.A.J. Berni, P.J. Zarco-Tejada, L. Suarez, E. Fereres, Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *IEEE Trans. Geosci. Remote Sens.* **47**(3), 722–738 (2009). doi:10.1109/TGRS.2008.2010457
- W. Bourgeois, L. Ma, P. Chen, Z. Song, Y.Q. Chen, Simple and efficient extrinsic camera calibration based on a rational model, in *2006 International Conference on Mechatronics and Automation*, Luoyang (IEEE, 2006), pp. 177–182. doi:10.1109/ICMA.2006.257492
- H. Chao, Y.Q. Chen, *Cooperative Remote Sensing and Actuation Using Networked Unmanned Vehicles* (Wiley-IEEE, Hoboken, 2012), p. 232
- H. Chao, M. Baumann, A. Jensen, Y. Chen, Y. Cao, W. Ren, M. McKee, Band-reconfigurable multi-UAV-based cooperative remote sensing for real-time water management and distributed irrigation control, in *Proceedings of the IFAC World Congress*, Seoul, 2008. Retrieved from <http://www.ifac-papersonline.net/Detailed/37674.html>
- H. Chao, A.M. Jensen, Y. Han, Y. Chen, M. McKee, in *Advances in Geoscience and Remote Sensing*, ed. by G. Jedlovec (InTech, Vukovar, 2009). doi:10.5772/955
- N.D. Chea, K.S. McCulloch, J.A. Powell, R.C. Sims, Daphnia-Algae modeling of the Logan Wastewater Lagoons. Utah State University Biological Engineering Department, IBE Poster, 2010

- C. Coopmans, B. Stark, C.M. Coffin, A payload verification and management framework for small UAV-based personal remote sensing systems, in *2012 5th International Symposium on Resilient Control Systems*, Salt Lake City (IEEE, 2012), pp. 184–189. doi:10.1109/ISRCSS.2012.6309316
- L. Di, Y.Q. Chen, Autonomous flying under 500 USD based on RC aircraft, in *Volume 3: 2011 ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications, Parts A and B*, Washington, D.C., vol. 2011 (ASME, 2011), pp. 929–936. doi:10.1115/DETC2011-47851
- D. Dye, R. Sims, I. Hamud, R. Thompson, E. Griffith, Algae bioremediation and biofuels at the Logan, Utah wastewater treatment facility. Utah State University Sustainable Waste-to-Bioproducs Engineering Center, Poster, 2010
- E.W. Frew, C. Dixon, B. Argrow, T. Brown, Radio source localization by a cooperating UAV Team, in *Infotech@Aerospace*, Arlington, 2005, pp. 1–11. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.85.1416>
- Goodrich ISR Systems (2012), Retrieved 30 Sept 2012, from www.sensorsinc.com.
- Google Earth (2012), Retrieved 30 Sept 2012, from <http://earth.google.com>.
- R.H. Gray, J.M. Haynes, Spawning migration of adult chinook salmon (*Oncorhynchus tshawytscha*) carrying external and internal radio transmitters. *J. Fish. Res. Board Can.* **36**(9), 1060–1064 (1979). doi:10.1139/f79-148
- R.W. Hillyard, E.R. Keeley, R. Ernest, Distribution and spawning migrations of fluvial Bonneville cutthroat trout in the Bear river, Idaho, Department of Biological Sciences, Idaho State University, 2009. http://www.pacificorp.com/content/dam/pacificorp/doc/Energy_Sources/Hydro/Hydro.Licensing/Bear_River/2009.1_BCT-Telemetry_Study.pdf
- J. Hunt, W.D. Hively, S.J. Fujikawa, D.S. Linden, C.S.T. Daughtry, G.W. McCarty, Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sens.* **2**(1), 290–305 (2010). doi:10.3390/rs2010290
- Infrared Cameras Incorporated (2012), Retrieved 30 Sept 2012, from <http://www.infraredcamerasinc.com>.
- A. Jensen, gRAID: a geospatial real-time aerial image display for a low-cost autonomous multispectral remote sensing platform (AggieAir). Utah State University, 2009. Retrieved from <http://proquest.umi.com/pqdlink?Ver=1&Exp=01-04-2016&FMT=7&DID=1949509671&RQT=309&attempt=1&cfc=1>
- A.M. Jensen, T. Hardy, M. McKee, Y.Q. Chen, Using a multispectral autonomous unmanned aerial remote sensing platform (AggieAir) for riparian and wetland applications, in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Vancouver, 2011
- Z. Jiao, Y. Chen, I. Podlubny (eds.), Introduction, in *Distributed-Order Dynamic Systems* (Springer, London, 2012), pp. 1–10
- F. Korner, R. Speck, A.H. Goktogan, S. Sukkarieh, Autonomous airborne wildlife tracking using radio signal strength, in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei (IEEE, 2010), pp. 107–112. doi:10.1109/IROS.2010.5654385
- Lotek Corporation (2012), Retrieved 30 Sept 2012, from <http://www.lotek.com>
- MosaicMill EnsoMOSAIC (2012), Retrieved 30 Sept 2012, from <http://www.ensomosaic.com>
- NSF, Sustainable Energy Pathways (SEP) program solicitation (2011), Retrieved 30 Sept 2012, from <http://www.nsf.gov/pubs/2011/nsf11590/nsf11590.htm>
- NSF, Cyber-physical systems program solicitation (2012), Retrieved 12 Sept 2012, from <http://www.nsf.gov/pubs/2011/nsf11516/nsf11516.htm>
- P. Oh, Forward, in *Remote Sensing and Actuation Using Unmanned Vehicles*, ed. by H. Chao, Y. Chen (Wiley, Hoboken, 2012), pp. xxi–xxii
- M. Pollan, *The Omnivore's Dilemma: A Natural History of Four Meals* (Penguin Group, New York, 2006), p. 450. Retrieved from <http://books.google.com/books?hl=en&lr=&id=Qh7dkdVsbDkC&pgis=1>
- C. Tricaud, Y. Chen, *Optimal Mobile Sensing and Actuation Policies in Cyber-Physical Systems* (Springer, New York, 2011), p. 170

- Y. Yamagata, Y. Yasuoka, Classification of wetland vegetation by texture analysis methods using ERS-1 and JERS-1 images, in *Proceedings of IGARSS'93 – IEEE International Geoscience and Remote Sensing Symposium*, Tokyo (IEEE, 1993), pp. 1614–1616. doi:10.1109/IGARSS.1993.322073
- B. Zaman, M. McKee, C.M.U. Neale, Fusion of remotely sensed data for soil moisture estimation using relevance vector and support vector machines. *Int. J. Remote Sens.* **33**(20), 6516–6552 (2012). doi:10.1080/01431161.2012.690540
- B. Zaman, M. McKee, A. Jensen, Use of high-resolution multispectral imagery acquired with an autonomous unmanned aerial vehicle to quantify the spread of an invasive wetland species, in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Vancouver, 2011

Section XXI

Social and Ethical Implications

George R. Lucas Jr.

Kimon P. Valavanis and George J. Vachtsevanos

Social and Ethical Implications of the Handbook aims at answering the question “*What are the social, legal, and ethical concerns associated with the use of UAVs?*”

► [Ethics and UAVs](#) by Lucas provide an overview of ethical, legal, and “robot morality” challenges arising from the military and security uses of robotics and other unmanned vehicles in a number of relevant conflict environments, including the increased domestic uses of both remotely controlled and fully autonomous unmanned aerial, maritime, and ground systems for immigration control, border surveillance, drug interdiction, and domestic law enforcement.

► [The International Governance of Autonomous Military Robots](#) by Marchant, Allenby, Arkin, Borenstein, Gaudet, Kittrie, Lin, Lucas, O’Meara, and Silberman aims at providing a background of some of the principal legal issues that may motivate and initiate a much-needed legal and ethical dialogue related to the use of lethal autonomous robotic technologies in the military context.

► [The Moral Case Against Autonomous and SemiAutonomous UAVs](#) by Sharkey focuses on ethical concerns about the application of armed robots in areas with mixed combatant and civilian populations, making the claim that one must scrutinize the mapping between the applications of the new technology and the current laws to ensure that those laws are preserved and followed.

► [The Moral Case for Autonomy in Unmanned Systems](#) by Arkin centers on the need to restrain the technology itself above and beyond the human limits of

K.P. Valavanis (✉)

John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO, USA

e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos

Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of Technology, Atlanta, GA, USA

e-mail: gjv@ece.gatech.edu

the warfighters themselves, making the case for the use of ethical autonomy in unmanned systems, which is deemed necessary since warfare will continue and autonomous robots will ultimately be deployed in its conduct. In this way, one may hope that battlefield atrocities will not become progressively worse given the progression of standoff weapons and increasing use of technology.

► [Moral Predators: The Duty to Employ Uninhabited Aerial Vehicles](#) by Strawser focuses on the supposed objections to the use of UAVs in principle, defending the view that society has a duty to protect an agent engaged in a justified act from harm to the greatest extent possible so long as that protection does not interfere with the agent's ability to act justly. Since UAVs do afford such protection, an argument is made that a nation is obligated to employ UAV weapon systems if it can be shown that their use does not significantly reduce a warfighter's operational capability.

► [Killer Robots: Ethical Issues in the Design of Unmanned Systems for Military Applications](#) by Sparrow argues that designers of unmanned systems used for military applications must consider ethical, operational requirements and limits of such systems. This chapter identifies and discusses a number of issues and offers some analysis of the implications of each issue and how it might be addressed.

The concluding chapter in this section, ► [Ethical Issues of UAVs for National Academy of Sciences](#) contains a transcript of testimony presented by Lucas to a select ad hoc committee convened by the National Academy of Sciences (USA) in response to a request from the U.S. Defense Advanced Research Projects Agency (DARPA) to evaluate the "Ethical and Societal Implications of Advances in Militarily Significant Technologies that are Rapidly Changing and Increasingly Globally Accessible." This testimony was delivered at the inaugural meeting of the committee on 31 August 2011 at the NAS Beckman Center, on the campus of the University of California – Irvine.

Although collectively this section presents diverse but justifiable points of view, we are far from reaching closure on these issues, which will often be revisited as the technology progresses.

George R. Lucas Jr.

Contents

119.1 Remotely Piloted Systems 2867

119.2 Autonomous UAVs 2868

119.3 Autonomous UAVs and Force Multiplication 2869

119.4 Ethics and Engineering 2872

119.5 Conclusion 2874

References 2876

Abstract

This introduction provides an overview of ethical and legal challenges arising from the military and security uses of robotics and “unmanned” or “uninhabited” (and sometimes “remotely piloted”) vehicles in a number of relevant conflict environments. The discussion focuses primarily upon the use of autonomous unmanned platforms in combat and low-intensity international conflict but also offers guidance for the increased domestic uses of both remotely controlled and fully autonomous unmanned aerial, maritime, and ground systems for immigration control, border surveillance, drug interdiction, and domestic law enforcement. An outline of the emerging debate concerning “robot morality” and computational models of moral cognition and an examination of the implications of this debate for the future reliability, safety, and effectiveness of autonomous systems (whether weaponized or unarmed) that might come to be deployed in both domestic and international conflict situations are also included.

G.R. Lucas Jr.
United States Naval Academy, Annapolis, MD, USA
Naval Postgraduate School, Monterey, CA, USA
e-mail: grlucas@nps.edu

The legal and ethical controversy surrounding the use of armed UAVs in military operation began in earnest with the publication of an article entitled “Killer Robots,” by Australian philosopher, Robert Sparrow (2007), arguing that the development of autonomous lethal UAVs in particular would constitute a violation of international law of armed conflict, inasmuch as no predictable or meaningful notion of accountability for “war crimes” could be connected with their use. Subsequently, the Hon. Philip Alston, UN Special Rapporteur for Extrajudicial Executions, publicly denounced the use of UAVs by the USA in particular to carry out “targeted killings” of enemy insurgents within the sovereign borders of nations with which it was not formally at war. These developments and the social and political factors leading to an ever-increasing reliance of conventional military forces on the use of unmanned systems of all varieties was outlined dramatically in a widely read work, *Wired for War*, by Brookings Institution scholar Peter W. Singer (2009).

The case for and against the use of remotely piloted “drones” and the risks associated with proposals to equip subsequent systems with the ability for independent, non-supervised operations (“autonomy”) was thoroughly outlined in a lengthy report funded by the U.S. Office of Naval Research (Lin et al. 2008), while a comprehensive review of the existing legal regime in domestic and international law, coupled with proposals for its clarification and improvement, was undertaken by an interdisciplinary and interinstitutional consortium of legal scholars, engineers, scientists, and philosophers (Marchant et al. 2011). Meanwhile, two of the world’s leading authorities on robotics and computer science cited in Singer’s book subsequently began a series of debates in public appearances and in print, on the one hand defending efforts to develop lethal autonomous systems (e.g., Arkin 2010), while on the other hand roundly condemning such operations, not only as illegal under current international law but as profoundly reckless, dangerous, and irresponsible (e.g., Sharkey 2010).

Finally, a former U.S. Air Force officer and ethics instructor published a widely read article proposing what he termed “the Principle of Unnecessary Risk” (PUR), according to which responsible governments owed their war fighters and security forces every advantage and precaution that could be provided them in the pursuit of their otherwise legal and morally justifiable operations in behalf of national defense and security or law enforcement (Strawser 2010). This meant, according to the author, that the current use of remotely piloted systems, at very least, was not only legally sound but morally obligatory on the part of responsible governments, *provided* that the military and security operations in which they were employed were, in all other respects, firmly within the bounds of law and morality.

The use of “manned” or “remotely piloted” UAVs in pursuit of a nation’s enemies across sovereign national borders, or by nonmilitary personnel carrying out targeted assassinations, continues to provoke legal and moral controversy, while industrial and engineering efforts to develop systems that would operate largely without human oversight or supervision present enormous new legal and moral challenges, particularly for the engineers and defense industries in the logistical procurement and supply chain involved in the design, development, manufacture, and eventual deployment of such systems. A select and representative sample of the authors

and contrasting positions outlined above is presented in this section, while a more detailed survey and analysis of the larger debate, focused particularly upon those issues surrounding engineering design and eventual military use, together with a complete list of references, is provided below.

119.1 Remotely Piloted Systems

A view prevalent among defense industry engineers and U.S. Department of Defense (DoD) policymakers is that, in their present nonautonomous status, unmanned or remotely piloted systems raise no genuinely new legal or moral questions. They merely place the operator at a safe distance from the mission, which is otherwise unchanged. Inasmuch as the distance between a weapons platform and its operator is not itself a relevant factor in determining the justification of a military mission, moral concerns specifically addressed to the development and deployment of these new systems are misplaced. As with most exotic new technologies, the novelty of the technology itself blinds us to the fact that the moral issues involved in its use are entirely familiar and conventional and not appreciably different from those associated with the development of previous and current weapons technology.

There is some measure of truth in this claim, at least as applied to “remotely piloted” systems. A morally and legally justified military mission does not suddenly lose its justification, simply because the pilot is removed from the cockpit. The moral issues *are* nevertheless “technology driven,” at least in the sense that UAVs make feasible missions whose objectives would otherwise prove exceptionally risky, dangerous, and difficult (if not impossible) to undertake from a logistical standpoint. Hence, the criticisms of drone warfare in the literature reflect the legitimate concerns raised by this *dramatic expansion of mission feasibility*, to include an increase in the capacity to undertake or inflict:

1. Assassination or “targeted killing,” especially across or within the sovereign borders of nations with which the attacker is not formally at war
2. Collateral damage to innocents and their property
3. Thereby magnifying the probability for errors in targeting judgment (i.e., mistakes) that would not otherwise be made (since many of the missions in question could not otherwise be undertaken)
4. Perhaps most troublingly, increasing the opportunities and incidents of use of deadly force in a military context by nonmilitary personnel (such as civilian intelligence operatives or private security contractors)

The last emerging aspect of drone warfare is disconcerting on many levels but especially when it involves the summary execution, without any form of “due process,” of a nation’s own citizens allegedly engaged in criminal enterprises abroad.

While it is quite true that such anomalous and morally murky activities often occurred during the conduct of espionage and covert actions, it is equally true that drone technology substantially magnifies the prospects for engaging in such missions. This is one striking feature of unconventional or “irregular” warfare

(Lucas 2009b, 2010). First, adversaries and insurgent interest groups deal with the radical asymmetries in our favor in conventional settings, by disrupting social systems and attacking the weak links in logistical supply chains through the use (for example) of suicide bombers and IEDs that throw conventional military forces seriously off balance. Subsequently, a new technology, initially owned only by the besieged conventional forces, becomes the optimal response. Drone attacks by a technologically advanced nation are systemic in precisely the same sense as non-state actor's (or insurgent's) IEDs and suicide bombers, i.e., disrupting the adversary's command structure, relentlessly hunting him out where he lives and hides, and so demoralizing him in return and hopefully breaking his ability and will to fight. Subsequent legal and ethical questions that might arise stem from this relentless tactical arms race – this perennial “give and take” between political adversaries, or between security forces representing the rule of law, and international criminals intent on circumventing that rule.

119.2 Autonomous UAVs

In what way, if at all, does rendering the robotic or unmanned systems technology autonomous, self-governed, and self-directed (requiring little or no human oversight, once enabled and launched) significantly alter or transform this debate?

Paradoxically, if one is concerned (as in #4 above) about nonmilitary personnel operating military assets and undertaking inherently military operations, then the advent of machine autonomy would indeed transform that debate, since it would no longer be the case that nonuniformed personnel were engaging in military operations. Rather, these missions would henceforth be undertaken solely by machines (presumably with military authorization and oversight). This, in effect, turns “topsy-turvy” the “accountability dilemma” first raised by Robert Sparrow (2007, 2009, 2011) and Noel Sharkey (2007a, 2008b, 2010), in that defenders of these practices could now plausibly claim (albeit again rather disingenuously) that no “nonmilitary personnel” are engaged in questionable actions like extrajudicial killing of adversaries or of allegedly treasonous U.S. citizens. Only machines (and not persons) are undertaking the activities in question. In short, one can make the moral and legal anxieties raised by the policies and practices considered in the other essays in this volume simply disappear, merely by redefining the activities in question as involving “machine behavior exclusively,” and hence immune from the charges leveled against the remotely piloted missions.

This rather odd line of argument merely rings a change on a more familiar denial of responsibility by engineers involved in the research, development, manufacture, and ultimately deployment of any new weapons technology during wartime. That argument consists in the observation that a nation's defense industries, scientists, and privately contracted personnel are, to a large extent, servants or handmaidens of the policy decisions of the ruling government. It is the government and its political leadership and not its defense industries, scientists, or contractors that determine whether and how to prosecute its military conflicts and, accordingly,

whether to use military or nonmilitary means and personnel to do so. Indeed, the senior-level managers and directors of private military contractor and defense contracting firms are often eager to emphasize the point that their efforts support the defense of the State and that their industries and organizations exist solely to support the requirements of the State in carrying out this inherently governmental responsibility.

Such a neat division of labor, conveniently absolving defense contractors of their individual or organizational responsibilities, is hardly as definitive as it might appear, of course, inasmuch as leading military and political decision-makers (or trusted colleagues and associates) often leave government service to work in defense industries or for private military contractors, and vice versa. The boundaries between what or who is military or nonmilitary, public or private, are extremely porous. Even beyond these potential conflicts of interest, however, the public-private division of authority does not specifically excuse defense industries or their employees, in particular, for performing merely as mindless (let alone narrowly self-interested) technocrats. CEOs, as well as essential scientists, engineers, and employees of such industries, are also citizens of the State, who should be concerned to avoid increasing either the risk or the incidence of war through their efforts (even if they might justifiably claim that war's most destructive effects are lessened through their inventions). Their dual responsibilities as citizens, as well as subject-matter experts in military technology, might be said to run parallel to those of military personnel themselves to offer faithful advice, grounded in their professional experience, on the prospects and problems inherent in political policies regarding preparation for or engagement in military conflict (Cook 2004; Lucas 2009a). Thus, as P.W. Singer (2009, 2010), in particular, has forcibly maintained, scientists, engineers, and captains of industry engaged in the development and manufacture of military robots and unmanned systems generally must take upon themselves much more explicitly the responsibility for ensuring their wise and lawful use. This criticism has profound implications for the vocation of defense engineering that are treated in this chapter's conclusion.

119.3 Autonomous UAVs and Force Multiplication

From yet another vantage point, it is a well-recognized problem in robotics and unmanned systems research that the most substantial dividends from automating certain features of the battlefield would come from what is termed "force multiplication." On the one hand, at present, it is already the case that a single Predator or Reaper, remotely piloted by anywhere from one to five human operators, considerably magnifies the ability of military forces engaged in justified missions to accomplish those missions with greater success, precision, and reduction of risk of harm to innocents than their "manned" counterparts. This is an important dimension of what Bradley J. Strawser (2010) defines as "the Principle of Unnecessary Risk" (PUR): when otherwise engaged in legal and morally justifiable military or security missions, military and security personnel are owed as much safety and minimized

risk of harm as available military technology can afford them. The ability to operate several of these platforms simultaneously, under the supervision of a single operator, however, would considerably magnify mission capability without a corresponding increase in scarce and expensive (and nonexpendable) human personnel. Thus, the pursuit of “force multiplication” through some kind of systems engineering modification or technological enhancement of existing military weapons systems is simply an additional logical implication of PUR.

In principle, this objective of “force multiplication” with respect to military robotics in particular can be achieved, in turn, in either of two ways:

1. Either by improving the human-automation interface (e.g., cleaning up the current messy and cluttered Predator/Reaper control stations and replacing them with vastly simplified and more effective governance hardware)
2. Or else by endowing each unmanned platform with enhanced autonomy and independence of operation

The first of these alternative strategies is relatively straightforward and achievable through the application of thoroughly conventional engineering ingenuity (Cummings 2010a,b). One might, in principle, easily replace the four to five computer terminals (and their corresponding keyboards and “mice”) – all of which are presently required to operate a Predator drone – with a single joystick or smartphone “controller.” Importantly, *this path of engineering modification in the pursuit of force multiplication would not invoke any new ethical or legal challenges* beyond those already considered in the ongoing debate about the morality and legality of remotely piloted ISR and lethal missions that are wholly under human supervision and control. That is, *if* it is otherwise deemed ethical as well as legally permissible within the framework of international humanitarian law for a team of operators, for example, to conduct a remote strike against a Taliban or al-Qaeda stronghold, then it remains so even if one replaces the four or five remotely located operators with a single operator employing more efficient command and control technology. Again, this conclusion is consistent with Strawser’s PUR.

Far more esoteric and appealing to engineers and many policy analysts, however, is the “force multiplier” dividend attained by dispensing altogether with the human operator, whether “in” or merely “on” the loop (exercising merely supervisory or “executive” oversight). The appeal is many faceted, from the straightforward projected cost savings anticipated by acquisitions and supply officers to the improvements in “latency” (i.e., signal and reaction time), targeting, and overall mission effectiveness imagined by unmanned systems operators and commanders in the field. Computer scientists and artificial intelligence (AI) researchers are eager to take on such an interesting and engaging challenge.

But to concerned critics, such as Sparrow, Sharkey, Singer, or the eminent computer scientist and “father” of American robotics, George Bekey (2005), this *relentless drive toward machine autonomy* is the source of the most problematic moral and legal conundrums. The seemingly unreflective eagerness of scientists, engineers, and military and political leaders to move ahead with the development of autonomous weapons systems, notwithstanding these objections and concerns of critics, constitutes an attitude toward public welfare and the substantial risk of

unintended consequences that is characterized by these critics as ranging from what, in domestic tort and criminal law, are termed *reckless endangerment* to outright *criminal negligence*.

To make matters even worse, much of this dispute between proponents and critics of enhanced machine autonomy is mired in a nearly hopeless kind of conceptual confusion and linguistic equivocation. Proponents of increased machine autonomy often complicate the issues unnecessarily by invoking spurious concepts like machine “morality” and proposing “ethical governors” for lethally armed autonomous robots (e.g., Arkin et al. 2012). They misleadingly describe autonomous combat weapon systems that would be empowered to make “moral decisions and judgments” and also (in principle at least) experience the machine equivalent of “guilt” from sorties gone wrong and “learn” from those experiences. Consequentially, proponents argue, lethally armed, autonomous military robots will be “more ethical” and even “more humane” than their human counterparts. This use of moral terminology appears unnecessarily provocative and highly misleading, when what is actually being proposed is legal compliance, attained through machine and software program design and not capabilities more akin to human moral judgment.

Critics for their part worry needlessly about “killer robots” running amok (Asaro 2008; Krishnan 2009), as well as the inappropriateness of machines “making decisions to kill humans” (Larkin 2011; Billitteri 2010), or the lack of meaningful accountability for resulting “war crimes” that might consequently be committed (Sparrow 2007; Asaro 2008; Lucas 2011). Such critics appear to envision cyborgs (like “the Terminator”), or the infamous intelligent computer “HAL” (from Arthur C. Clarke’s science fiction novel, 2001: *A Space Odyssey*) in command on the bridge of a nuclear submarine, or “R2D2” and “C3PO,” fully weaponized and roaming the mountains of southern Afghanistan but unable to distinguish (without human supervision) between an enemy insurgent and a local shepherd.

Both extremes are, frankly, preposterous. Enhanced machine autonomy, even when married with lethal force, represents something far different from what either critics or proponents describe or imagine and presents its own unique ethical and legal challenges for engineering and industry, often quite different from those currently under discussion. For one thing, nothing so fanciful nor technologically infeasible as the outlandish scenarios outlined above is required, envisioned, or desired. Machines need not be “ethical,” and it is highly misleading to speak in this fashion. Likewise, machines cannot commit “war crimes,” inasmuch as they lack intentionality or self-motivation, and are utterly devoid of the interests (or emotions) required as ingredients for criminal culpability. Ethics, and accountability under international law, remain firmly and solely in the domain of human experience.

Likewise, the “autonomy” requisite for moral decision-making is something quite distinct from “machine autonomy.” The latter merely involves unmanned systems performing in complex environments without the need for continuous human oversight. In the latter, purely mechanical sense of “autonomy,” a Cruise missile and an iRobot “Roomba” vacuum cleaner are both “autonomous,” in that they perform their assigned missions, including encountering and responding to obstacles, problems, and unforeseen circumstances with minimal human oversight.

But the missile does not unilaterally change its mission en route or reprogram its targeting objectives – let alone does it raise “moral objections” about the appropriateness of the targets selected for it. Likewise, one would probably not wish to equip their Roomba with the additional capacity to “decide” whether or not it is necessary or appropriate to shoot an intruder their home.

In general, one neither needs nor desires autonomous machines to make those kinds of moral judgments. “Single -state, mission oriented” robot sentries in highly scripted environment, such as an “area-denial” scenario, constitute perhaps the sole exception to this general rule (Lucas 2011): e.g., sentry robots in a border or demilitarized” zone or protecting the so-called “no-go” zone around a Naval flotilla, in which ample warning is given, and all attempts made to dissuade violation of the boundaries or prohibited zone.

Paradoxically, the rhetoric of ethics and law have not proven very useful analytical tools in the analysis of machine autonomy, largely because both concepts are so widely misunderstood, misused, and misinterpreted by the participants in this debate. Critics of autonomous lethal platforms worry that these might be indiscriminate and uncontrollable, and in any case, unaccountable for their inevitable “war crimes,” and so illegal from the outset. But this concern, likewise, rests upon a conceptual equivocation: “R2D2” and “C3PO” in the imaginary Afghan scenario above could not ever commit “war crimes.” Unlike these fictional Hollywood portrayals, real autonomous systems have no intentionality or self-awareness. They do not care about their own survival. They have no “interests.” Importantly, they cannot “get scared,” or “get angry,” nor could they possibly try to “get even” (human emotions and intentions which are most often the source of war crimes). Rather, unmanned autonomous platforms would do precisely and only what they were programmed or commanded to do, *unless they happen to malfunction*. The anthropomorphic, romantic nonsense attached to robotics in the popular minds by Hollywood science fiction fantasies seriously compromises the ethical analysis of the use of genuine, real-world military robots within the confines of international law.

119.4 Ethics and Engineering

Robots that, by definition, cannot commit *war crimes* per se but only malfunction or make mistakes require a radically different metric for assessment than the moral metric that is too loosely and inaccurately applied. Moral evaluation of behavior pertains solely to human agents. For robots, as tools, by contrast, one must talk about *safety, reliability, and risk*. It would be madness, for example, to deploy a weapons system that was unsafe or unreliable, that killed the wrong people, or that killed innocent people, instead of destroying the legitimate enemy targets they were designed to attack. Accordingly, in international law as in domestic law, the expectation that would be laid down regarding accountability for wrongful death and destruction would be “due care” and the absence of reckless endangerment or criminal negligence (Lucas 2011). Deploying a patently unreliable system is surely

reckless (and therefore morally culpable) and in extreme circumstances might be found explicitly to be “criminally negligent.” The attitude that Singer, Sharkey, and others discern among engineers and defense industries, all too ready to pursue autonomy without adequate investigation of these problems, would constitute one glaring example of “criminal negligence.”

By contrast, deploying an autonomous platform that has proven to be safe and reliable under rigorous testing in stringent conditions would, on the whole, constitute an acceptable, even a morally responsible, action. If such a system malfunctioned (even as humans and their manned systems sometimes make mistakes, hit the wrong targets, or inadvertently kill the wrong people), then the procedure in the machine case would parallel that of the human case in similar circumstances. An inquiry would be held, and an investigation into circumstances conducted. If no intentional wrongdoing or culpable error by operators was discerned, then well-intentioned governments and their militaries would presumably issue an apology and do their best to make restitution for the harm inflicted.

In these respects, the dilemma of military robotics differs little from that of human combatants, who must be similarly trained to follow the applicable rules of engagement (ROEs) for a given conflict that in turn represent the translations by military lawyers of international humanitarian law and law of armed conflict into actionable mission parameters in given conflicts. Failure to comply due to misjudgments, accidents, and mistakes (that are not attributed to criminal negligence or recklessness, in the human case) is then acknowledged by the respective militaries and their governments: apologies are issued, and compensation for damages (ineffectual though that may be) is offered, in analogy with similar kinds of product liability situations in domestic law.

This process is grounded, however, in the prior expectation of what Michael Walzer (1977/2010) first termed “double intention” (in contrast to the more lenient moral criterion of double effect). Walzer’s principle of “double intention” would, in this instance, hold manufacturers and designers strictly accountable for exercising *due care* (as well as simply for not deliberately *intending* to do harm) in testing and assuring the reliability and safety of their commercial and industrial products (for robots and drones, when all is said and done, are nothing more than that). One would certainly seek to define the engineering design specifications for these products to require that they perform (in morally and legally relevant respects) as well or better than human combatants under similar circumstances. This specification would ensure that autonomous unmanned systems were fully compliant with the constraints of the law of armed conflict and applicable rules of engagement for a given conflict. If they can, and if they do achieve this benchmark engineering specification, then their use is morally justifiable. If they can’t, *or if designers and manufacturers have not taken due care to ensure that they can*, they one has no business building or deploying them. Just like Aegis and Patriot missiles and other semiautonomous systems, *due care* in the development and manufacture of new autonomous systems requires reliable target recognition threat-level escalation, sufficient to lead to a proportionate, discriminate, and therefore appropriate response in the use of deadly force.

The criterion that robots comply as or more effectively with applicable constraints of LOAC on their use of force and doing of harm than human combatants under similar circumstances constitutes the “Arkin Test” for robot “morality” (although this phrasing remains somewhat misleading, as the criterion pertains straightforwardly to compliance with international law, not with the exhibiting of moral judgment; Lucas 2010, 2011). In this sense, the test for “morality” (i.e., for the limited ability to comply with legal restrictions on the use of force) is similar to the “Turing Test” for machine intelligence: the criterion is satisfied whenever machine behavior is indistinguishable from (or better than) human behavior in any given context.

Satisfying this engineering criterion for safety and reliability is quite enough of a technological challenge, without muddying the waters with science fiction or specious metaphysical concerns. Engineers, in any case, are not close to being able to design machines that are fully capable of such activities. The systems they do propose to build and deploy at present have “autonomy” only in a very limited and highly scripted sense. Platforms such as “Sea Scouts,” Israeli-designed “Harpies,” and Korean “border sentry” robots, for example, are all designed to reliably target adversarial threats in extremely limited and very well-defined scenarios. In addition, these weapons systems usually focus on disarming a perceived threat or (as in the case of “Harpies”) dismantling the target’s weapons and command/control technology. Such systems are designed to employ force only in very limited situations that leave little room for ambiguity or error. The mistakes that might nonetheless be made are regrettable but usually not criminally negligent or culpable (as when a child strays into the DMZ or a boater foolishly transgresses the well-defined and well-publicized “no-go” zone around a military aircraft carrier). Naval engineer John Canning has long argued that this targeting of an adversary’s weapon systems (rather than the adversaries themselves) is the proper objective of lethally armed, fully autonomous, unmanned systems (Canning 2008; Canning et al. 2004). Otherwise, such systems should at most be armed with nonlethal weapons or programmed to undertake evasive action rather than use force in self-defense.

119.5 Conclusion

Machines equipped with lethal force need to be designed to operate according to precise engineering specifications, including the specifications that their actions comply accurately and unerringly with relevant international law pertaining to distinction, proportionality, and the avoidance of needless suffering (the so-called cardinal principles of international humanitarian law), as well as to the operant constraints of “standing” and specific rules of engagement (ROEs). Such constraints are identical to those currently imposed upon human combatants in the field (often with less than perfect success). It is needlessly provocative to cite the potential for robots to behave “more ethically” or for autonomous, lethally armed platforms to be “more humane” than human combatants. They are an entirely different class of

entity. Just as with rifles, missiles, and jet aircraft, one demands that they be reliable and safe to operate (meaning they don't malfunction and inadvertently destroy the operator or creator or wantonly destroy property or innocent human life). Ever greater degrees of machine autonomy may be desirable in order to increase the efficiency and "force multiplier" effects of using unmanned systems in a military's overall force mix. But they cannot, nor would a scientifically literate policymaker wish to have them, "behave ethically." Invoking this speculative criterion, in lieu demanding merely reliable operation of UAVs in strict compliance with the law international law of armed conflict, simply confuses the proper objectives of robotics research.

Singer (2009, 2010) is highly critical of what might be characterized as "ethical illiteracy" among engineers, whom he describes as uncomfortable with these questions and all too willing to proceed with their scientific and technological work while leaving the moral concerns to others outside their field. This criticism is not entirely fair: this introductory chapter on ethics and UAVs includes references to the extensive contributions to moral debate and the framing of moral issues in military robotics made by a number of scientists and engineers in the field (e.g., Arkin, Canning, Sharkey). Their work is every bit as significant, and perhaps more credible in their fields, than that undertaken by moral philosophers or experts in international relations. Moreover, much of the engineering contributions to the ethics of unmanned and future autonomous systems is appropriately published in well-regarded scientific and professional journals, such as the *Proceedings of the IEEE*.

Singer may be more accurate in complaining that far too many practitioners in the field are simply unaware of, or unconcerned with, such matters. Far too many members of the engineering profession simply have not stayed current with the ongoing and published research undertaken by their colleagues. Arkin, for example, has developed methods and a plan for operationalizing what he labels the "ethical governance" (i.e., legal compliance) for autonomous machine behavior. Other research teams have pursued alternative, less complex methods of achieving this goal and hence for satisfying the "Arkin Test" (e.g., McGhee et al. 2011; Tessier et al. 2010). Satisfying this test in *some* fashion, however, is not optional but constitutes instead a *strict operational and design specification* for future unmanned systems armed with lethal force. Together, these efforts define a program of research involving alternative testable and falsifiable operational parameters, designed to address this essential design specification for future autonomous systems. It therefore constitutes a serious lapse of professional ethics and must henceforth be recognized as professionally inexcusable, for practitioners in the field of military robotics engineering to remain unaware, unconcerned, and uninvolved in this scientific research. Deliberate and willful neglect of this essential dimension of the overall robotics engineering project constitutes what is known as "culpable ignorance." Practitioners guilty of it could henceforth find themselves morally culpable and very possibly criminally liable as well, simply for otherwise pursuing their current lines of research in the absence of this important dimension.

References

- K. Anderson, Targeted killing and drone warfare: how we came to debate whether there is a 'Legal Geography of War'. Washington College of Law Research paper #2011-16, American University, Washington, DC, 2009. Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1824783
- M. Anderson, The new robotics. *Discov. Mag.* 37–41 (2010). <http://www.discoverymagazine.com>
- R.C. Arkin, Governing lethal behavior: embedding ethics in a hybrid deliberative/reactive robot architecture, Report GIT-GVU-07-11, Georgia Institute of Technology's Gvu Center, Atlanta, 2007. <http://www.cc.gatech.edu/ai/robot-lab/online-publications/formalizationv35.pdf>
- R.C. Arkin, *Governing Lethal Behavior in Autonomous Robots* (Chapman & Hall/Taylor & Francis Group, Boca Raton, 2009)
- R.C. Arkin, The case for ethical autonomy in unmanned systems, in *New Warriors and New Weapons: Ethics & Emerging Military Technologies*, ed. by G.R. Lucas Jr. *J. Mil. Ethics* 9(4), 347–356 (2010)
- R.C. Arkin, P. Ulam, An ethical adaptor: behavioral modification derived from moral emotions, in *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA-09)*, Dajeon, Dec 2009a
- R.C. Arkin, A. Wagner, B. Duncan, Responsibility and lethality for unmanned systems: ethical pre-mission responsibility advisement, in *Proceedings of 2009 IEEE Workshop on Roboethics*, Kobe, May 2009b
- R.C. Arkin, P. Ulam, A.R. Wagner, Moral decision making in autonomous systems: enforcement, moral emotions, dignity, trust, and deception. *Proc. IEEE* 100(3), 571–589 (2012)
- P. Asaro, How just could a Robot war be?, in *Current Issues in Computing and Philosophy*, ed. by P. Brey, A. Briggle, K. Waelbers (IOS Press, The Netherlands, 2008), pp. 50–64
- S. Banda, The challenges of achieving UAV autonomy. Graduate School of Engineering and Applied Sciences Distinguished Lecture Series, Naval Postgraduate School (2 Dec 2010). Excerpted in "Update: Naval Postgraduate School Newsletter" (Jan 2011), p. 5. <http://www.nps.edu/About/News/World-Renowned-UAV-Control-Expert-Presents-GSEAS-Distinguished-Guest-Lecture.html>. Accessed 17 Jan 2011
- G. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control* (MIT, Cambridge, 2005)
- T.J. Billitteri, Drone warfare: are strikes by unmanned aircraft ethical? *CQ Res.* 20(28), 653–676 (2010)
- D. Brutzman et al., The phoenix autonomous underwater vehicle, in *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, ed. by D. Kortenkamp et al. (MIT, Cambridge, 1998), pp. 323–360
- J. Canning, Weaponized unmanned systems: a transformational warfighting opportunity, government roles in making it happen, in *Proceedings of Engineering the Total Ship (ETS)*, Falls Church, 23–25 Sept 2008
- J. Canning, G.W. Riggs, O.T. Holland, C. Blakelock, A concept for the operation of armed autonomous systems on the battlefield, in *Proceedings of Association for Unmanned Vehicle Systems International's (AUVSI) Unmanned Systems North America*, Anaheim, 3–5 Aug 2004
- M.L. Cook, *The Moral Warrior* (State University of New York Press, Albany, 2004)
- M.L. Cummings, Creating moral buffers in weapons control design. *IEEE Technol. Soc. Mag.* 41(Fall), 2–33 (2004)
- M.L. Cummings et al., The role of human-automation consensus in multiple unmanned vehicle scheduling. *Hum. Factors J. Hum. Factors Ergon.* 52(1), 17–27 (2010a)
- M.L. Cummings et al., Assessing operator workload and performance in expeditionary multiple unmanned vehicle control, in *Proceedings of the 48th AIAA Aerospace Sciences Meeting*, Orlando, Jan 2010b

- Human Rights Institute, Columbia Law School, Targeting operations with drone technology: humanitarian law implications, in *Background Note for the American Society of International Law Annual Meeting*, New York, 25 Mar 2011. http://www.law.columbia.edu/ipimages/Human_Rights_Institute/BackgroundNoteASILColumbia.pdf
- A. Krishnan, *Killer Robots: Legality and Ethicality of Autonomous Weapons* (Ashgate Press, London, 2009)
- M.S. Larkin, Brave new warfare: autonomy in lethal UAVs. Master's thesis, Naval Postgraduate School, Monterey, 2011. [https://wiki.nps.edu/download/attachments/15073701/Brave+New+Warfare+\(Larkin,+Matthew+Thesis\).pdf?version=1&modificationDate=1301324368000](https://wiki.nps.edu/download/attachments/15073701/Brave+New+Warfare+(Larkin,+Matthew+Thesis).pdf?version=1&modificationDate=1301324368000)
- P. Lin, G. Bekey, K. Abney, *Autonomous Military Robotics: Risk, Ethics, and Design* (U.S. Department of the Navy, Office of Naval Research, Washington, DC, 2008), p. 112
- G.R. Lucas Jr., Advice and dissent: the 'uniform' perspective. *J. Mil. Ethics* **8**(2), 141–161 (2009a)
- G.R. Lucas Jr., 'This is not your father's war': confronting the moral challenges of 'unconventional' war. *J. Natl. Secur. Law Policy* **3**(2), 331–342 (2009b)
- G.R. Lucas Jr., Postmodern war, in *New Warriors and New Weapons: Ethics & Emerging Military Technologies*. *J. Mil. Ethics* **9**(4), 289–298 (2010)
- G.R. Lucas Jr., Industrial challenges of military robotics. *J. Mil. Ethics* **10**(4), 274–295 (2011)
- G. Marchant et al., International governance of autonomous military robotics. *Columbia Sci. Technol. Law Rev.* **12**(272) (2011). <http://www.stlr.org/cite.cgi?volume=12&article=7>
- McCain, New warriors/new weapons, in *Executive Summary of the 2010 McCain Conference of Service Academies and War Colleges*, ed. by Col. Edward Barrett, USAF Reserve; G.R. Lucas Jr. (ed.), *New Warriors and New Weapons: Ethics & Emerging Military Technologies*. *J. Mil. Ethics* **9**(4), 416–423 (2010). Also available at <http://www.usna.edu/ethics/Publications/Mac10exec.pdf>
- R.B. McGhee, D.P. Brutzman, D.T. Davis, A universal multiphase mission execution automation (MEA) with prolog implementation for unmanned untethered vehicles, in *Proceedings of the 17th International Symposium on Unmanned Untethered Submersible Technology*, Portsmouth, 2011. Available at <https://savage.nps.edu/AuvWorkbench/website/documentation/papers/UUST11MEA-2011June22.pdf>
- OSD, Office of the secretary of defense, in *FY 2009–2034 Unmanned Systems Integrated Roadmap*, 2nd edn. (2009). <http://www.acq.osd.mil/psa/docs/UMSIntegratedRoadmap2009.pdf>
- N. Sharkey, Robot wars are a reality, in *The Guardian* (UK, 2007a), p. 29. <http://www.guardian.co.uk/commentisfree/2007/aug/18/comment.military>
- N. Sharkey, Automated killers and the computing profession. *Computer* **40**, 122–124 (2007b)
- N. Sharkey, Cassandra or false prophet of doom: AI robots and war. *IEEE Intell. Syst.* **23**, 14–17 (2008a)
- N. Sharkey, Grounds for discrimination: autonomous robot weapons. *RUSI Def. Syst.* **11**(2), 86–89 (2008b)
- N. Sharkey, Saying 'No!' to lethal autonomous targeting, in *New Warriors and New Weapons: Ethics & Emerging Military Technologies*, ed. by G.R. Lucas Jr. *J. Mil. Ethics* **9**(4), 299–313 (2010)
- P.W. Singer, *Wired for War* (Penguin Press, New York, 2009)
- P.W. Singer, The ethics of 'killer apps', in *New Warriors and New Weapons: Ethics & Emerging Military Technologies*, ed. by G.R. Lucas Jr. *J. Mil. Ethics* **9**(4), 314–327 (2010)
- R. Sparrow, Killer robots. *J. Appl. Philos.* **24**(1), 62–77 (2007)
- R. Sparrow, Building a better WarBot: ethical issues in the design of unmanned systems for military applications. *Sci. Eng. Ethics* **15**(2), 169–187 (2008)
- R. Sparrow, Predators or plowshares? Arms control of robotic weapons. *IEEE Technol. Soc. Mag.* **28**(1), 25–29 (2009)
- R. Sparrow, Robotic weapons and the future of war, in *New Wars and New Soldiers: Military Ethics in the Contemporary World*, ed. by P. Tripodi, J. Wolfendale (Ashgate Publishing Ltd., London, 2011), pp. 117–133

-
- B.J. Strawser, Moral predators: the duty to employ uninhabited vehicles, in *New Warriors and New Weapons: Ethics & Emerging Military Technologies*, ed. by G.R. Lucas Jr. *J. Mil. Ethics* **9**(4), 357–383 (2010)
- C. Tessier et al., Détection et résolution de conflits d'autorité dans un système homme-robot. *Revue D'intelligence Artificielle* **24**(3), 325–356 (2010)
- W. Wallach, C. Allen, *Moral Machines: Teaching Robots Right from Wrong* (Oxford University Press, New York, 2009)
- M. Walzer, *Just and Unjust Wars* (Basic Books, New York, 1977/2010)

Gary E. Marchant, Braden Allenby, Ronald C. Arkin,
Jason Borenstein, Lyn M. Gaudet, Orde Kittrie, Patrick Lin,
George R. Lucas Jr., Richard O'Meara, and Jared Silberman

G.E. Marchant (✉)

Arizona State University School of Law, Tempe, AZ, USA

e-mail: Gary.Marchant@asu.edu

B. Allenby

School of Sustainable Engineering and the Built Environment in Civil Engineering, Arizona State University, Tempe, AZ, USA

e-mail: Braden.Allenby@asu.edu

R.C. Arkin

Department of Computer Science, Georgia Tech University, Atlanta, GA, USA

e-mail: ronald.arkin@cc.gatech.edu

J. Borenstein

Center for Ethics & Technology, Georgia Tech University, Atlanta, GA, USA

e-mail: jason.borenstein@gatech.edu

L.M. Gaudet

Center for Law, Science & Innovation Sandra Day O'Connor School of Law, Arizona State University, Tempe, AZ, USA

O. Kittrie

Arizona State University School of Law, Tempe, AZ, USA

e-mail: orde.kittrie@asu.edu

P. Lin

Center for Ethics & Emerging Technologies, California Polytechnic University, San Luis Obispo, CA, USA

e-mail: palin@calpoly.edu

G.R. Lucas Jr.

United States Naval Academy, Annapolis, MD, USA

Naval Postgraduate School, Monterey, CA, USA

e-mail: grlucas@nps.edu

R. O'Meara

Division of Global Affairs - Rutgers, The State University of New Jersey, Conklin Hall, NJ, USA

J. Silberman

Associate Counsel for Arms Control and International Law, United States Navy Office of Strategic Systems Programs (SSP), Washington, DC, USA

Contents

120.1 Introduction 2880

120.2 Background on Autonomous Military Robotics 2881

120.3 Ethical and Policy Aspects 2885

 120.3.1 Responsibility and Risks 2886

 120.3.2 Legal Status of Civilians 2886

 120.3.3 Complexity and Unpredictability 2887

 120.3.4 Just War Theory 2888

 120.3.5 Civilian Applications 2889

 120.3.6 Broader Ethical and Social Considerations 2889

120.4 Existing Governance Mechanisms for Military Robots 2891

120.5 Legally Binding International Agreements 2897

 120.5.1 Menu of Types of Restrictions Contained in Current
 International *Legal Arms Control Instruments* 2898

 120.5.2 Cautionary Note Regarding the Utility of International Legal
 Arms Control Instruments 2901

120.6 Soft Law/Governance Approaches 2901

 120.6.1 Codes of Conduct 2902

 120.6.2 Transgovernmental Dialogue 2904

 120.6.3 Information-Sharing and Confidence-Building Measures 2906

 120.6.4 Framework Convention 2906

120.7 Conclusion 2907

References 2908

Abstract

This article seeks to provide a background of some of the principal legal issues and initiate a much-needed legal and ethical dialogue related to the use of lethal autonomous robotic technologies in the military context. Following a brief introduction, Sect. 120.2 of this article provides a brief history and illustrations of autonomous robots in the military, including the pending development of LARs. Section 120.3 sets forth a number of important ethical and policy considerations regarding the use of robots in military endeavors. Section 120.4 reviews the current patchwork of guidelines and policies that apply to the use of military robots. Section 120.5 considers the role that international treaties and agreements might play in the governance of LARs, while Sect. 120.6 investigates the potential role of soft law governance mechanisms such as codes of conduct.

120.1 Introduction

Military technology is a field driven by change – the constant pursuit to be better, faster, and stronger. Certain technological achievements like guns and planes have happened in the purview of the public and have revolutionized the world of war as we know it. Yet many technological changes have occurred under the radar, in military labs and private test fields, with the majority of citizens unaware of the leaps and bounds of progress. Robotics is one such modern military technology advancement that has largely escaped public attention to date. Combining the most

advanced electronic, computer, surveillance, and weapons technologies, the robots of today have extraordinary capabilities and are quickly changing the landscape of battle and dynamics of war. One of the most important achievements has been the creation of robots with autonomous decision-making capability [1]. In particular, the development of autonomous robots capable of exerting lethal force, known as lethal autonomous robots (“LARs”) has significant implications for the military and society.

A variety of never before anticipated complex legal, ethical, and political issues have been created, issues in need of prompt attention and action. There have recently been growing calls for the potential risks and impacts of LARs to be considered and addressed in an anticipatory and preemptive manner. For example, in October 2010, a United Nations human rights investigator recommended in a report to the United Nations that “[t]he international community urgently needs to address the legal, political, ethical and moral implications of the development of lethal robotic technologies” [2]. In September 2010, a workshop of experts on unmanned military systems held in Berlin issued a statement (supported by a majority but not all of the participants) calling upon “the international community to commence a discussion about the pressing dangers that these systems pose to peace and international security and to civilians...” [3]. While there is much room for debate about what substantive policies and restrictions (if any) should apply to LARs, there is broad agreement that now is the time to discuss those issues. The recent controversy over unmanned aerial vehicles (“UAVs”) that are nevertheless human-controlled (often referred to as “drones”) demonstrates the importance of anticipating and trying to address in a proactive manner the concerns about the next generation of such weapons, autonomous lethal robotics [4].

120.2 Background on Autonomous Military Robotics

In the United States there has been a long tradition of applying innovative technology in the battlefield that has often translated into military success [5]. The Department of Defense (“DOD”) naturally extended this approach to robotics. Primary motivators for the use of intelligent robotic or unmanned systems in the battlefield include:

- Force multiplication* – with robots, fewer soldiers are needed for a given mission, and an individual soldier can now do the job of what took many before.
- Expanding the battlespace* – robots allow combat to be conducted over larger areas than was previously possible.
- Extending the warfighter’s reach* – robotics enable an individual soldier to act deeper into the battlespace, for example, seeing farther or striking farther.
- Casualty reduction* – robots permit removing soldiers from the most dangerous and life-threatening missions.

The initial generation of military robots generally operate under direct human control, such as the “drone” unmanned aerial vehicles being used by the U.S.

Fig. 120.1 Phalanx Close-In Weapons System (U.S. Navy Photograph)



military for unmanned air attacks in Pakistan, Afghanistan, and other theaters [6]. However, as robotics technology continues to advance, a number of factors are pushing many robotic military systems toward increased autonomy. One factor is that as robotic systems perform a larger and more central role in military operations, there is a need to have them to continue to function (just as a human soldier would) if communication channels are disrupted. In addition, as the complexity and speed of these systems increase, it will be increasingly limiting and problematic for performance levels to have to interject relatively slow human decision-making into the process. As one commentator recently put it, “[m]ilitary systems (including weapons) now on the horizon will be too fast, too small, too numerous and will create an environment too complex for humans to direct” [7].

Based on these trends, many experts believe that autonomous, and in particular lethal autonomous, robots are an inevitable and relatively imminent development [8]. Indeed, several military robotic automation systems already operate at the level where the human is still in charge and responsible for the deployment of lethal force, but not in a directly supervisory manner. Examples include (i) the Phalanx system for Aegis-class cruisers in the Navy “capable of autonomously performing its own search, detect, evaluation, track, engage and kill assessment functions” [9] (Fig. 120.1); (ii) the MK-60 encapsulated torpedo (CAPTOR) sea mine system, one of the Navy’s primary antisubmarine weapons capable of autonomously firing a torpedo and cruise missiles (Fig. 120.2); (iii) the Patriot anti-aircraft missile batteries; (iv) “fire and forget” missile systems generally; and (v) antipersonnel mines or alternatively other more discriminating classes of mines (e.g., antitank) [10]. These devices can each be considered to be robotic by some definitions, as they all are capable of sensing their environment and actuating, in these cases, through the application of lethal force.

In 2001, Congress issued a mandate that stated that by 2010 one-third of all U.S. deep strike aircraft should be unmanned and by 2015 one-third of all ground vehicles should be likewise unmanned [11]. More recently, the U.S. Department of Defense (“DOD”) issued in December 2007 an Unmanned Systems Roadmap spanning



Fig. 120.2 Tomahawk cruise missile on display (U.S. Navy Photograph)

25 years, reaching until 2032, that likewise anticipated and projected a major shift toward greater reliance on unmanned vehicles in U.S. military operations [12].

As early as the end of World War I, the precursors of autonomous unmanned weapons appeared in a project on unpiloted aircraft conducted by the U.S. Navy and the Sperry Gyroscope Company [13]. Multiple unmanned robotic systems are already being developed or are in use that employ lethal force such as the ARV (armed robotic vehicle), a component of the Future Combat System (“FCS”); Predator UAVs (unmanned aerial vehicles) equipped with hellfire missiles, which have already been used in combat but under direct human supervision; and the development of an armed platform for use in the Korean Demilitarized Zone to name a few [14].

The TALON SWORDS platform (Fig. 120.3) developed by Foster-Miller/QinetiQ has already been put to test in Iraq and Afghanistan and is capable of carrying lethal weaponry (M240 or M249 machine guns or a Barrett .50 caliber rifle). Three of these platforms have already served for over a year in Iraq and as of April 2008 and were still in the field at the time, contrary to some unfounded rumors [15]. A newer version, referred to as MAARS (Modular Advanced Armed Robotic System), is ready to replace the earlier SWORDS platforms in the field. The newer robot can carry a 40 mm grenade launcher or an M240B machine gun in addition to various nonlethal weapons. The President of QinetiQ stated the purpose of the robot is to “enhance the warfighter’s capability and lethality, extend his situational awareness and provide all these capabilities across the spectrum of combat” [16].

It is interesting to note that soldiers have already surrendered to unmanned aerial vehicles (“UAVs”) even when the aircraft has been unarmed. The first documented instance of this occurred during the 1991 Gulf War. An RQ-2A Pioneer UAV, used

Fig. 120.3 Foster-Miller
TALON SWORDS robot
(Department of Defense
Photograph)



for battle damage assessment for shelling originating from the USS Wisconsin, was flying toward Faylaka Island, when several Iraqis hoisted makeshift white flags to surrender, thus avoiding another shelling from the battleship [17]. Anecdotally, most UAV units during this conflict experienced variations of attempts to surrender to the Pioneer. A logical assumption is that this trend will only increase as UAVs' direct response ability and firepower increases.

The development of autonomous lethal robotics raises questions regarding if and how these systems can conform as well or better than our soldiers with respect to adherence to the existing laws of war. This is no simple task however. In the fog of war it is hard enough for a human to be able to effectively discriminate whether or not a target is legitimate. Fortunately for a variety of reasons, it may be anticipated, despite the current state of the art, that in the future autonomous robots may be able to perform better than humans under these conditions, for the following reasons [18]:

- The ability to act conservatively: i.e., they do not need to protect themselves in cases of low certainty of target identification. Autonomous armed robotic vehicles do not need to have self-preservation as a foremost drive, if at all.

They can be used in a self-sacrificing manner if needed and appropriate without reservation by a commanding officer.

- The eventual development and use of a broad range of robotic sensors better equipped for battlefield observations than humans' currently possess.
- They can be designed without emotions that cloud their judgment or result in anger and frustration with ongoing battlefield events. In addition, "Fear and hysteria are always latent in combat, often real, and they press us toward fearful measures . . ." [19]. Autonomous agents need not suffer similarly.
- Avoidance of the human psychological problem of "scenario fulfillment" is possible, a factor believed partly contributing to the downing of an Iranian airliner by the USS Vincennes in 1988 [20]. This phenomenon leads to distortion or neglect of contradictory information in stressful situations, where humans use new incoming information in ways that only fit their preexisting belief patterns, a form of premature cognitive closure. Robots can be developed so that they are not vulnerable to such patterns of behavior.
- Robots can integrate more information from more sources far faster before responding with lethal force than a human possibly could in real time. This data can arise from multiple remote sensors and intelligence (including human) sources, as part of the Army's network-centric warfare concept and the concurrent development of the Global Information Grid [21]. "Military systems (including weapons) now on the horizon will be too fast, too small, too numerous and will create environments too complex for humans to direct" [22].
- When working in a team of combined human soldiers and autonomous systems as an organic asset, they have the potential capability of independently and objectively monitoring ethical behavior in the battlefield by all parties and reporting infractions that might be observed. This presence alone might possibly lead to a reduction in human ethical infractions.

The trend is clear: warfare will continue and autonomous robots will ultimately be deployed in the conduct of warfare. The ethical and policy implications of this imminent development are discussed next, followed by a discussion of governance options.

120.3 Ethical and Policy Aspects

There are numerous ethical, policy, and legal issues relating to creation and deployment of lethal autonomous robots. Although an exhaustive list of these issues will not be offered here, a number of the key ones will be outlined. Rather than defending a particular point of view on the technology, the primary aim is to promote awareness of these issues and to encourage lawyers, policymakers, and other relevant stakeholders to consider what may be appropriate legal and regulatory responses to LARs as they are being developed.

120.3.1 Responsibility and Risks

Australian philosopher Robert Sparrow has been a prominent voice in debates about the ethics of lethal autonomous robots. For instance, he examines the complexities associated with assigning ethical and legal responsibility to someone, or something, if an autonomous robot commits a war crime [23]. He considers several possible solutions to this puzzle, including “the programmer,” “the commanding officer,” and “the machine” itself but concludes that each option has its profound share of difficulties [24]. Remaining neutral on whether Sparrow is correct, assigning responsibility to a LAR’s behavior (or misbehavior) is an important matter that warrants further investigation. Along related lines, Peter Asaro doubts whether a robot can be punished in a meaningful way since it is unlikely to possess any form of moral agency, observing that traditional notions from criminal law such as “rehabilitation” and “deterrence” do not seem applicable here [25].

One of the principal justifications for relying on autonomous robots is that they would have access to a greater amount of information than any human soldier. Yet this advantage raises key questions, including whether a robot should ever be permitted to refuse an order and, if so, under what conditions. Refusing an order on the battlefield is of course a serious matter, but should a robot be given more latitude to do so than a human soldier?

Battlefield units containing human soldiers and LARs will raise additional ethical issues relating to risk and responsibility. According to Sparrow, “human beings will start to be placed in harm’s way as a result of the operations of robots” [26]. But will soldiers genuinely be aware of the kinds of risks they are being exposed to when working with robotic counterparts? If not, Sparrow fears that soldiers might place too much trust in a machine, assuming, for example, it has completed its assigned tasks [27]. And even if soldiers are fully aware of the risks associated with reliance on robotics, how much additional risk exposure is justifiable?

120.3.2 Legal Status of Civilians

In any treatment of LARs, the question of potential liability to civilians must be considered. Civilians initially responsible for empowering or placement of LARs may not necessarily be absolved of legal responsibility should the LAR perform unintended consequences. For example, it is entirely possible that a failure to recognize and obey an attempt for surrender could invoke a violation of the Law of Armed Conflict (“LOAC”). If a civilian software writer failed to initially write code that recognized the right to surrender through a flag of truce or other means, then a “reach back” to civilian liability for the breach might be possible. Likewise, if the civilian software writer failed in attempts to program identifiable legally protected structures such as places of worship, hospitals, or civilian schools, the code writer may be subject to potential liability in those scenarios as well.

The use of smaller yet lethal robots is gaining acceptance in battlefield operations. It is entirely possible that a small three foot or less autonomous robot might be thrown or “launched” into an open building or window with lethal gunfiring capabilities much like a whirling dervish [28]. Should this whirling dervish either not be programmed to accept or fail to recognize internationally established symbols of surrender such as a white flag, would the robot’s responsible forces, probably including civilian software code writers, escape legal liability for these omissions?

Similarly, well-intentioned and seemingly complete software programming might go astray in other ways. Under law of war concepts, and specifically the Hague Convention (Hague VIII), it is forbidden to lay unanchored automatic contact mines unless they will become harmless at 1 h or less after the mines are no longer under the control of the person (nation) who laid them [29]. If one examines unmanned vehicles, particularly unmanned underwater sea vehicles launched from a manned “mother ship,” in the event that the mother ship become lost or disabled, the unmanned “robotic” ship vehicle must, consistent with international navigational regulations and protocols, be able on its own to comply with the navigational regulations and responsibilities. No matter the degree of robotic autonomy, a legal responsibility is likely to exist for the actions of the untethered “loose” unmanned vehicle.

Consider the situation described by P.W. Singer in “Wired for War” in which a robot programmed to perform sentry work failed to identify a threatening action [30]. In that scenario in which a bar patron suffers from the hiccups, a robot trained to act as a sentry with accompanying lethal force could reasonably fail to identify the well-intentioned furtive handgun gesture of the bartender so that the patron might be scared so as to lose the hiccups. If the robot sentry failed to identify this mimicked handgun gesture and mistakenly shot the bartender “thinking” a lethal threat existed, it is conceivable that the software writer of the code might be responsible for the mistaken actions.

The above examples illustrate how seemingly complete autonomous robotic systems may still pose legal liability issues upon the civilians initially responsible for their use within battlespace operations. Amidst these scenarios, the civilian software code writer’s work and ultimate responsibilities may enjoy a much longer and unanticipated legal life.

120.3.3 Complexity and Unpredictability

Unfortunately, a full awareness of the risks from autonomous robots may be impossible. Wallach and Allen discuss how predicting the relevant dangers can be fraught with uncertainty [31]. For example, a semiautonomous antiaircraft gun accidentally killed several South African soldiers [32]. Roger Clarke pointed out years ago that, “[c]omplex systems are prone to component failures and malfunctions, and to intermodule inconsistencies and misunderstandings” [33]. Blay Whitby echoes the notion by arguing that computer programs often do not behave as predictably

as software programmers would hope [34]. Experts from computing, robotics, and other relevant communities need to continue weighing in on the matter so the reliability of LARs can be more thoroughly assessed.

Perhaps robot ethics has not received the attention it needs, given a common misconception that robots will do only what we have programmed them to do. Unfortunately, such a belief is sorely outdated, harking back to a time when computers were simpler and their programs could be written and understood by a single person. Now, programs with millions of lines of code are written by teams of programmers, none of whom knows the entire program; hence, no individual can predict the effect of a given command with absolute certainty, since portions of large programs may interact in unexpected, untested ways. Even straightforward, simple rules such as Asimov's Laws of Robotics can create unexpected dilemmas [35]. Furthermore, increasing complexity may lead to *emergent behaviors*, i.e., behaviors not programmed but arising out of sheer complexity [36].

Related major research efforts also are being devoted to enabling robots to learn from experience. Learning may enable the robot to respond to novel situations, an apparent blessing given the impracticality and impossibility of predicting all eventualities on the designer's part. But this capability raises the question of whether it can be predicted with reasonable certainty *what* the robot will learn. Arguably, if a robot's behavior could be adequately predicted, the robot would just be programmed to behave in certain ways in the first place instead of requiring learning. Thus, unpredictability in the behavior of complex robots is a major source of worry, especially if robots are to operate in unstructured environments, rather than the carefully structured domain of a factory or test laboratory.

120.3.4 Just War Theory

An overarching concern is whether the use of LARs is consistent with time-honored principles, rules, and codes that guide military operations, including just war theory, the Law of Armed Conflict ("LOAC"), and the Rules of Engagement. Scholars are already starting to analyze whether LARs will be capable of fulfilling the requirements of just war theory. Noel Sharkey, for example, doubts that robots will be capable of upholding the principle of discrimination, which involves being able to distinguish between combatants and non-legitimate targets such as civilians and surrendering soldiers [37]. While discussing military robots and the principle of proportionality, Sparrow argues that "decisions about what constitutes a level of force *proportionate to the threat* posed by enemy forces are extremely complex and context dependent and it is seemingly unlikely that machines will be able to make these decisions reliably for the foreseeable future" [38]. However, at this point, it remains an open question whether the differences between LARs and existing military technology are significant enough to bar the former's use.

Further, Asaro examines whether military robots may actually encourage wars by altering pre-conflict proportionality calculations and last resort efforts [39]. A fundamental impediment to war is the loss of human life, especially the lives of

fellow citizens; casualties are a significant reason why wars are not more common. Sending an army of machines to war – rather than friends and relatives – may not exact the same physical and emotional toll on a population [40]. Assuming the existence of a just cause, one could celebrate this new calculus, which more readily permits legitimate self-defense. However, this reduced cost may, in turn, reduce the rigor with which nonviolent alternatives are pursued and thus encourage unnecessary – and therefore unjust – wars. While this possible moral hazard obviously does not require us to maximize war costs, it does require efforts to inform and monitor national security decision-makers.

Finally, Singer suggests that these LARs weapons could undermine counterinsurgency efforts, where indigenous respect and trust is crucial to creating a reasonable chance of success [41]. Unmanned weapons may be perceived as indicative of flawed characters and/or tepid commitments and are incapable of developing necessary personal relationships with local citizens. And even if remote controlled or autonomous are more discriminate than soldiers, they are commonly perceived as less so.

120.3.5 Civilian Applications

Technology developed for military purposes frequently has civilian applications and vice versa. For instance, Singer notes how the REMUS, the Remote Environmental Monitoring Unit, was originally used by oceanographers, but later an altered version of it was deployed in Iraq [42]. Further, many federal and state agencies have sought permission to use military technology such as UAVs [43]. Consequently, once LARs are developed for military use, what might the implications be down the road for their use in civilian contexts [44]?

120.3.6 Broader Ethical and Social Considerations

Any decisions and policies regarding military development and use of LARs will impact, and be impacted by, broader technological and social considerations [45]. The failure to acknowledge these considerations upfront, and include them in the analysis, can lead to dysfunctional results and unnecessary failure of legal and policy initiatives. Accordingly, we will highlight some relevant considerations and their potential implications.

Consider first what a LAR actually is: one of many potential functions platformed on a generic technology base, which itself may be highly variable. Thus, a “lethal” function, such as a repeating kinetic weapon mounted on a robotic system, is one that is intentionally programmed to identify, verify, and eliminate a human target. At other times, the same basic robotic system might be fitted for cargo carrying capacity, for surveillance, or for many other functions, either in a military or a civil capacity. “Autonomous” means that the platform is capable of making the necessary decisions on its own, without intervention from a human. This, again, may involve

the lethality function, or it may not: one might, for example, tell a cargo robot to find the best way to a particular destination, and simply let it go [46].

Similarly, “robot” may sound obvious, but it is not. Tracked machines, such as the Talon or PackBot, or UAVs, such as the Predator or Raven, are fairly obvious robotic technologies, but what about “smart” land mines that are inert until they sense the proper triggering conditions and are capable of jumping, or exploding, or doing whatever else they are built to do [47]? What about bombs that, once deployed, glide above the battlefield until finding an enemy target, which are then attacked while sparing cars, busses, and homes [48]? And what about a grid of surveillance/attack cybersects (insect size robots or cyborgs consisting of biological insects with robotic functions integrated into them)? Each cybersect taken alone may be too insignificant dumb to be considered a robot, but the cybersect grid as a whole may actually be quite intelligent [49]. Going one step further, what should we call a weapons platform that is wirelessly connected directly into a remote human brain (in recent experiments, a monkey at Duke University with a chip implanted in its brain that wirelessly connected it to a robot in Japan kept the Japanese robot running by thought so that the robot was in essence an extension of its own physicality) [50]? Even now, the Aegis computer fire control system deployed on Navy vessels comes with four settings: “semiautomatic,” where humans retain control over the firing decision; “automatic special,” where humans set the priorities but Aegis determines how to carry the priorities out; “automatic,” where humans are kept in the loop but the system works without their input; and “casualty,” where the system does what it thinks necessary to save the ship [51].

This brief digression raises serious questions about what a LAR actually is. Certainly, it has a technology component, but in some ways this is almost trivial compared to its social, political, ethical, and cultural dimensions. In fact, one might well argue that in many important ways, a LAR is more of a cultural construct than a technology, with its meaningful dimension being intent rather than the technology system [52]. This is an important point, for it suggests that any legal or regulatory approach that focuses on technology may be misplaced; conversely, it means that the underlying technologies which come together in a LAR will continue to evolve independent or irrespective of any direct controls on LARs – including the functionality of the physical hardware, the sophistication of the software, and the integrated technological capability we call “autonomy.”

It is because the technologies are separate from the use that the discussion of LARs is frequently confused: LARs are often discussed as if they were a “military technology,” when in fact they are a set of technologies that can be integrated in ways that are effective and desirable given current military conditions. Let us begin by identifying two levels at which technologies function: Level I, or the shop floor level, and Level II, or the institutional and social system level [53]. Thus, for example, if one gives a vaccine to a child to prevent her from getting a particular disease, one is dealing with a Level I technology system: the desired goal, no disease, is inherent in use of the technology. On the other hand, if one starts a vaccine program in a developing country in order to encourage economic growth because of

better health, it is a Level II system: use of vaccines may contribute to such a goal, but there are many intervening systems, pressures, policies, and institutions.

To return to LARs, then, one might begin by asking why deploy such a technology in any form? Here, one has serious coupling between Level I and Level II issues. The immediate Level I response is that deployment of LARs would save soldiers' lives on the side that deployed them; many explosive devices in Iraq and Afghanistan that might otherwise have killed and maimed soldiers have been identified, and eliminated, by robots. But this is in some ways only begging a serious Level II question. In World War I, for example, generals thought little of killing 100,000 men at a go by sending them into the teeth of concentrated machine gunfire. Consequently, simply avoiding casualties is an inadequate explanation. That world, however, has changed, especially for the U.S. military, which faces a particularly stark dilemma. It is charged by its citizens with being able to project force anywhere around the world, under virtually any conditions. But, for a number of reasons, American civilians have become increasingly averse to any casualties. So the U.S. military finds itself in the dilemma of being required to project its power without American soldiers dying. Additionally, the long-term demographics are not good: Americans, like other developed countries, are looking at an aging demographic, with the immediate implication that there are less young people to fill boots on the ground [54].

The institutional and social context of military operations for the United States is increasingly one where better military productivity becomes paramount, with productivity measured as mission accomplishment per soldier lost. And robots can potentially contribute significantly to achieving such productivity. It's not just about saving soldier's lives, a Level I technology. It's also about building the capability to continue to project power with fewer casualties, and to do so because culture and society are changing to make fatalities, whether soldier or civilian, less acceptable, which are Level II trends.

In sum, LARS raise a broad range of complex ethical and social issues, which have only begun to address here. Suffice it to say, though, that any attempt to regulate or govern such technology systems must address these issues in addition to the more concrete technological and legal issues. The various models available to attempt this task are discussed in the following sections.

120.4 Existing Governance Mechanisms for Military Robots

At present, there are no laws or treaties specifically pertaining to restrictions or governance of military robots, unmanned platforms, or other technologies currently under consideration within the purview of this article. Instead, aspects of these new military technologies are covered piecemeal (if at all) by a patchwork of legislation pertaining to projection of force under international law, treaties or conventions pertaining to specific technologies and practices, international humanitarian law, and interpretations of existing principles of the Law of Armed Conflict (LOAC) [55].

There are, for example, multiple conventions in international law which purport to deal with specific technologies and practices, such as agreements pertaining to biological weapons [56], chemical weapons [57], certain types of ammunition [58], the hostile use of environmental modification [59], land mines [60], incendiary weapons [61], blinding laser weapons [62], and numerous others [63]. The United States is not a party to all of these conventions, and to the extent their requirements do not rise to the level of customary international law, the United States is not specifically bound by them. On the other hand, the United States has taken considerable interest in the articulation of standards which purport to regulate conduct generally on the battlefield, including how weapons are used. Thus, while no international agreements specifically regulate the use of LARs today, it is possible that such agreements might be negotiated and implemented in the future, as discussed later in this article.

In the interim, it bears mention that there are a variety of other potential existing constraints found in military doctrines, professional ethical codes, and public “watchdog” activities (as well as in international law) that might pertain to the present governance dilemma regarding military robotics. These constraints, generally, were created to address a variety of issues which are not wholly consistent with or applicable to the challenges created by the development and use of robots for military and security purposes. Yet, their existence does provide an architecture upon which to build a system of governance regarding the military use of robots on the battlefield.

As we contemplate employing this existing architecture toward the governance of military robotics, it bears noting that governance systems that are successful in obtaining compliance with a particular policy, rule, or directive share a number of important characteristics. Successful systems of “good governance” involve *clearly defined and articulated* expectations: that is, they identify the precise problems to be solved, changes to be made, or goals to be sought through governance in straightforward terms. The solutions proposed to these problems, moreover, are *realistic*: that is, they do not attempt to articulate ideal norms of what *ought* to be but rather provide feasible norms describing what *can*, in fact, be accomplished, under existing political, cultural, and legal constraints. Successful systems of governance, moreover, are *holistic and inclusive*, in the sense that all stakeholders are identified and involved in some fashion in making the rules. Finally, they issue rules or principles that are *subject to assessment*: that is, the results are capable of measurement and evaluation of effectiveness, in a manner that allows for subsequent amendment and improvement of the requirements when appropriate [64].

If these principles of good governance are not adhered to, expectations and pronouncements often go unheeded. In light of these canons of best practice for good governance, we argue that the goal of technological innovation governance should be to insure that all technological innovation is accomplished within the framework of a culture that respects the long-term effects of such work, while considering, insofar as possible, the likely ramifications of the proposed innovation and development. Appropriate governance should also insure that future end users or consumers of the specified technological innovations are aware of those

ramifications, ideally in the design phase, but at very least well before development or application of the innovations in question. All this should be accomplished, moreover, without placing too heavy of a legislative hand on, nor otherwise discouraging, the creative and competitive energies that generate much-needed innovation.

Measured against the foregoing standards, contemporary governance architecture regarding the innovation and use of military robots would appear wholly inadequate to the task. And yet, there is considerable professional, national, and international infrastructure upon which to hang a regime of articulated goals and proscriptions.

At the professional level, for example, there are multiple codes for ethical guidance regarding both best practices and limits on acceptable professional practice for a wide range of academic and professional disciplines. These are ethical codes that might conceivably find themselves applied in innovation in the field of robotics, especially for participants from professions such as engineering, computer science, biology, medicine, law, and psychology. As a general rule, these ethical codes or guidelines for professional practice are grounded in the traditional responsibilities of their individual professions and do not contemplate the challenges which can be said to presently exist for innovation generally or within the field of robotics specifically. Professions, for example, are often regulated at the state level based upon varying degrees of oversight by private organizations and societies. Those codes speak primarily to issues of the professional's relationship and responsibilities toward clients and customers, as well as toward likely competitors, and likewise address important moral and legal issues such as privacy, intellectual property, and education but often lack any concrete obligations relating to broader social responsibilities for technology development [65].

Some of these internal ethical codes also appear to contemplate the future contexts in which professionals will have to operate. For example, a "Pledge of Ethical Conduct" printed in the commencement program for the College of Engineering at the University of California, Berkley, in May 1998, reads:

I promise to work for a BETTER WORLD where science and technology are used in socially responsible ways. I will not use my EDUCATION for any purpose intended to harm human beings or the environment. Throughout my career, I will consider the ETHICAL implications of my work before I take ACTION. While the demands placed upon me may be great, I sign this declaration because I recognize that INDIVIDUAL RESPONSIBILITY is the first step on the path to PEACE [66].

To date, the most relevant initiative relating to the ethics of military technologies such as robotics is a "Code of Ethics" for robots being proposed by the Republic of South Korea (although the terms of the Code have yet to be fleshed out) [67]. The main focus of the charter appears to deal with social problems, such as human control over robots and humans becoming addicted to robot interaction (e.g., using robots as sex toys) [68]. The document will purportedly deal with legal issues, such as the protection of data acquired by robots and establishing clear identification and traceability of the machines [69].

These internal professional codes and norms are complemented by a host of nongovernmental organizations ("NGOs") which contribute to the transparency

of innovation programs, especially those performed on behalf of the State. The goals and agendas of these organizations are as varied as their names, but their methodologies generally help to educate the end user or consumer about what is being developed and what the future may portend. Such NGOs often succeed in establishing a record of evidence and impact regarding a particular thread of innovation and placing this evidence before the public and state funders (legislatures, policymakers, and appropriate government agencies) and providing news media with the expertise to report on the likely ramifications of proposed technological innovations [70]. An NGO specifically focused on promoting arms control for military robots has recently been formed, called the International Committee for Robot Arms Control ("ICRAC") [71].

At the national level in the United States, existing governance can be described as *decentralized* and, in one sense, *reactionary*. It reflects the *push* and *pull* of multiple constituencies and philosophies regarding the efficacy of support for technological innovation. U.S. federal law and regulation reflect the belief that innovation is best encouraged, on the one hand, by vigorous and unrestrained marketplace competition [72] while recognizing, on the other hand, the need for the government to organize federal funding, encourage innovation, and regulate the more egregious results of commercialization [73]. Within the United States, for example, there appears to be no urgency regarding the coordination of governance of emerging technologies such as robotics within the federal government generally, nor is there any evidence of a prevailing belief that the present governance architecture requires any type of thorough overhaul to respond to the challenges of the twenty-first century. Indeed the President's Council of Advisors on Science and Technology, in its report of April 2008 on nanotechnology, concluded:

[T]here are *no ethical concerns that are unique to nanotechnology today*. That is not to say that nanotechnology does not warrant careful *ethical* evaluation. As with all new science and technology development, all stakeholders have a shared responsibility to carefully evaluate the ethical, legal, and societal implications raised by novel science and technology developments. However, the[re is] . . . no apparent need at this time to reinvent fundamental ethical principles or fields, or to develop novel approaches to assessing societal impacts with respect to nanotechnology [74].

Turning to military uses of robotics, specifically, development and use of robots for military purposes continues to be constrained, as mentioned above, by various restrictions regarding the projection of force found in international law, as translated variously into national laws and regulations. There are, as cited above, multiple conventions which purport to deal with specific technologies and practices. Even though the United States is not a party to all of these conventions nor necessarily bound by all of them, it is nonetheless the case that the United States has taken considerable interest in the articulation of standards which purport to regulate conduct generally on the battlefield, including how weapons are used.

There are five principles which run through the language of the various humanitarian law treaties [75] (the rules) which the United States acknowledges and generally honors regarding the conduct of warfare. These are (i) a general prohibition on the employment of weapons of a nature to cause superfluous

injury or unnecessary suffering, (ii) military necessity, (iii) proportionality, (iv) discrimination, and (iv) command responsibility. These principles, as discussed below, impose ethical and arguably legal restraints on at least some uses of lethal autonomous robots.

First, some weapons, it is argued, are patently inhumane, no matter how they are used or what the intent of the user is. This principle has been recognized since at least 1907 [76], although consensus over what weapons fall within this category tends to change over time. The concept here is that some weapons are *A design-dependent*: that is, their effects are reasonably foreseeable even as they leave the laboratory. In 1996, the International Committee of the Red Cross at Montreux articulated a test to determine if a particular weapon would be the type which would foreseeably cause superfluous injury or unnecessary suffering [77]. The so-called “SIrUS” criteria would ban weapons when their use would result in:

- A specific disease, specific abnormal physiological state, specific and permanent disability, or specific disfigurement
- Field mortality of more than 25 % or a hospital mortality of more than 5 %
- Grade 3 wounds as measured by the Red Cross wound classification scale
- Effects for which there is no well-recognized and proven treatment [78]

The operative term here is *specific*; the criteria speak to technology specifically designed to accomplish more than render an adversary *hors de combat*. This test for determining weapons exclusion is a medical test and does not take into consideration the issue of military necessity. For this reason, these SIrUS criteria have been roundly criticized and rejected by the United States specifically and by the international community generally, notwithstanding support for the general principle against the use of inhumane weapons [79].

The second principle, *military necessity*, requires a different analysis. This principle “...justifies measures of regulated force not forbidden by international law which are indispensable for securing the prompt submission of the enemy, with the least possible expenditures of economic and human resources” [80]. *Military necessity* recognizes the benefit to friend and foe alike of a speedy end to hostilities. Protracted warfare, it assumes, creates more rather than less suffering for all sides. In order to determine the necessity for the use of a particular technology, then, one needs to know what the definition of victory is and how to measure the submission of the enemy in order to determine whether the technology will be *necessary* in this regard.

The third principle, *proportionality*, is of considerable concern to the developer and user of new technologies. A use of a particular technology is not *proportional* if the loss of life and damage to property incidental to attacks is excessive in relation to the concrete and direct military advantage expected to be gained [81]. In order to make this determination, it can be argued, one must consider the military necessity of a particular use and evaluate the benefits of that use in furtherance of a specific objective against the collateral damage that may be caused.

Discrimination, the fourth principle, goes to the heart of moral judgment. Indiscriminant attacks (uses) are prohibited under the rules. Indiscriminant uses occur whenever such uses are not directed against a specific military objective

or otherwise employ a method or means of combat, the effects of which cannot be directed at a specified military target (e.g., indiscriminant bombing of cities). Indiscriminate usage also encompasses any method or means of combat, the effects of which cannot be limited as required, or that are otherwise of a nature to strike military and civilian targets without distinction.

A final principle is *command responsibility*, that principle which exposes a multiple of superiors to various forms of liability for failure to act in the face of foreseeable illegal activities. This is a time-honored principle, grounded on the contract between soldiers and their superiors, which requires soldiers to act and superiors to determine when and how to act. It has a long history reflective of the need for control on the battlefield [82].

A 1997 Protocol to the Geneva Convention requires that each state party “determine whether the employment of any new weapon, means or method of warfare that it studies, develops, acquires or adopts would, in some or all circumstance, be prohibited by international law” [83]. The legal framework for this review is the international law applicable to the State, including international humanitarian law (“IHL”). In particular this consists of the treaty and customary prohibitions and restrictions on specific weapons, as well as the general IHL rules applicable to all weapons, means, and methods of warfare. General rules include the principles described above, such as protecting civilians from the indiscriminate effects of weapons and combatants from unnecessary suffering. The assessment of a weapon in light of the relevant rules will require an examination of all relevant empirical information pertinent to the weapon, such as its technical description and actual performance, and its effects on health and the environment. This is the rationale for the involvement of experts of various disciplines in the review process [84].

Once again, the United States is not a signatory to this Protocol and, thus, technically not bound by its requirements. Nonetheless, to the extent that it sets out reasonable requirements and methodologies for use by states fielding new and emerging technologies, however, this treaty could well set the standard in international law for what may be considered appropriate conduct. A final constraint worth noting is the emerging trend in international law to hold those responsible for fielding weapons which allegedly contravene the principles enunciated above through the use of litigation based on the concept of *universal jurisdiction* [85]. While litigation to date has revolved primarily around allegations of practices such as genocide, torture, rendition, and illegal interrogation, there is no reason to believe that future prosecutions may be justified where decisions regarding illegal innovation, adaptation, and use of weapons systems are made.

These various principles and requirements of international humanitarian law and ethical rules of military conduct would clearly impose some limitations on the development and use of lethal autonomous robots. However, given the ambiguous meaning and uncertain legal binding status of these principles, they are unlikely to adequately constrain and shape the development and use of LARs on their own. Additional oversight mechanisms may therefore be warranted, which are further explored in the subsequent section.

120.5 Legally Binding International Agreements

A more formal and traditional approach for oversight of a new weapons category such as LARs would be some form of binding international arms control agreement [86]. Under existing international law, there is no specific prohibition on lethal autonomous robots. In September 2009, robotics expert Noel Sharkey, physicist Jurgen Altmann, bioethicist Robert Sparrow, and philosopher Peter Asaro founded the International Committee for Robot Arms Control (“ICRAC”) to campaign for limiting lethal autonomous robots through an international agreement modeled on existing arms control agreements such as those restricting nuclear and biological weapons [87]. ICRAC called for military robots to be barred from space and that all robotic systems should be prohibited from carrying nuclear weapons [88]. The ICRAC is a small group at this time and as of yet its campaign does not yet seem to have gained the momentum necessary to spark a new international legal regime. However, there is precedent for a nongovernmental organization, the International Committee to Ban Landmines, successfully leading the charge toward banning a weapons system [89].

While ICRAC’s work has raised the issue of limiting lethal autonomous robots through an international arms control agreement, the wisdom of such a course of action is far from clear. Do explicit international legal restrictions on lethal autonomous robots make sense? Are they feasible – both from a political and a technological perspective? Does the ICRAC’s specific proposal make sense? The goal of this section is to make some preliminary points about what the options may be for international legal restrictions on lethal autonomous robots, if a policy choice is made to attempt to restrict them.

International law contains a significant number and diversity of precedents for restricting specific weapons. Existing legally binding arms control agreements and other instruments include a wide variety of different types of restrictions on targeted weapons, including prohibitions and limitations (restrictions that fall short of prohibition) on (i) acquisition, (ii) research and development, (iii) testing, (iv) deployment, (v) transfer or proliferation, and (vi) use.

These various types of prohibitions and limitations form a kind of menu from which the drafters of an international legal instrument addressing lethal autonomous robots – or other emerging warfighting technologies – could choose in accordance with their goals and the parameters of political support for such restrictions. A similar menu could be created of the various types of monitoring, verification, dispute resolution, and enforcement mechanisms that implement the prohibitions and limitations contained in existing international legal arms control instruments.

These prohibitions and limitations (as well as any accompanying monitoring/verification, dispute resolution, and enforcement provisions) can be contained in any of a number of different types of international legal instruments. They are typically contained in legally binding multilateral agreements, including in multilateral agreements primarily focused on arms control and also in the Rome Statute of the International Criminal Court. However, there are also examples of prohibitions and limitations contained in legally binding bilateral agreements, as well as examples of

prohibitions and limitations contained in legally binding resolutions of the United Nations Security Council or in customary international law (which consists of rules of law derived from the consistent conduct of States acting out of the belief that the law required them to act that way). As with the content of the restrictions and their implementing provisions, the choice of type of instrument depends on the drafters' goals and the parameters of political support for the desired restrictions and implementing provisions.

New international legal arms control instruments are typically freestanding. However, there is also at least one existing multilateral legal framework agreement with respect to which it is worth exploring whether that agreement could usefully be amended to itself provide a vehicle for some or all desired restrictions on lethal autonomous robots. This is the 1980 Convention on Prohibitions or Restrictions on the Use of Certain Conventional Weapons Which May Be Deemed to Be Excessively Injurious or to Have Indiscriminate Effects (the CCW) [90], which has been ratified by over 100 state parties [91].

The operative provisions of the CCW are contained within its protocols. The five protocols currently in force contain rules for the protection of military personnel and, particularly, civilians and civilian objects from injury or attack under various conditions by means of fragments that cannot readily be detected in the human body by x-rays (Protocol I), land mines and booby traps (amended Protocol II), incendiary weapons (Protocol III), blinding lasers (Protocol IV), and explosive remnants of war (Protocol V) [92]. It is worth noting that the case that lethal autonomous robots should be restricted by the CCW could be made most effectively if it were argued that such robots are contrary to the "principle," cited in the CCW preamble, "that prohibits the employment in armed conflicts of weapons, projectiles and material and methods of warfare of a nature to cause superfluous injury or unnecessary suffering" [93].

120.5.1 Menu of Types of Restrictions Contained in Current International Legal Arms Control Instruments

Some international legal arms control instruments prohibit a full range of activities involving the targeted weapons. For example, state parties to the Convention on the Prohibition of the Use, Stockpiling, Production and Transfer of Anti-Personnel Mines and on Their Destruction, typically referred to as the "Mine Ban Treaty," commit to not developing, producing, acquiring, retaining, stockpiling, or transferring antipersonnel land mines [94]. The following menu contains additional examples of existing international legal instruments which adopt the specified types of restrictions on a narrower basis:

120.5.1.1 Prohibitions and Limitations on Acquisition

Several international legal arms control instruments completely prohibit the acquisition of targeted weapons. For example, the Biological Weapons Convention ("BWC") prohibits all state parties from acquiring, producing, developing,

stockpiling, or retaining – and requires all state parties to within 9 months destroy or divert to peaceful purposes – (1) biological agents and toxins “of types and in quantities that have no justification for prophylactic, protective or other peaceful purposes” and (2) weapons, equipment, and delivery vehicles “designed to use such agents or toxins for hostile weapons or in armed conflict” [95]. The Convention on the Prohibition of the Development, Production, Stockpiling and Use of Chemical Weapons and on Their Destruction (“CWC”) prohibits all state parties from producing or acquiring, as well as developing, stockpiling, or retaining, chemical weapons [96].

In contrast, the Treaty on the Non-Proliferation of Nuclear Weapons (“NPT”) creates two classes of states with regard to nuclear weapons [97]. Nuclear-weapon state parties are those which had manufactured and exploded a nuclear weapon or other nuclear explosive device prior to January 1, 1967 (China, France, Russia, the United Kingdom, and the United States) [98]. The NPT does not require nuclear-weapon state parties to give up their nuclear weapons but does require those parties to “pursue negotiations in good faith on effective measures relating to cessation of the nuclear arms race at an early date and to nuclear disarmament” [99]. Non-nuclear-weapon state parties to the NPT are prohibited from receiving, manufacturing, or otherwise acquiring nuclear weapons [100].

The Inter-American Convention on Transparency in Conventional Weapons Acquisitions [101] provides a very different model, with a focus on transparency rather than prohibition of acquisitions. The convention does not prohibit any acquisitions but does require its state parties to annually report on their imports of certain specified heavy weapons, as well as submit notifications within 90 days of their incorporation of certain specified heavy weapons into their armed forces inventory, whether those weapons were imported or produced domestically [102].

120.5.1.2 Prohibitions and Limitations on Research and Development

Few, if any, international legal arms control instruments prohibit all research that could be useful for targeted weapons. Limitations on development differ from instrument to instrument. The CWC flatly prohibits the development of all chemical weapon munitions and devices [103]. In contrast, the BWC contains a more nuanced prohibition, banning the development, production, acquisition, and retention of (1) microbial or other biological agents or toxins “of types and in quantities that have no justification for prophylactic, protective or other peaceful purposes” and (2) weapons, equipment, or means of delivery “designed to use such agents or toxins for hostile purposes or in armed conflict” [104]. It is important to note that restrictions based on quantities or intended use rather than the underlying nature of the technology can be exceptionally difficult to verify, at least without highly intrusive inspections.

120.5.1.3 Prohibitions and Limitations on Testing

Prohibitions and limitations on testing of targeted weapons are most prominent in the nuclear-weapon context. For example, the Comprehensive Test Ban Treaty (“CTBT”), which has not yet entered into force, prohibits “any nuclear weapon test

explosion or any other nuclear explosion” [105]. The CTBT’s entry into force awaits ratification by nine key countries, including the United States [106]. In contrast, the 1963 Treaty Banning Nuclear Weapon Tests in the Atmosphere, in Outer Space and Under Water (also known as the “Limited Test Ban Treaty”) – which unlike the CTBT is in force – specifically prohibits nuclear weapon tests “or any other nuclear explosion” only in the atmosphere, in outer space, and underwater [107]. The Limited Test Ban Treaty also prohibits nuclear explosions in all other environments, including underground, if they cause “radioactive debris to be present outside the territorial limits of the State under whose jurisdiction or control” the explosions were conducted [108].

120.5.1.4 Prohibitions and Limitations on Deployment

Some international legal arms control instruments focus on limiting deployment of the targeted weapons, such as with overall or regional numerical caps. For example, the Strategic Offensive Reductions Treaty, entered into by the United States and Russia in 2002, requires the two countries to reduce their operationally deployed strategic nuclear forces to between 1,700 and 2,200 warheads by December 31, 2012 [109]. The Conventional Armed Forces in Europe Treaty, ratified by the United States in 1992, contains bloc and regional limits on deployment of certain weapons [110].

120.5.1.5 Prohibitions and Limitations on Transfer/Proliferation

Many international legal arms control instruments include prohibitions or limitations on transfer or other proliferation of the targeted weapons. For example, the NPT prohibits parties that possess nuclear weapons from transferring the weapons to any recipient as well as from assisting, encouraging, or inducing any non-nuclear-weapon state to manufacture or otherwise acquire such weapons in any way [111].

The CWC bans the direct or indirect transfer of chemical weapons [112]. The CWC also bans assisting, encouraging, or inducing anyone to engage in CWC-prohibited activity [113]. Similarly, the BWC bans the transfer to any recipient, directly or indirectly, and assisting any state, group of states, or international organizations to manufacture or otherwise acquire (1) biological agents and toxins “of types and in quantities that have no justification for prophylactic, protective or other peaceful purposes” and (2) weapons, equipment, and delivery vehicles “designed to use such agents or toxins for hostile weapons or in armed conflict” [114]. In contrast, the Inter-American Convention on Transparency in Conventional Weapons Acquisitions does not prohibit exports but does require its state parties to annually report on their exports of certain specified heavy weapons [115].

120.5.1.6 Prohibitions and Limitations on Use

Several international legal arms control instruments include prohibitions or limitations on use of the targeted weapons. The International Court of Justice, in its 1996 advisory opinion on the Legality of the Threat or Use of Nuclear Weapons, ruled that “the threat or use of nuclear weapons would generally be contrary to the rules

of international law applicable in armed conflict, and in particular the principles and rules of humanitarian law; However, in view of the current state of international law, and of the elements of fact at its disposal, the Court cannot conclude definitively whether the threat or use of nuclear weapons would be lawful or unlawful in an extreme circumstance of self-defence, in which the very survival of a State would be at stake” [116]. The Rome Statute of the International Criminal Court prohibits (1) employing poison or poisoned weapons, (2) employing poisonous gases, and (3) employing bullets which flatten or expand easily in the human body [117]. This list is potentially expandable. While the CWC bans chemical weapons use or military preparation for use [118], the BWC does not ban the use of biological and toxin weapons but reaffirms the 1925 Geneva Protocol, which prohibits such use [119].

Protocol IV of the CCW prohibits the use of lasers specifically designed to cause permanent blindness [120]. It further obliges state parties to make every effort to avoid causing permanent blindness through the use of other lasers [121]. While prohibiting the use of blinding lasers, the convention does not rule out their development or stockpiling [122]. However, it does outlaw any trade in such arms [123].

120.5.2 Cautionary Note Regarding the Utility of International Legal Arms Control Instruments

It is worth noting that even the broadest and most aggressively implemented international legal arms control instruments suffer from certain inherent weaknesses. For example, existing international legal arms control instruments only apply to states. Their impact on non-state actors is at best indirect (e.g., the CWC and BWC require state parties to prohibit activities on their territory that are prohibited directly for them). Yet non-state actors, particularly transnational terrorist groups, may present a significant threat of utilizing lethal autonomous robots. In addition, these international legal arms control instruments typically require state consent – states can choose not to ratify these agreements and can withdraw from them if they do join [124].

120.6 Soft Law/Governance Approaches

There have been a number of proposed strategies for managing the risks and regulating the uses of emerging military technologies [125]. Those strategies vary in formality and scope. Proposed measures range from formal, binding agreements such as treaties to informal initiatives such as codes of conduct [126]. Oversight of lethal autonomous robots likely falls toward the latter end of this spectrum at least initially, in that some form of coordinated international oversight is warranted, but formal, “hard law” treaties may not be practicable, at least in the short term.

In a growing number of areas of international oversight, ranging from environmental to commercial to social to military issues, traditional “hard law”

treaties and agreements are being supplemented or in some cases displaced by new “soft law” approaches [127]. “Soft law” approaches seek to create and implement substantive principles or norms without creating enforceable legal requirements. The traditional model of binding international regulation that relies on formal treaties negotiated by government officials has a number of limitations, including the excessive resources and time needed to negotiate a formal international agreement, problems in enforcement of and compliance with such agreements, and the lack of flexibility and responsiveness in adapting such instruments to changing circumstances [128].

Many new models of international oversight or harmonization have been developed to circumvent such problems. These new models tend to be more flexible and reflexive, capable of being launched relatively quickly and adapted easily to changing technological, political, and security landscapes. These new soft law approaches have their own limitations, including perhaps most importantly that they are not as binding and often specific as traditional legal agreements. Yet, their growing popularity is due to advantages such as the relative ease by which they can be adopted and updated and the broader roles they create for stakeholders to participate in their substantive formulation [129].

Some of the key soft law/governance approaches that have been applied in other areas of emerging technologies that could conceivably be adapted to military use of robotics are briefly described below.

120.6.1 Codes of Conduct

Codes of conduct are nonbinding and often somewhat general guidelines defining responsible, ethical behavior and which are intended to promote a culture of responsibility. They can be developed and implemented by a variety of different entities, including governmental agencies, industry groups, individual companies, professional or scientific societies, nongovernmental organizations, or collaborative partnerships involving two or more of these entities.

One of the first, and possibly most successful, codes of conduct was at the outset of the field of genetic engineering. The Asilomar guidelines on recombinant DNA research were adopted in the 1970s in response to safety concerns about some early genetic engineering experiments [130]. These guidelines were initially developed by scientists based on discussions at a 1975 conference held at Asilomar, California, were subsequently adopted into more binding guidelines (at least for funding recipients) by the National Institutes of Health in 1976, and were widely complied with by scientists around the world [131]. More recently, codes of conduct have emerged at the forefront of discussions to restrict the use of genetic engineering to create new biological weapons [132]. Although there are concerns that unenforceable codes of conduct will not provide strong enough assurances against the creation of new genetically engineered biological weapons, they may play an important bridging role in providing some initial protection and governance until more formal legal instruments can be negotiated and implemented [133].

In the same way, codes of conduct may play a similar transitional role in establishing agreed-upon principles for the military use of robots.

Codes of conduct are being created for other emerging technologies with potential military applications. The areas of synthetic biology and nanotechnology are two examples. In synthetic biology three different groups have recently proposed competing codes of conduct to manage security implications [134]. The groups are the U.S. Government, the International Association Synthetic Biology (IASB), and the International Gene Synthesis Consortium (IGSC) [135]. This proliferation of competing codes flags a key question about who has the authority and influence to promulgate effective codes of conduct that relevant parties will comply with.

A number of codes of conduct have also been created in the field of nanotechnology. The first code was developed by the Foresight Institute in the form of “guidelines,” with a primary objective of discouraging the creation and deployment of autonomous replication nanosystems [136]. The Foresight Institute guidelines have since been updated six times [137], demonstrating the flexibility and adaptivity that is possible with codes of conduct which can be relatively easily updated (at least compared with a treaty or other more formal instrument). The current Foresight guidelines are extremely thorough and address issues and implications of nanotechnology in professional, industry, military, health, policy, and other contexts [138]. The guidelines are based on the premise that professional ethics and soft law measures can be at least as effective as hard law in promoting safe practices [139]. The drafters of the Foresight guidelines also recognize the value in promoting the least restrictive legal alternative while developing good practices in areas of emerging technology [140].

The European Union recently adopted a code of conduct for nanotechnology researchers [141]. The code promotes a responsible and open approach to research conducted within a safe, ethical, and effective framework [142]. Regular monitoring and revision of the code will occur in order to keep the code current with advances in nanoscience and nanotechnology [143].

Another nanotechnology code of conduct originating in Europe, but with international applicability, is the Responsible NanoCode [144]. The Responsible Nanocode is an example of a code of conduct developed as a result of significant collaboration and designed to reach a wide target audience. The creators are United Kingdom’s Royal Society, a nanotechnology industry trade group and a public interest organization [145]. The code is for companies that handle nanomaterials, and the specific objective of the code is to “establish a consensus of good practice in the research, production, retail and disposal of products using nanotechnologies” [146]. The code was developed to be universally applicable; it was devised “to be adopted by organisations in any part of the world, under any regulatory regime” [147].

Much of the discourse concerning codes of conduct tend to refer to codes of conduct as a single concept with a singular meaning or interpretation. But codes of conduct can exist on a continuum with respect to their objectives, specificity, audience, and expectations of compliance. The goal of all codes is to affect behavior, but different types of codes seek to shape behavior in different ways [148].

There are actually three primary types of codes: codes of ethics, codes of conduct, and codes of practice [149]. Codes of ethics entail professionalism; codes of conduct espouse guidelines of appropriate behavior; codes of practice embody practices to be enforced [150]. Many commonly referred to codes of conduct may, in reality, be a combination of two or even all three types of codes.

The principal benefit of codes of conduct may not be the codes themselves but rather the educational and cooperation-building effects of working on developing a code [151]. So, notwithstanding the ultimate utility and efficacy (or lack thereof) of a code, the production of the code itself may have expressive value. The discussion and collaboration required to develop a code raises awareness of relevant issues and prompts dialogue between relevant parties [152]. It is important that individuals, institutions, and countries are aware of their ethical obligations [153]. Once a code is created, it can help achieve that end by being a valuable educational tool [154]. It is critical that countries are aware of other countries' intentions and limits when it comes to the use of autonomous robots, and so it is reasonable to expect that the exercise of attempting to develop a code of conduct addressing these issues could have both utility and educational benefits.

Even though their tangible outcomes may be hard to identify, measure, and quantify, codes of conduct possess unique features which make them attractive informal measures. The multiple codes of conduct that exist in the synthetic biology and nanotechnology industries highlight some of the salient benefits and drawbacks of codes of conduct. In terms of benefits, they can be created rather quickly – compared to the time it would take to develop formal legal regulations – they can be drafted by interested parties who are knowledgeable in the area, and they can be customized to address unique properties of a technology. On the other hand, there can be difficulties in application. Because anyone can craft a code of conduct, when multiple codes are introduced into an area, whose code takes precedent? There is no hierarchical relationship among codes of conduct that would provide a clear sense of priority. The strengths of codes of conduct and other soft law mechanisms – being voluntary, cooperative, flexible measures – come at a price: they have no rank order; they are all on the same playing field.

Despite sounding like a straightforward concept, it is not a simple process to create a code of conduct. Drafting a comprehensive, appropriate, and effective code requires a thorough consideration of a myriad of issues, attention to detail, as well as a proper balancing of the policy interests of interested parties [155]. While a code of conduct will not likely be sufficient to ensure the appropriate and ethical use of autonomous military robots, the process of creating and disseminating such a code is undeniably a step forward and can be an important piece or at least starting point of eventual treaty or agreement in this area [156].

120.6.2 Transgovernmental Dialogue

“Transgovernmental dialogue” refers to a growing number of informal and flexible arrangements under which governmental officials from different countries meet

on a regular basis to discuss and coordinate policies. These opportunities for discussion provide a forum to share information and “best practices,” to seek to harmonize policies and oversight mechanisms, to coordinate enforcement practices, and to help anticipate, prevent, and resolve international disputes [157]. Through these practices, transgovernmental dialogue can greatly enhance cooperation and influence policy outcomes. It achieves that end through collaborative mechanisms and countries’ shared desire to address a common problem or goal. These types of dialogues are beneficial to the nations involved and are becoming increasingly common in areas requiring international coordination, with national security issues being a prime example [158]. They offer “a structure that is less threatening to democratic governance than private transnational action and less costly than interstate negotiations, yet they can lay a firm foundation for harmonized national regulation and even, if appropriate, for international regulation” [159].

An example of a transgovernmental dialogue in the national security context is the Australia Group, which is an informal forum of officials from 41 nations with a common interest in preventing proliferation of materials that could be used for chemical or biological weapons [160]. Arising out of the experiences of the Iraq/Iran war of the late 1980s, the Australia Group was chiefly concerned with the use of chemical and biological weapons deployed in that conflict, but the Group has subsequently developed a list of dual use [161] items that each country agrees to control through national export regulations. Achieving the goals set forth by the Australia Group depends completely on the voluntary good-faith commitment of the individual countries to the Group’s goals because member countries do not have any legally binding obligations [162]. The group meets annually to discuss ways to prevent proliferation of chemical and biological agents through national export licensing policies and other measures.

Another example of an existing transgovernmental institution is the *International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH)*, which brings together pharmaceutical regulators from the United States, Europe, and Japan, along with pharmaceutical industry representatives from the same three jurisdictions, to coordinate pharmaceutical regulatory policy issues with a view to harmonization [163]. In addition to increasing harmonization, another goal of the ICH is to reduce the need for duplication of testing products, an accomplishment which is intended to reduce delays in the development and distribution of new medicines around the world [164].

A third example is the Organization for Economic Co-operation and Development (OECD), an organization of 30 industrialized countries that has created 2 committees (the Working Party on Manufactured Nanomaterials (WPMN) and the Working Party on Nanotechnology (WPN)) to undertake a variety of informal harmonization activities [165]. The objective of the WPN is to promote international cooperation that fosters the research, development, and responsible commercialization of nanotechnology [166]. The WPN facilitates communication between governments which promotes discussion, awareness, and ideally a coordination of policy responses [167]. Meanwhile, the WPMN is international effort to analyze the environmental health and safety risks posed by nanotechnology [168].

Another example from the nanotechnology realm is the International Dialogue on Responsible Research and Development of Nanotechnology, a forum that has brought together regulators from almost 50 nations every 2 years (2004, 2006, 2008) to discuss nanotechnology regulation [169]. The initial meeting of this forum was sponsored by an NGO (the Meridian Institute), but the national governments volunteered to sponsor subsequent meetings (Japan in 2006 and the EU in 2008).

These various examples of transnational dialogue are effective in starting a discussion among policymakers from different nations and are relatively easy and quick to implement. The only major requirement is a sponsoring organization or nation to convene at least the initial meeting. While not usually creating any binding international policies themselves, such dialogue initiatives can be an important first step to more concrete policy measures in the future and may serve a useful function in that regard to start the dialogue among government policymakers on international policies for military robots.

120.6.3 Information-Sharing and Confidence-Building Measures

Various types of information-sharing and other confidence-building measures have been used to enhance stability, trust, and security for a variety of problems, including in the military and national security contexts. Information-sharing and confidence-building measures (“CBMs”) can be adopted unilaterally or can be coordinated or negotiated among several parties. The concept of CBMs arose in the sphere of international relations, and such measures are frequently used in international conflicts as the initial steps for reducing hostilities between enemies. In this international context, CBMs usually involve some mix of communication, constraint, transparency, or verification measures. Information-sharing mechanisms are one type of CBM. An example of legally implemented information-sharing mechanism is the Biosafety Clearinghouse adopted under the Cartagena Protocol to the UN Convention on Biodiversity, which creates a web-based portal that provides a forum for nations to share information on scientific, risk, regulatory, legal, and ethical determinations from each nation [170].

One could envision a variety of potential CBDs for military robotics, consisting of either unilateral or multilateral initiatives. For example, individual or groups of nations could commit to a limited moratorium on the deployment of such systems. Nations could share information on technical issues with regard to lethal autonomous robots, including issues relating to potential compliance and verification of any future international agreement concerning LARs. Nations could even take relatively minor but helpful steps by convening an international conference to discuss the issue of LARs.

120.6.4 Framework Convention

A final potential soft law mechanism, which is really more of a hybrid between soft law and the more formal international agreements discussed in the previous section,

is a framework agreement or convention. A framework convention, at its most basic level, creates a process and an institutional basis for gradually developing a more substantive international agreement. It consists of an initial agreement to create the framework agreement, often with little or no substantive “teeth” originally, but which sets in place an annual meeting of representatives of the participating nations, perhaps a small secretariat to manage the process, and provisions for negotiating and adopting more substantive protocols that member nations can then consider ratifying on an individual basis. For example, the Vienna Convention for the Protection of the Ozone Layer began as a rather weak agreement but was gradually strengthened by various additions and protocols over time such as the Montreal Protocol and its amendments. Other prominent examples of such framework agreements include the Framework Convention on Climate Change and the Framework Convention on Tobacco.

While the lack of substantive content at the outset may seem a weakness of such agreements, it has the benefit of thereby being nonthreatening to uncertain or recalcitrant states who may otherwise be reluctant to sign onto any international agreement because of major scientific or technological uncertainties or differences in perspective or distrust of other participating states. Once the nations agree to participate in the framework agreement, the process that is created can be the vehicle for greater convergence of positions and building of trust going forward, opening the door for more substantive commitments on an incremental basis. Thus, a framework agreement provides several potential benefits to states, including to:

- Acknowledge that a problem may exist, legitimating it as an international concern.
- Draw the attention of relevant experts, interest groups, and the public to the problem.
- Commit themselves to take, or at least consider, more substantive action if the problem proves to be sufficiently serious.
- Demonstrate that they are taking the issue seriously [171].

In addition to these benefits, the framework agreement can sometimes involve more concrete steps, such as the establishment of technical or scientific committees that can provide important for a for understanding and cooperating on such issues, and also potentially the inclusion of reporting and other information-sharing mechanisms that can improve transparency on an issue. All of these mechanisms and benefits would potentially be timely and useful for addressing LARs.

120.7 Conclusion

Many years ago, David Collingridge recognized a fundamental tension in the governance of new technologies – prior to the development and deployment of the technology, not enough is known about the potential risks of that technology to warrant or guide any restrictions or limitations, whereas once the technology has been developed and deployed, it is often too late to undertake meaningful regulations because the commercial momentum behind the technology is now too strong and entrenched to sway [172]. Lethal autonomous robotics offers an opportunity to

break this cycle. Enough concern and information exists now to consider appropriate governance models in a timely and proactive manner. Yet, the time to take action is short before the current window of opportunity to design a relevant governance or oversight system for LARs closes.

The need to take action on LARs is therefore urgent and timely. This article has not tried to prescribe a specific action or form of oversight but rather has sought to identify the range of possible governance mechanisms that can be brought to bear on this problem. This can range from ethical principles implemented through modifications or refinements of national policies, law of war and rules of engagement, international treaties or agreements, or a variety of other “soft law” governance mechanisms. Of course, any governance approach need not be restricted to any one of these modes of oversight but could involve a combination of two or more approaches. The critical point is that discussions on these options, at both national and international levels, must advance promptly if we are to successfully manage the technological convergence of rapidly advancing robotics technology with military needs, incentives, and temptations.

References

1. See generally Ronald C. Arkin, *Governing Lethal Behavior in Autonomous Robots* (2009)
2. Patrick Worsnip, *U.N. Official Calls for Study of Ethics, Legality of Unmanned Weapons*, Wash. Post, Oct. 24, 2010
3. The Statement of the 2010 Expert Workshop on Limiting Armed Tele-Operated and Autonomous Systems, Berlin, 22nd September, 2010, available at <http://www.icrac.co.cc/Expert%20Workshop%20Statement.pdf> (last visited November 14, 2010)
4. P.W. Singer, *Military Robots and the Laws of War*, *The New Atlantis*, Winter 2009, at 25, 43
5. Material from this section is derived with permission from Arkin, *supra* note 2
6. See generally Peter W. Singer, *Wired for War* (2009); Peter Bergen & Katherine Tiedemann, *Revenge of the Drones: An Analysis of Drone Strikes in Pakistan*, New America Foundation, October 19, 2009, available at http://www.newamerica.net/publications/policy/revenge_of_the_drones (last Visited November 14, 2010)
7. Thomas K. Adams, *Future Warfare and the Decline of Human Decisionmaking*, *Parameters*, U.S. Army War College Quarterly, Winter 2001–02, at 57–58
8. Arkin, *supra* note 2, at 7–10; George Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control* (2005); Robert Sparrow, *Building a Better WarBot: Ethical Issues in the Design of Unmanned Systems for Military Applications*, 15 *Sci. Eng. Ethics* 169, 173–74 (2009) [hereinafter Sparrow, *Building a Better Warbots*]
9. U.S. Navy, “Phalanx Close-in Weapons Systems”, United States Navy Factfile, available at <http://www.navy.mil/navydata/fact.display.asp?cid=2100&tid=800&ct=2> (last visited February 12, 2010)
10. Anti-personnel mines have been banned by the Ottawa Treaty on antipersonnel mines, although the U.S., China, Russia, and 34 other nations are currently not party to that agreement. Convention on the Prohibition of the Use, Stockpiling, Production and Transfer of Anti-Personnel Mines and on their Destruction (Ottawa Treaty), Sept. 18, 1997, 2056 U.N.T.S. 211. Recent developments, however, indicate that the U.S. is evaluating whether to be a part of the Ottawa Treaty. See Mark Landler, *White House Is Being Pressed to Reverse Course and Join Land Mine Ban*, *N.Y. Times*, May 8, 2010, at A9
11. Adams, *supra* note 8, at 57–58

12. U.S. Department of Defense, DOD Unmanned Systems Roadmap: 2007–2032 (2007), available at <http://www.fas.org/irp/program/collect/usroadmap2007.pdf> (last visited Sept. 21, 2010)
13. Adams, *supra* note 8, at 57
14. See Arkin, *supra* note 2, at 10
15. Foster-Miller Inc., Products & Service: TALON Military Robots, EOD, SWORDS, and Hazmat Robots (2008), available at <http://www.foster-miller.com/lemming.htm>, 2008 (last visited February 10, 2010)
16. QinetiQ, Press Release: QinetiQ North America Ships First MAARS Robot, June 5, 2008, available at http://www.qinetiq.com/home/newsroom/news_releases.homepage/2008/2nd_quarter/qinetiq_north_america0.html (last visited February 12, 2010)
17. Rebecca Maksel, *Predators and Dragons*, Air & Space Magazine, July 1, 2008, available at http://www.airspacemag.com/history-of-flight/Predators_and_Dragons.html (last visited September 21, 2010)
18. Arkin, *supra* note 2, at 29–30
19. Michael Walzer, *Just and Unjust Wars* 251 (4th ed., 1977)
20. Scott D. Sagan, *Rules of Engagement*, in *Avoiding War: Problems of Crisis Management* (ed., A. George) (1991), pp. 443–70
21. DARPA (Defense Advanced Research Projects Agency), Broad Agency Announcement 07-52, *Scalable Network Monitoring*, Strategic Technology Office, August 2007, available at <https://www.fbo.gov/index?s=opportunity&mode=form&tab=core&id=b524ff8d8f7390061d4c5d5444c9e620&tab=documents&tabmode=list> (last visited Sept. 22, 2010).
22. Adams, *supra* note 8, at 58
23. Robert Sparrow, *Killer Robots*, 24 *Journal of Applied Philosophy* 66 (2007) [hereinafter Sparrow, *Killer Robots*]
24. *Ibid.* at 69–73.
25. Peter Asaro, *Robots and Responsibility from a Legal Perspective*, Proceedings of the IEEE 2007 International Conference on Robotics and Automation, Workshop on RoboEthics, April 14, 2007, Rome, Italy
26. Sparrow, *Building a Better WarBot*, *supra* note 9, at 172
27. *Ibid.* at 172–73
28. For example, the TALON robot has been cited by its manufacturer for its extensive use in military operations in Afghanistan and Iraq. It is small and is capable for a variety of uses including the ability to deliver weapons fire. The manufacture’s web site specifically states that “TALON’s multi-mission family of robots includes one specifically equipped for tactical scenarios frequently encountered by police SWAT units and MPs in all branches of the military. TALON SWAT/MP is a tactical robot that can be configured with a loudspeaker and audio receiver, night vision and thermal cameras and a choice of weapons for a lethal or less-than-lethal response”. Available at <http://www.foster-miller.com/lemming.htm> (last visited Feb. 28, 2010)
29. The Law of War in conjunction with the laying of contact underwater mines is covered in Article I to the Hague Convention VIII; October 18, 1907. Article I states: “It is forbidden — 1. To lay unanchored automatic contact mines, except when they are so constructed as to become harmless one hour at most after the person who laid them ceases to control them; To lay anchored automatic contact mines which do not become harmless as soon as they have broken loose from their moorings; 3. To use torpedoes which do not become harmless when they have missed their mark.” Convention (VIII) Relative to the Laying of Automatic Submarine Contact Mines, Oct. 18, 1907, 36 U.S.T. 541
30. Singer, *supra* note 7, at 81
31. Wendell Wallach & Colin Allen, *Moral Machines: Teaching Robots Right from Wrong* 189–214 (2009)
32. Noah Shachtman, *Robot Cannon Kills 9, Wounds 14*, *Wired.com*, October 18, 2007, available at <http://blog.wired.com/defense/2007/10/robot-cannon-ki.html> (last visited Feb. 8, 2010)

33. Roger Clarke, *Asimov's Laws of Robotics: Implications for Information Technology-Part II*, 27 Computer 57, 65 (1994)
34. Blay Whitby, *Computing Machinery and Morality*, 22 AI & Society 551, 551–63 (2008)
35. Issac Asimov, I, Robot (1950)
36. See e.g., Ray Kurzweil. The Age of Spiritual Machines: When Computers Exceed Human Intelligence (1999); Ray Kurzweil. The Singularity is Near: When Humans Transcend Biology (2005)
37. Noel Sharkey, *Cassandra or False Prophet of Doom: AI Robots and War*, 23(4) IEEE Intelligent Systems 14, 14–17 (2008); Noel Sharkey, *The Ethical Frontiers of Robotics*, 32 Science 1800, 1800–01 (2008)
38. Sparrow, *Building a Better WarBot*, *supra* note 9, at 178
39. Peter Asaro, *How Just Could a Robot War Be?*, in Adam Briggie, Katinka Waelbers, and Philip Brey (eds.), *Current Issues in Computing and Philosophy* 1, 7–9 (2008)
40. Robert Sparrow, *Predators or Plowshares? Arms Control of Robotic Weapons*, IEEE Tech. Soc'y Magazine, Spring 2009, at 25, 26 (hereinafter "*Sparrow, Predators or Plowshares?*")
41. Singer, *supra* note 7, at 299
42. *Ibid.* at 37–38
43. Anne Broache, *Police Agencies Push for Drone Sky Patrols*, CNET News, August 9, 2007, available at http://news.cnet.com/Police-agencies-push-for-drone-sky-patrols/2100-11397_3-6201789.html (last visited Feb. 12, 2010)
44. See generally Gary Marchant and Lyn Gulley, *National Security Neuroscience and the Reverse Dual-Use Dilemma*, 1 Am. J. Bioethics Neuroscience 20, 20–22 (2010)
45. This consideration led the Lincoln Center for Applied Ethics at Arizona State University, the Inamori International Center for Ethics and Excellence at Case Western Reserve University, and the U. S. Naval Academy Stockdale Center for Ethical Leadership to found the Consortium for Emerging Technology, Military Operations, and National Security, or CETMONS. See <http://cetmons.org>, last visited June 17, 2010. See generally also Max Boot, *War Made New* (2006); John Keegan, *A History of Warfare* (1993)
46. See, e.g., www.darpa.mil/grandchallenge/index.asp, last visited June 17, 2010, detailing the progress in autonomous vehicles as a result of DARPA's Grand Challenge initiative
47. See, e.g.,

The Moral Case Against Autonomous and Semiautonomous UAVs

121

Noel Sharkey

Contents

| | | |
|-------|---|------|
| 121.1 | Introduction | 2920 |
| 121.2 | Armed “Man-in-the-Loop” Systems | 2920 |
| 121.3 | Moral Disengagement | 2921 |
| 121.4 | Targeted Killings | 2922 |
| 121.5 | From <i>in</i> the Loop to <i>on</i> the Loop to <i>Out of</i> the Loop | 2926 |
| 121.6 | Some Lessons to Learn | 2930 |
| | References | 2931 |

Abstract

Robots in warfare are an inevitable stage in the evolution of weapon development designed to separate fighters from their foes. Weapon technology has evolved to enable killing from ever-increasing distances throughout the history of warfare. From stones to pole weapons to bows and arrow cannon to aerial bombing to cruise missiles, killing has become ever easier. Battlefield robots may not change the character of war, but they will change the way that wars are fought. The focus of this chapter accordingly is on ethical concerns about the application of armed robots in areas with mixed combatant and civilian populations. It is unclear whether international humanitarian law (the Laws of War) will eventually need to be amended to accommodate emerging technologies. But for now one must scrutinize the mapping between the applications of the new technology and the current laws to ensure that those laws are preserved and followed.

From the Journal of Military Ethics, vol. 9, no. 4 (2010); reprinted by permission of the publisher (Taylor and Francis, Ltd., <http://www.tandf.co.uk/journals>).

N. Sharkey

Department of Computer Science, University of Sheffield, Sheffield, UK

e-mail: noel@dcs.shef.ac.uk; robot@blueyonder.com

121.1 Introduction

In 2010, the ultimate distance weapons in the sights of military organizations throughout the world are armed robots. To be clear, the robots discussed here are mobile platforms that can carry multiple weapon systems, bombs, cameras, sensors, or other payloads. They can be recalled, refueled, and re-weaponized. Operationally, they can be sent into the theatre on a fishing expedition without a preset goal. It is this type of flexibility that separates them from other robots such as cruise missiles and torpedoes. Of course there are some exceptional systems such as loitering munitions that fall between the cracks. For example, the Israeli Harpy and Harop made by IAI are a cross between a missile and an unmanned aerial vehicle (UAV). They are launched like a missile but hover over an area and select their own targets (hostile radar signatures) as they arise and then impact them like a missile. Thus they have some mission flexibility but not reusability.

Many thousands of robots have been used during both the Iraq and Afghanistan conflicts (up from 150 in 2004) for tasks such as bomb disposal and cave clearance on the ground and for surveillance from the air. The undisputed success of UAVs (now more commonly known remotely piloted aircraft (RPA) or drones) for gathering intelligence in the Iraq/Afghanistan conflicts has made them a showcase for military powers everywhere. There is an almost insatiable military demand for UAVs. Troops do not like to move without them. This same demand is reflected in more than 40 other countries that are either developing or purchasing military robots. These include Russia, China, Israel, Pakistan, Korea, India, and Iran.

The main ethical concerns expressed here are about the planned use of the armed autonomous “decision-making” robots that have appeared in all of the U.S. military’s road maps since 2004. However, before examining the issues surrounding autonomous lethal force in detail, it will be useful to look at some of the ethical problems that have arisen for man-in-the-loop or remotely piloted drones. There are lessons to be learned that will highlight some of the dangers of fully autonomous armed robots.

121.2 Armed “Man-in-the-Loop” Systems

The USA leads the field with its armed drones. The Predator MQ-1 was first armed by the CIA in 2001 with its now standard two hellfire missiles. Nowadays “pilots” of the 432nd Air Expeditionary Wing fly official military missions over Iraq and Afghanistan from Creech Air Force Base in the Nevada desert, thousands of miles away from the action. The Royal Air Force 39th Squadron also participates at Creech. Each plane has a two person crew: a remote pilot and a sensor operator who sit at what looks like game consoles making decisions about the application of lethal force. The planes are flown around the clock in shifts, and it is easy for pilots to take a break from “battle” at any time or even go home for dinner with their families.

In October 2007, the Predator fleet was joined by the larger and more powerful MQ-9 Reaper with its larger payload capacity of 1,700 kg – up to 14 hellfire missiles or a mixture of missiles and bombs. The number of Reapers flying in operations

doubled after their first year, and General Atomics have reported difficulties in keeping up with the demand. By March 2009, the number of armed flying drones in the field was reported to be 195 Predators and 28 Reapers. The number has grown dramatically since.

Anticipated increases can be seen in the training figures for Predator crews. In 2005, 32-person crews consisting of a pilot and sensor operator were being trained per year, whereas in 2008, 160 crews were trained, and in late 2008, an additional \$412m was added to the budget for training more non-pilot operators. In 2009, the number of remote pilot operators trained outnumbered the number of conventional pilots. Figures for the size of the current fleet of Predators and Reapers were not publicly available at the time of writing, but in May 2009 the Obama administration budgeted \$5.5 billion to add to the fleet in 2010. The military demand for the armed UAVs is highlighted by the more than a million mission hours flown by the MQ-1 by 2010 (up from 250,000 in June 2007).

121.3 Moral Disengagement

Flying a UAV from thousands of miles away from the battlefield clearly alleviates one of the two fundamental obstacles that warfighters must face – fear of being killed. As Daddis (2004) pointed out, fear is one of the greatest obstacles to a soldier's effectiveness in battle. It is obvious that the greater the distance from the enemy, the less fear will play in the action. Many battles throughout history have been lost by men bolting in panic as fear swept through the ranks (Holmes 2003).

Certainly the remote pilots of attack planes like the Reaper and the Predator have no need to worry about their personal safety. Sitting in cubicles thousands of miles away from the action, they can give a mission their full attention without worrying about being shot at. They are in highly secure home ground where no pilot has ever been safer. The second fundamental obstacle is *resistance to killing*. Grossman (1995) argues that “not firing” is not cowardice but really a compulsion of ordinary men not to kill. In his book, *Acts of War*, Holmes (ibid.) argues that the *hit rates* in a number of historical battles show that many soldiers were not prepared to fire directly at the enemy. This is also borne out by the analyses and findings of Marshall (1947) from his interviews of soldiers after WWII.

Similarly, Royakkers and van Est (2010) discuss the emotional and moral disengagement of UAV pilots, or “cubicle warriors” as he calls them, from the consequences of their actions in the field. The crux is that the type of control system used encourages a “PlayStation” mentality that makes an operator more careless about decisions to kill. Royakkers and van Est suggest that new recruits may have been playing video games for many years and may not see a huge contrast with that of being a cubicle warrior. He provides examples from Peter Singers book *Wired for War* in which a young pilot is reported to have said, “It’s like a video game. It can get a little bloodthirsty. But it’s fucking cool” (Singer 2009, 308–309). And in a further example, a young pilot says, “The truth is, it wasn’t all I thought it was cracked up to be. I mean, I thought killing somebody would be this life-changing experience. And then I did it, and I was like ‘All right, whatever.’ . . . Killing people

is like squashing an ant. I mean, you kill somebody and it's like 'All right, let's go get some pizza.'" (Singer 2009, 391–392).

In contrast to this view, there are reports of remote pilots suffering from different types of stress than onboard pilots. After an attack the operator gets to see the aftermath of using high-resolution cameras. This is more than a conventional fighter pilot ever sees, and, at least on the surface, it would seem that remote operators would be less morally disengaged than the crew of a high-altitude bomber.

However, for an article appearing in the *Air Force Times* (Lindlaw 2008), five remote pilots interviewed said that they had not been particularly troubled by their missions although they can sometimes make for a strange existence. Col. Chris Chambliss, a commander of the 432nd Wing at Creech, said that only on four or five occasions had sensor operators gone to see the Chaplain or their supervisors and that this was a very small proportion of the total number of remote operators.

All of this is anecdotal evidence and proper scientific research needs to be conducted to find out what kind of stresses, if any, the operators are facing. It is well known that making good decisions is more difficult under stressful conditions. Longitudinal studies would be required as the operators may start out being stressed by graphic scenes but then become inured with sufficient exposure. This is bit like the process trainee surgeons and nurses go through. They are known to often faint when they start out but quickly habituate to horrific medical events.

Killing distance does not have to be physical distance. It can be psychological as well. Grossman (ibid.) cites Clausewitz and du Picq for expounding at length that the vast majority of deaths in battle occurred when the victors chased the losing side in retreat. Du Picq suggests that Alexander the Great lost fewer than 700 men over all his battles because his army was never pursued by a victorious enemy. Although UAV operators have high-resolution images on large screens, the cameras are looking down from above and it can be very hard to make out the faces of their victims. Is this so different from attacking retreating troops?

One type of stress that keeps getting mentioned in the press as if to counter the game mentality notion is that of remote pilots not getting a chance to wind down together and bond in their mutual operations. They return home to their families after their shifts have finished and even go to parents evening at school. The problem is that they have no debriefing about their day on the battlefield and can't talk to their families about it. This is a very different issue from whether or not they are morally engaged or disengaged from the killer strikes.

121.4 Targeted Killings

Up until now the discussion has centered on conventional forces using armed drones to support troops on the ground in the war zones of Afghanistan and Iraq. But there is a second armed UAV "Air Force" controlled by the CIA out of Langley in Virginia. The first ever U.S. kill with a drone was in Yemen in 2002 when a CIA-operated MQ-1 killed a known al-Qaeda leader and five other men traveling in the same vehicle. This was controversial at the time but was considered, by DoD lawyers, to be a legitimate defensive preemptive strike against al-Qaeda. Since then, the use of

Table 121.1 UAV Strikes Carried out by CIA

| Numbers killed | | | | |
|-------------------|-------------------|---------------|--------------|----------------|
| Year | Estimated strikes | High estimate | Low estimate | Leaders killed |
| 2004–2007 | 9 | 109 | 86 | 3 |
| 2008 | 34 | 296 | 263 | 11 |
| 2009 | 53 | 709 | 413 | 10 |
| 2010 ^a | 86 | 719 | 426 | 10 |

^aThrough October 10, 2010

drones for targeted killings or “decapitation strikes” in states that are not at war with the USA has been rising at an accelerating pace. It is what the Asia Times has called “the most public ‘secret’ war of modern times” (Turse 2010).

According to estimates published on the New America Foundation website (NAF 2010), the attacks in Pakistan have risen from 9 between 2004 and 2007 up to 85 in 2010 by mid-October. The estimates are shown in Table 121.1. The number of civilian deaths has been very difficult to estimate and has ranged from as few as 20 to more than a 1,000. There are many different counts and no objective way to decide between them. The New America Foundation gives a figure of 1 in 3 civilian to insurgent ratio, while the Brookings Institute study in 2009 indicated that only 1 in 10 of those killed was an insurgent.

The legality of targeted killing is at best questionable. The word decapitation is often used to mean cutting off the leaders of an organization or nation fighting a war from the body of their warfighters. The aim of the current spate of aerial decapitation strikes was initially to target al-Qaeda and Taliban leaders without risk to U.S. military personnel. The idea is to eventually leave only replacements from the shallowest end of the talent pool that will be ineffective and easy to defeat. With less risk, the targeted leaders do not have to be at the highest level. This explains the increasing number of death-delivering attacks in Pakistan using Predators and Reapers.

These individually targeted killings come despite the banning in the USA of all politically motivated killing of individuals since the famous Church commission report on the CIA political assassinations in 1975. In 1976, President Ford issued a presidential executive order that “no person employed by or acting on behalf of the United States Government shall engage in, or conspire to engage in, assassination.” This is not codified in the U.S. law but is an executive order that the president can change at will and without public notice. President Reagan issued Executive Order 12333, which expanded the ban to include persons acting on *behalf* of the U.S. government and all subsequent presidents have supported it. However, EO 12333 does not limit lawful self-defense options against legitimate threats to the national security of U.S. citizens. During wartime, a combatant is considered to be a legitimate target at all times. If a selected individual is sought out and killed, it is not termed an assassination. According to a memorandum on EO 12333, which is said to be consistent with United Nations (UN) Charter 51, “a decision by the President to employ clandestine, low-visibility, or overt military force would not constitute assassination if U.S. military forces were employed against the combatant forces of

another nation, a guerrilla force, or a terrorist or other organization whose actions pose a threat to the security of the United States” (NAF 2010).

But the real legal question is does the CIA have a right to assassinate *alleged* insurgent combatants without due process. Seymour Hersh (2002), whose writings were one of the main motivations for the Church commission, complained that “the targeting and killing of individual al-Qaeda members without juridical process has come to be seen within the Bush Administration as justifiable military action in a new kind of war, involving international terrorist organizations and unstable states.” The insurgents have been redefined as combatants, but without receiving the rights of prisoners of war and without being given the chance to surrender or to face trial. This move, in combination with Article 51, provides legal cover for the right to assassinate insurgent combatants.

The legality of the targeted killings has also been challenged at the UN General Assembly meeting in October 2009 by Philip Alston, UN special rapporteur on extrajudicial killings. He made a request for U.S. legal justification about how the CIA is accountable for the targets that they are killing. The USA turned down the request by refusing to comment on what they said were covert operations and a matter of national security.

A rebuttal by Harold Koh (2010), Legal Adviser, U.S. Department of State, stated, “U.S. targeting practices including lethal operations conducted by UAVs comply with all applicable law including the laws of war.” However, there are no independent means of determining how the targeting decisions are being made. It remains unclear as to what type and level of evidence is being used to sentence non-state actors to death by hellfire without right to appeal or right to surrender. It is also unclear as to what other methods were exhausted or attempted to bring the targets to justice. The whole process sits behind an unjustified cloak of national secrecy.

The CIA use of drones was also questioned at a senate hearing in a prepared statement by U.S. law professor Kenneth Anderson (2010). Anderson said that Koh “nowhere mentions the CIA by name in his defense of drone operations. It is, of course, what is plainly intended when speaking of self-defense separate from armed conflict. One understands the hesitation of senior lawyers to name the CIA’s use of drones as lawful when the official position of the U.S. government, despite everything, is still not to confirm or deny the CIA’s operations.”

However, the director of the CIA, Leon Panetta (2008), has been very open about the CIA use of drones. He told the Pacific Council on International Policy in 2008 that “it’s the only game in town in terms of confronting and trying to disrupt the al-Qaeda leadership.” Revealing the CIA’s intentions on the expansion of targeted drone kills, Panetta went on to say of al-Qaeda that “If they’re going to go to Somalia, if they’re going to go to Yemen, if they’re going to go to other countries in the Middle East, we’ve got to be there and be ready to confront them there as well. We can’t let them escape. We can’t let them find hiding places.”

This expansion of targeted killing is just what was concerning the UN special rapporteur on extrajudicial killings. A subsequent report by Alston (2010) to the UN General Assembly discusses drone strikes as violating international and human

rights law because both require transparency about the procedures and safeguards in place to ensure that killings are lawful and justified: “a lack of disclosure gives States a virtual and impermissible license to kill.” Some of Alston’s arguments also revolve around the notion of “the right to self-defense” and whether the drone strikes are legal under Article 51. He also examines some of the complex legal issues surrounding proportionality on the different conditions that pertain when the military action is called “self-defense.”

Panetta and many others have made the case that armed UAVs are more accurate and will kill less civilians than a B-52 bomber in attacking the tribal regions in Pakistan. But as a former CIA operative told me, there is no way that Pakistan or other state actors not at war with the USA could “turn a blind eye” to the bomber strikes as they do now for drones. It can be argued that it is their perceived precision and accuracy that allow them to penetrate areas and kill people in ways that would not previously have been available without major political and legal obstacles.

Attacking with a remotely piloted vehicle is no different under the Laws of War than attacking with a manned helicopter gunship or even with a rifle. The worry is that the nature of an unmanned vehicle with no risk to military personnel, an ability to hover over an area for very many hours, and its perceived accuracy may lead to considerable expansion of potential targets.

I qualified “accuracy” with the word “perceived” here both because the attacks are often on buildings where the targets cannot be verified, and, if the Brookings or New America Foundation figures discussed above are anything to go by, the accuracy is not as high as some might imagine. One of the oft-cited targeting methods of the CIA is to locate the locations of people through their cell phones; switch on your phone and you are dead. But a recent copyright lawsuit between two companies sheds doubt on the accuracy of this targeting method (Stein 2010). A small company called Intelligent Integration Systems alleges that one of its client companies, Netezza, reverse engineered their software, Geospatial, on a tight deadline for the CIA. The court heard that the illegal version of the software could produce locations that were out by as much as 40 ft and that the CIA had knowingly accepted the software.

But even if targeting was 100 % accurate, how can we be sure that alleged insurgents are “guilty as charged.” Information about target identity and their role and position is heavily dependent on the reliability of the intelligence on which it is based. There are lessons that should have been learned from the Vietnam War investigations of Operation Phoenix in which thousands were assassinated. It turned out that many of those on the assassination list had been put there by South Vietnamese officials for personal reasons such as erasing gambling debts or resolving family quarrels.

Philip Alston (*ibid.*) reports that during a mission to Afghanistan, he found out how hard it was for forces on the ground to obtain accurate information. “Testimony from witnesses and victims’ family members showed that international forces were often too uninformed of local practices, or too credulous in interpreting information, to be able to arrive at a reliable understanding of a situation.” He suggests that

“States must, therefore, ensure that they have in place the procedural safeguards necessary to ensure that intelligence on which targeting decisions are made is accurate and verifiable.”

121.5 From *in the Loop* to *on the Loop* to *Out of the Loop*

Since 2004 all of the road maps and plans of the U.S. forces have made clear the desire and intention to develop and use autonomous battlefield robots. Fulfillment of these plans to take the human out of the loop is well underway. The end goal is that robots will operate autonomously to locate their own targets and destroy them without human intervention (Sharkey 2008a). The Committee on Autonomous Vehicles in Support of Naval Operations National Research Council (2005) wrote that “The Navy and Marine Corps should aggressively exploit the considerable war-fighting benefits offered by autonomous vehicles (AVs) by acquiring operational experience with current systems and using lessons learned from that experience to develop future AV technologies, operational requirements, and systems concepts.” We can only hope that the “lessons learned” will include ethical and moral lessons. The ethical problems will only be amplified and added to by the use of autonomous robots.

Before delving into these problems, it is important to clarify what is meant by “robot autonomy” here. This is often confused with science fiction notions of robots with minds of their own with the potential to turn on humanity. The reality is very different. The autonomous robots being discussed for military applications are closer in operation to your washing machine than to a science fiction *Terminator*. The way the term “autonomy” is used in robotics should not be confused with how the term is used in philosophy, politics, individual freedom, or in common parlance. It is more related to the term automatic. An automatic robot carries out a preprogrammed sequence of operations or moves in a structured environment. A good example is a robot arm painting a car. An autonomous robot is similar except that it operates in open or unstructured environments. To do this, the robot is controlled by a program, takes inputs from its sensors, and adjusts the speed and direction of its motors and actuators as specified. If the goal of a robot is to avoid objects and the sensors detect an object, the program would adjust the motors so that the robot moves to avoid it. For example, if the left hand sensors detect the object, the robot would move right, and if the right hand sensors detect the object, the robot would move left.

This leads us to another often-misunderstood topic, the robot decision process. This should not be confused with human decision making except by analogy. A computer decision process can be as simple as, IF object on left, THEN turn right OR IF object on right, THEN turn left, ELSE continue. Alternatively, the activity on a sensor may activate a different subprogram to help with the decision. For example, to get smoother passage through a field laden with objects, a subprogram could be called into calculate if a turn to the left would result in having to negotiate more obstacles than a turn to the right.

Programs can become very complex through the management of several sub-programs and making decisions about which subprogram should be initiated in particular circumstances. But the bottom line for decision making by machine, whether it is using mathematical decision spaces or AI reasoning programs, is the humble IF/THEN statement.

The other important thing to say about autonomy is that a system does not have to be exclusively autonomous or exclusively remote operated. The U.S. Army, Navy, and Air Force have all at some point discussed the classification of robot control on a continuum from totally human operated to fully autonomous. Each has separate development programs and each has its own operational definitions of the different levels of robot autonomy. The Army has ten levels, while the Air Force has four. The Navy goes for a characterization in terms of mission complexity but points to three different classes of autonomous robot vehicle: (i) *scripted* in which the robot carries out a preplanned script of the “point, fire, and forget” variety; (ii) *supervised* in which some or all of the functions of planning, sensing, monitoring, and networking are carried out with the help of human operators to make decisions; (iii) *intelligent* in which attributes of human intelligence are used in software to make decisions, perceive and interpret the meaning of sensed information, diagnose problems, and collaborate with other systems.

The first of the Navy classifications, *scripted*, is the closest to “automatic” and includes cruise missiles, torpedoes, and automated weapons such as the Phalanx CIWS. The *supervised* category refers to both the current “man-in-the-loop” systems such as the Predator and Reaper control and the newly proposed “man-on-the-loop” systems. And the *intelligent* category refers to fully autonomous operation although frankly I am not sure exactly what they mean by “attributes of human intelligence in software.” Perhaps there is a little slippage into wishful thinking here.

There are a number of reasons that make an autonomous robot militarily desirable including the following: (i) remote-operated systems are more expensive to manufacture and require many support personnel to run them; (ii) it is possible to jam either the satellite or radio link or take control of the system; (iii) one of the military goals is to use robots as force multipliers so that one human can be a nexus for initiating a large-scale robot attack from the ground and the air; and (iv) the delay time in remote piloting a craft via satellite (approximately 1.5 s) means that it could not be used for interactive combat with another aircraft. At a press briefing in December 2007, Dyke Weatherington, deputy director of the U.S. DoD’s Unmanned Aerial Systems Task Force, said, – Certainly the roadmap projects an increasing level of autonomy . . . to fulfill many of the most stressing requirements. Let me just pick one for example. Air-to-air combat – there’s really no way that a system that’s remotely controlled can effectively operate in an offensive or defensive air combat environment. That has to be “the requirement of that is a fully autonomous system” (DoD 2007).

The most recent United States Air Force Unmanned Aircraft Systems Flight Plan 2009–2047 (USAF 2009) opens the strategy for a staged move from current remote piloted systems to fully autonomous systems. It represents a push to shrink the role of the man-in-the-loop. To begin with, autonomous operation will be mainly for

tasks such as take off, landing, and refueling. As unmanned drones react in micro- or nanoseconds the “humans will no longer be ‘in the loop’ but rather ‘on the loop’ – monitoring the execution of certain decisions. Simultaneously, advances in AI will enable systems to make combat decisions and act within legal and policy constraints without necessarily requiring human input” (ibid., p. 41).

The idea of a human on the loop means that the human will be in executive control overall to call in or call off the robots rather than being in control of each individually. In other words, the robots will be essentially autonomous: “SWARM technology will allow multiple MQ-Mb aircraft to cooperatively operate in a variety of lethal and nonlethal missions at the command of a single pilot” (ibid., p. 39). Such a move will require decisions being made by the swarm – human decision making will be too slow and not able to react to the control of several aircraft at once. With the increasing pace of the action and with potential of several aircraft to choose targets at the same time, it will not be possible to have the human make all of the decisions to kill.

The main ethical concern is that allowing robots to make decisions about the use of lethal force could breach both the principle of distinction and the principle of proportionality as specified by international humanitarian law (Sharkey 2008b). Currently and for the foreseeable future, no autonomous robots or artificial intelligence systems have the necessary properties to enable discrimination between combatants and civilians or to make proportionality decisions.

Under the principle of distinction, only combatants/warriors are legitimate targets of attack. All others, including children, civilians, service workers, and retirees, should be immune from attack. The same immunity covers combatants that are wounded, have surrendered, or are mentally ill (Ford 1947). The principle of proportionality applies in circumstances where it is not possible to fully protect noncombatants in an action. It requires that the loss of life and damage to property incidental to attacks must not be excessive in relation to the concrete and direct military advantage.

The discrimination between civilians and combatants is problematic for any robot or computer system. First, there is the problem which is the specification of “civilianness.” A computer can compute any given procedure that can be written as a program. We could, for example, give the computer on a robot an instruction such as “if civilian, do not shoot.” This would be fine if and only if there was some way to give the computer a precise specification of what a civilian is. The Laws of War don’t help. The 1949 Geneva Convention requires the use of common sense to determine the difference between a civilian and combatant, while the 1977 Protocol 1 essentially defines a civilian in the negative sense as someone who is not a combatant:

1. A civilian is any person who does not belong to one of the categories of persons referred to in Article 4 A (1), (2), (3), and (6) of the Third Convention and in Article 43 of this Protocol. In case of doubt whether a person is a civilian, that person shall be considered to be a civilian.
2. The civilian population comprises all persons who are civilians.

3. The presence within the civilian population of individuals who do not come within the definition of civilians does not deprive the population of its civilian character. Protocol 1 Additional to the Geneva Conventions, 1977 (Article 50).

Even if a clear definition of civilian did exist, it would have to be couched in a form that enabled the relevant information to be extracted from the sensing apparatus. All that is available to robots are sensors such as cameras, infrareds, sonars, lasers, temperature sensors, and ladars. While these may be able to tell us that something is a human or at least an animal, they could not tell us much about combat status. There are systems that can identify a face or a facial expression, but they do not work well on real-time moving people. And even if they did, how useful could they be in the fog of war or from the air? British teenagers beat the surveillance cameras just by wearing hooded jackets.

In a war with nonuniformed combatants, knowing who to kill would have to be based on situational awareness and of having human understanding of other people's intentions and their likely behavior. In other words, human inference is required. Humans understand one another in a way that machines cannot. Cues can be very subtle and there are an infinite number of circumstances where lethal force is inappropriate. Just think of children being forced to carry empty rifles or of insurgents burying their dead.

The problem has been put clearly by Major Daniel Davis (2008), a combat veteran of Iraq 1991 and Afghanistan 2005, who writes, "Suggesting that within the next 12-plus years technology could exist that would permit life-and-death decisions to be made by algorithms is delusional. A machine cannot sense something is wrong and take action when no orders have been given. It doesn't have intuition. It cannot operate within the commander's intent and use initiative outside its programming. It doesn't have compassion and cannot extend mercy". Davis also quotes Colonel Lee Fetterman who has a high regard for remote controlled robots: "The function that robots cannot perform for us – that is, the function we should not allow them to perform for us – is the decide function. Men should decide to kill other men, not machines," he said. "This is a moral imperative that we ignore at great peril to our humanity. We would be morally bereft if we abrogate our responsibility to make the life-and-death decisions required on a battlefield as leaders and soldiers with human compassion and understanding. This is not something we would do. It is not in concert with the American spirit" (Fetterman quoted in Davis, *ibid.*).

Turning to the principle of proportionality, there is no way for a robot could perform the human subjective balancing act required to make proportionality decisions. No clear objective methods are provided for calculating what is proportionate in the Laws of War (Sharkey 2009). The phrase "excessive in relation to the concrete and direct military advantage expected to be gained" is not a specification. It is also practically impossible to calculate a value for "the actual military advantage." What could the metric be for assigning value to killing an insurgent relative to the value of noncombatants, particularly children who could not be accused of willingly contributing to insurgency activity? Proportionality calculations should be based on

the likely differences in military outcome if the military action killing innocents had not been taken (Chakwin et al. 2002).

Commanders have to weigh the circumstances before making a decision, but ultimately this is a subjective matter. Clearly the extremes of wiping out a whole city to eliminate even the highest value target, say Osama bin Laden, is out of the question. There has to be circumstantial and subjective estimates about just how many innocent people are equal to the military value of the successful completion of a given mission. A computer system could be used to assist commanders to determine which weapons would cause the minimum of collateral damage. However, after the calculation is completed, a decision must still be made about whether the minimal number of civilian casualties and damage to civilian property is proportional to the military advantage gained.

In the heat of battle, both the principles of discrimination and proportionality can be problematic. Humans do make errors and can behave unethically, but they can be held accountable. Armed autonomous robots could violate both of these principles but could not be held accountable (Sharkey 2008c). There is no way to punish a robot. We could threaten to switch it off but that would be like telling your washing machine that if it does not remove stains properly, you will break its door off. This leaves the question about who is responsible for the long causal chain that includes the following: the manufacturer for the description they gave, the programmer, the designer, the department of defense, the generals or admirals in charge of the operation, and the operator. Can we hold someone accountable, if not then legally these robots cannot be used (Sparrow 2007).

121.6 Some Lessons to Learn

It is quite likely that autonomous robots will come into operation in a piecemeal fashion. Research and development is well underway and the fielding of autonomous robot systems may not be far off. However, to begin with they are likely to have assistive autonomy onboard such as flying or driving a robot to a target destination and perhaps even selecting targets and notifying a human (the Israeli autonomous ground robot, Guardium, already has this functionality). This will breed public trust and confidence in the technology – an essential requirement for progression autonomy (Dahm 2010). The big worry is that allowing such autonomy will be a further slide down a slippery slope to give machines the power to make decisions about who to kill.

Apart from the legal and ethical problems of discrimination and proportionality for fully autonomous armed robots, there are lessons to be learned from the current application of remotely piloted vehicles. We discussed some of the controversy about the issue of moral disengagement of the operators and the PlayStation mentality (Royakkers and van Est 2010). While this is an interesting and plausible narrative, it is not possible to be conclusive because the evidence, outside of the anecdotal, is just not available. It must be noted, nonetheless, that the power of

graphic detail to desensitize should not be underestimated. The narrative about moral disengagement is even more compelling when it comes to with autonomous robot warriors. As Cumming (2006) puts it, “developing technologies of this sort also have the potential to provide for the creation of moral buffers that allow humans to act without adequately considering the consequences.”

Another lesson can be learned from the use of the remotely piloted armed aircraft by the CIA for targeted killings. At best, such targeted killings are legally questionable in terms of the Laws of War as discussed above. Apart from problems with verifying intelligence about targeting and the lack of a transparent judicial process, there is concern that the use of drones has greatly expanded the range of targets that can be picked off by the CIA. Autonomous aircraft that can hover on their own for extended periods and select their own targets could greatly exacerbate the problems. The DARPA Vulture program has just awarded a contract to Boeing (teamed with QinetiQ) to develop a heavier-than-air platform capable of keeping 1,000 lb of payload with 5 kW of power aloft for 5 years. These developments need to be watched very carefully.

It is difficult to design and develop these new technologies, but they are not so difficult to copy. We have already seen the rapid proliferation of drones in response to their perceived military success in the Iraq/Afghanistan conflicts. If armed autonomous robots are fielded in the same way with or without a human in the weapons loop, it will not be long before there is similar proliferation. If it turns out to be militarily advantageous to keep a human out of the loop – for example, in circumstances when it is not possible to maintain a radio or satellite link – then they will be used by all.

References

- P. Alston, Report of the special rapporteur on extrajudicial, summary or arbitrary executions, Philip Alston to the UN Human Rights Council, Fourteenth session, A/HRC/14/24/Add.6, 28 May 2010 (2008)
- K. Anderson, Written testimony to the US House of representatives committee on oversight and government reform, Subcommittee on National Security and Foreign Affairs, Subcommittee Hearing: “Drones II”, 28 Apr 2010
- B. Chakwin, D. Voelkel, S. Enright, Leaders as targets, Joint Forces Staff College, Seminar # 08 (2002)
- Committee on Autonomous Vehicles in Support of Naval Operations National Research Council, *Autonomous Vehicles in Support of Naval Operations* (The National Academies Press, Washington, DC, 2005)
- M.L. Cumming, Automation and accountability in decision support system interface design. *J. Technol. Stud.* **32**(1), 23–31 (2006)
- A. Daddis, Understanding fear’s effect on unit effectiveness. *Mil. Rev.* **84**, 22–27 (2004)
- W.J.A. Dahm, “Technology Horizons”: A Vision for Air Force Science & Technology, vol. 1, AF/ST-TR-10-01 (2010), http://www.airforce-magazine.com/SiteCollectionDocuments/TheDocumentFile/StrategyandConcepts/TechnologyHorizonsVol1_2010.pdf. Last accessed 14 Oct 2010
- D. Davis, Who decides: man or machine? *Armed Forces J.* (2008)

- DoD, DoD Press briefing with Mr. Weatherington from the Pentagon briefing room. US Department of Defense, 18 Dec 2007, www.defenselink.mil/transcriptstranscript.aspx?transcriptid=4108
- J.S. Ford, *The Morality of Obliteration Bombing* (Theological Studies, Woodstock, NY: Theological Studies, 1944), pp. 261–309
- D. Grossman, *On Killing: The Psychological Cost of Learning to Kill in War and Society* (Little, Brown and Co., Boston, 1995)
- S.M. Hersh, Manhunt: the Bush administration's new strategy in the war against terrorism. *New Yorker* 66 (23 Dec 2002)
- R. Holmes, *Acts of War: The Behaviour of Men in Battle* (Cassell, London, 2003)
- H. Koh, *Speech to the American Society of International Law*, Washington, DC, 25 Mar 2010
- S. Lindlaw, UAV operators suffer war stress. *Air Force Times* (2008), http://www.airforcetimes.com/news/2008/08/ap_remote_stress.080708/. Last accessed 10 Oct 2010
- S.L.A. Marshall, *Men Against Fire: The Problem of Battle Command* (First published by William Morrow & Company, 1947). (Reprinted: Oklahoma University Press, Norman, 2000)
- NAF, The year of the drones, New America Foundation website (2010), <http://counterterrorism.newamerica.net/drones>. Last accessed 15 Oct 2010
- L.E. Panetta, Director's remarks at the Pacific Council on international policy (2008), <https://www.cia.gov/news-information/speeches-testimony/directors-remarks-at-pacific-council.html>. Last accessed 15 Oct 2010
- L. Royakkers, R. van Est, The cubicle warrior: the marionette of digitalized warfare. *Ethics Inf. Technol.* **12**, 289–296 (2010)
- N. Sharkey, Cassandra or the false prophet of doom: AI robots and war. *IEEE Intell. Syst.* **23**(4), 14–17 (2008a). July/August Issue
- N. Sharkey, Grounds for discrimination: autonomous robot weapons. *RUSI Def. Syst.* **11**(2), 86–89 (2008b)
- N. Sharkey, The ethical frontiers of robotics. *Science* **322**, 1800–1801 (2008c)
- N. Sharkey, Death strikes from the sky: the calculus of proportionality. *IEEE Sci. Soc.* **28**, 16–19 (2009). Spring
- P.W. Singer, *Wired for War: The Robotics Revolution and Conflict in the 21st Century* (The Penguin Press, New York, 2009)
- R. Sparrow, Killer robots. *J. Appl. Philos.* **24**, 62–77 (2007)
- J. Stein, CIA drones could be grounded by software suit, *Washington Post*, SpyTalk, 11 Oct (2010), http://blog.washingtonpost.com/spy-talk/2010/10/cia_drones.could_be_grounded_b.html. Last accessed 14 Oct 2010
- N. Turse, Drone surge: today, tomorrow and 2047. *Asia Times* (2010). http://www.atimes.com/atimes/South_Asia/LA26Df01.html. Last accessed 15 Oct 2010
- USAF, *United States Air Force Unmanned Aircraft Systems Flight Plan 2009–2047* (Headquarters of the United States Air Force, Washington, DC, 2009)

Ronald C. Arkin

Contents

| | | |
|------------|---|------|
| 122.1 | Introduction | 2933 |
| 122.2 | Human Failings in the Battlefield | 2935 |
| 122.3 | A Way Forward | 2939 |
| References | | 2940 |

Abstract

The underlying thesis of research in ethical autonomy for lethal autonomous unmanned systems is that they will potentially be capable of performing more ethically on the battlefield than are human soldiers. In this article this hypothesis is supported by ongoing and foreseen technological advances and perhaps equally important by an assessment of the fundamental human warfighters in today’s battlespace. If this goal of better-than-human performance is achieved, even if still imperfect, it can result in a reduction in noncombatant casualties and property damage consistent with adherence to the Laws of War as prescribed in international treaties and conventions and is thus worth pursuing vigorously.

122.1 Introduction

The trend is clear: Warfare will continue and autonomous robots will ultimately be deployed in its conduct. Referring to the improving technology of the day

From the Journal of Military Ethics, vol. 9, no. 4 (2010); reprinted by permission of the publisher (Taylor and Francis, Ltd., <http://www.tandf.co.uk/journals>).

R.C. Arkin
Department of Computer Science, Georgia Tech University, Atlanta, GA, USA
e-mail: ronald.arkin@cc.gatech.edu

and its impact on the inevitability of warfare, Clausewitz stated “the tendency to destroy the adversary which lies at the bottom of the conception of War is in no way changed or modified through the progress of civilization” (Von Clausewitz 1832). More recently, Cook observed “The fact that constraints of just war are routinely overridden is no more a proof of their falsity and irrelevance than the existence of immoral behavior ‘refutes’ standards of morality: the standards are well known, and it also well known that human beings fall short of that standard with depressing regularity” (Cook 2004). Given this, questions then arise regarding if and how these systems can conform as well or better-than-human soldiers with respect to adherence to the existing Laws of War. If achieved, this would result in a reduction in collateral damage, i.e., noncombatant casualties and civilian property. The body of research conducted in Georgia Tech’s unmanned systems laboratory (Arkin 2009; Arkin and Ulam 2009; Arkin et al. 2009) focuses on this issue directly from a design perspective. As robots are already faster, stronger, and in certain cases (e.g., chess playing) smarter than humans, is it that difficult to believe they will be able to treat rival combatants more humanely in the battlefield than those combatants currently do each other?

This is no simple task, however. In the fog of war, it is hard enough for a human to be able to effectively discriminate whether or not a target is legitimate. Fortunately for a variety of reasons, it may be anticipated, despite the current state of the art, that in the future autonomous robots may be able to perform better than humans under these conditions, for the following reasons:

1. The ability to act conservatively, i.e., they do not need to protect themselves in cases of low certainty of target identification. Autonomous armed robotic vehicles do not need to have self-preservation as a foremost drive, if at all. They can be used in a self-sacrificing manner if needed and appropriate without reservation by a commanding officer. There is no need for a “shoot first, ask-questions later” approach.
2. The eventual development and use of a broad range of robotic sensors better equipped for battlefield observations than humans currently possess. This includes technological advances in electro-optics, synthetic aperture or wall-penetrating radars, acoustics, and seismic sensing, to name but a few.
3. Unmanned robotic systems can be designed without emotions that cloud their judgment or result in anger and frustration with ongoing battlefield events. In addition, “Fear and hysteria are always latent in combat, often real, and they press us toward fearful measures and criminal behavior” (Walzer 1977). Autonomous agents need not suffer similarly.
4. Avoidance of the human psychological problem of “scenario fulfillment” is possible, a factor believed partly contributing to the downing of an Iranian airliner by the USS *Vincennes* in 1988 (Sagan 1991). This phenomenon leads to distortion or neglect of contradictory information in stressful situations, where humans use new incoming information in ways that only fit their preexisting belief patterns, a form of premature cognitive closure. Robots need not be vulnerable to such patterns of behavior.

5. They can integrate more information from more sources far faster before responding with lethal force than a human possibly could in real time. This data can arise from multiple remote sensors and intelligence (including human) sources, as part of the army's network-centric warfare concept (McLoughlin 2006) and the concurrent development of the Global Information Grid (DARPA 2007). "Military systems (including weapons) now on the horizon will be too fast, too small, too numerous and will create an environment too complex for humans to direct" (Adams 2001–2002).
6. When working in a team of combined human soldiers and autonomous systems as an organic asset, they have the potential capability of independently and objectively monitoring ethical behavior in the battlefield by all parties and reporting infractions that might be observed. This presence alone might possibly lead to a reduction in human ethical infractions.

Aside from these ethical considerations, autonomous robotic systems offer numerous other potential operational benefits to the military: faster, cheaper, better mission accomplishment, longer range, greater persistence, longer endurance, higher precision, faster target engagement, and immunity to chemical and biological weapons among others (Guetlein 2005). All of these can enhance mission effectiveness and serve as drivers for the ongoing deployment of these systems. But recent research (e.g., Arkin 2009) focuses on enhancing ethical benefits by using these systems, ideally without eroding mission performance when compared to human warfighters.

122.2 Human Failings in the Battlefield

It is not my belief that an autonomous unmanned system will be able to be perfectly ethical in the battlefield, but I am convinced that they can perform more ethically than human soldiers are capable of performing. Unfortunately the trends in human behavior in the battlefield regarding adhering to legal and ethical requirements are questionable at best. "Armies, armed groups, political and religious movements have been killing civilians since time immemorial" (Slim 2008, p. 3). The dangers of abuse of unmanned robotic systems, such as the Predator and Reaper, in war are well documented, which occurs even when a human operator is directly in charge (Sullivan 2010; Filkins 2010; Adams 2010). Battlefield atrocities are as old as warfare. "Atrocity... is the most repulsive aspect of war, and that which resides within man and permits him to perform these acts is the most repulsive aspect of mankind" (Grossman 1995, p. 229).

Humanity's propensity to wage war has gone unabated for as long as history has been recorded. One could argue that man's greatest failing is being on the battlefield in the first place. Immanuel Kant asserted "War requires no motivation, but appears to be ingrained in human nature and is even valued as something noble" (Kant 1985, p. 125). Even Albert Einstein, who remained a pacifist well into his 50s, eventually acknowledged "as long as there will be man, there will be war" (Isaacson 2007, p. 494). Sigmund Freud was even more to the point: "... there is no

likelihood of being able to suppress humanity's aggressive tendencies" (Isaacson 2007, p. 382). This article, however, is concerned for the large part with the shortcomings humanity exhibits during the conduct of war (*jus in bello*) as opposed to what brought us there in the first place (*jus ad bellum*). "The emotional strain of warfare and combat cannot be quantified" (Bourke 1999, p. 232), but at least there has recently been a serious attempt to gather data on that subject. A recent report from the Surgeon General's Office (Surgeon General 2006) assessing the battlefield ethics and mental health of soldiers and marines deployed in Operation Iraqi Freedom is disturbing. The following findings are taken directly from that report:

1. Approximately 10 % of soldiers and marines report mistreating noncombatants (damaged/destroyed Iraqi property when not necessary or hit/kicked a noncombatant when not necessary). Soldiers that have high levels of anger experience high levels of combat or those who screened positive for a mental health problem were nearly twice as likely to mistreat noncombatants as those who had low levels of anger or combat or screened negative for a mental health problem.
2. Only 47 % of soldiers and 38 % of marines agreed that noncombatants should be treated with dignity and respect.
3. Well over a third of soldiers and marines reported torture should be allowed, whether to save the life of a fellow soldier or marine or to obtain important information about insurgents.
4. Seventeen percent of soldiers and marines agreed or strongly agreed that all noncombatants should be treated as insurgents.
5. Just under 10 % of soldiers and marines reported that their unit modifies the ROE to accomplish the mission.
6. Forty-five percent of soldiers and 60 % of marines did not agree that they would report a fellow soldier/marine if he had injured or killed an innocent noncombatant.
7. Only 43 % of soldiers and 30 % of marines agreed they would report a unit member for unnecessarily damaging or destroying private property.
8. Less than half of soldiers and marines would report a team member for engaging in unethical behavior.
9. A third of marines and over a quarter of soldiers did not agree that their NCOs and officers made it clear not to mistreat noncombatants.
10. Although they reported receiving ethical training, 28 % of soldiers and 31 % of marines reported facing ethical situations in which they did not know how to respond.
11. Soldiers and marines are more likely to report engaging in the mistreatment of Iraqi noncombatants when they are angry and are twice as likely to engage in unethical behavior in the battlefield than when they have low levels of anger.
12. Combat experience, particularly losing a team member, was related to an increase in ethical violations.

This formal study, although at the very least disconcerting, is by no means the first report of battlefield atrocities. "Atrocious behavior was a feature of combat in

the two world wars, as well as in Vietnam” (Bourke 1999, p. 163). One sociological study of fighting in Vietnam pointed out that for all men in heavy combat, 1/3 of men in moderate combat, and 8 % in light combat had seen atrocities or committed or abetted noncombatant murder (Strayer and Ellenhorn 1975). These numbers are staggering.

Possible explanations for the persistence of war crimes by combat troops are discussed in Bill (2000), Parks (1976a,b), Danyluk (2000), and Slim (2008). These include:

- High friendly losses leading to a tendency to seek revenge.
- High turnover in the chain of command, leading to weakened leadership.
- Dehumanization of the enemy through the use of derogatory names and epithets.
- Poorly trained or inexperienced troops. This lack of training is not just in being a good soldier but also in understanding the Laws of War.
- No clearly defined enemy.
- The issuance of unclear orders where the intent of the order may be interpreted incorrectly as unlawful.
- Shortage of personnel has been associated in producing stress on combatants that can lead to violations.
- Youth and immaturity of troops.
- An overpowering sense of frustration.
- Pleasure from the power of killing.
- External pressure, e.g., for a need to produce a high body count of the enemy.

There is clear room for improvement, and autonomous systems may help. Bourke points out that modern warfare enables violent acts in ways unlike before. Now, “Combatants were able to maintain an emotional distance from their victims largely through the application of. . . technology” (Bourke 1999, p. xvii). This portends ill for the reduction of atrocities by soldiers. Bombs are presently being dropped in Afghanistan and Iraq by UAV operators from almost halfway around the world in Nevada (Ure 2008). This use of technology enables a form of “numbed killing.” Bourke further notes that there is now a “technological imperative” to make full use of the new equipment provided. Although technological warfare has reduced the overall number of soldiers required to wage war, the price is that technology, while increasing the ability to kill, decreases “the awareness that dead human beings were the end product.” When killing at a maximum range, one can pretend they are not killing human beings and thus experience no regret (Grossman 1995). This physical distance detaches the warfighters from the consequences of the use of their weaponry.

The psychological consequences on U.S. servicemen and women in Afghanistan and Iraq have reached record levels. In 2007 alone, 115 soldiers committed suicide, up from 102 the previous year; 24 % of the suicides were those on their first deployment, and 43 % were those who had returned from deployment. The suicide rates of active duty soldiers as of August 2008 “were on pace to surpass both last year’s numbers and the rate of suicide in the general U.S. population for the first time since the Vietnam war, according to U.S. Army officials” (Mount 2008, p. 1).

This unfortunately was confirmed in July of 2010 (Fifield 2010). A statistically significant relationship has been established between the suicide attempts and the number of days spent deployed in Iraq or Afghanistan. To make matters worse, this is coupled with “a growing number of troops diagnosed with post traumatic stress disorder” (Sevastopulo 2008, p. 1).

These psychiatric casualties are quite significant and common (Grossman 1995): In World War II alone more than 800,000 men were classified unfit due to psychiatric reasons, but an additional 504,000 (approximately 50 divisions) were subsequently rendered unfit as a result of psychiatric collapse after induction; in the 1973 Arab-Israeli war one-third of the Israel casualties were psychiatric in origin, twice the number of dead troops. One WWII study showed that after 60 days of continuous combat, 98 % of all surviving troops suffer psychiatric trauma of some sort (Swank and Marchand 1946). These long-term exposures to combat are a recent trend in battle, emerging in the twentieth century. The psychiatric damage can result in many forms: battlefield fatigue, conversion hysteria, confusional states, anxiety states, obsession and compulsive states, and character disorders (Grossman 1995). The overall effect on the ability to wage war is obvious, let alone the damage to a nation’s surviving citizens.

Creating true warfighters in the first place is a daunting challenge. “No matter how thorough the training, it still failed to enable most combatants to fight” (Bourke 1999, p. 61). In World War II most men simply did not kill. In one U.S. Army interview of 400 men, only 15 % of men had actually fired at enemy positions (at least once) during an engagement despite the fact that 80 % had the opportunity to do so (Marshall 1947). There was no observed correlation between the experience, terrain, nature of the enemy, or accuracy of enemy fire on this percentage.

This applied to both land and air forces. One study of the Korean War indicated that 50 % of F-86 pilots never fired their guns and only 10 % of those had actually hit a target (Sparks and Neiss 1956). During World War II, most fighter pilots never even tried to shoot anyone down, let alone succeeding. Less than 1 % of the pilots accounted for 30–40 % of all downed enemy aircraft (Grossman 1995, p. 31).

One conclusion of this is that human soldiers, although not cowardly, lacked an “offensive spirit.” One possible reason for this lack of aggressiveness centers on the use of long-distance weapons making battlefields “lonely” and the feeling the enemy was not real but a phantom. This dehumanization of the enemy also quells guilt in killing (Bourke 1999).

The soldiers in the field are not alone in their complicity. “Atrocities are the dark secret of military culture” (Danyluk 2000, p. 38). “Servicemen of all ranks were unperturbed by most of these acts of lawless killing” (Bourke 1999, p. 173). In Vietnam, combat commanders viewed the Laws of War as “unnecessary” and “unrealistic” restraining devices which would decrease the opportunity for victory (Parks 1976a, p. 21). A lawyer, defending one general’s decision not to initiate a court martial for suspected war crime violations, stated, “It’s a little like the Ten Commandments – they’re there, but no one pays attention to them”.

Nonetheless the U.S. military aspires to higher ethical performance. General Douglas MacArthur stated:

The soldier, be he friend or foe, is charged with the protection of the weak and unarmed. It is the very essence and reason for his being. When he violates this sacred trust, he not only profanes the cult but threatens the very fabric of international society. (Parks 1976a, p. 18)

In addition, the impact of atrocities on public opinion, as clearly evidenced by the My Lai incident in the Vietnam War, and the consequent effect on troop morale are secondary reasons to ensure that events like these are prevented. Civilians are unfortunately killed during war by other humans for manifold reasons (Slim 2008):

- Genocidal thinking – ethnic or racial cleansing of populations.
- Dualistic thinking – dividing host populations into the “good guys” and the “bad guys.”
- Power dominance and subjugation – power lust and to exert force.
- Revenge – emotional striking back for perceived wrongs.
- Punishment and forced compliance – to shape the behavior of civilian populations.
- Utility – it furthers the war strategically.
- Asymmetrical necessity – tactical killing of civilians due to an inferior military position.
- Profit – mercenary and looting activity.
- Eradicating potential – preemptive removal of civilians who might otherwise become warfighters in the future.
- Recklessness – shooting anything that moves or other forms of Indiscriminate killing.
- Reluctant killing – through human error or accident, collateral damage.
- Collective and sacrificial thinking – killing of groups rather than individuals, they must be sacrificed for a greater good.

These forms of thinking are alien to current artificial intelligence efforts and likely are to remain so. Armed autonomous systems need not nor should be equipped with any of these forms of unacceptable human rationalization or action.

A primary conclusion is that it seems unrealistic to expect normal human beings by their very nature to adhere to the Laws of Warfare when confronted with the horror of the battlefield, even when trained. As a Marine Corps Reserves Captain commented, “If wars cannot be prevented, steps can be taken to ensure that they are at least fought in as ethical a manner as possible” (Danyluk 2000, p. 38). One could argue that battlefield atrocities, if left unchecked, may become progressively worse, with the progression of standoff weapons and increasing use of technology. Something must be done to restrain the technology itself, above and beyond the human limits of the warfighters themselves. This is the case for the use of ethical autonomy in unmanned systems.

122.3 A Way Forward

Research in our laboratory has provided the motivation, philosophy, formalisms, representational requirements, architectural design criteria, recommendations, and test scenarios to design and construct an autonomous robotic system architecture

capable of the ethical use of lethal force (Arkin 2009). These first steps toward that goal, however, are very preliminary and subject to major revision, but at the very least they can be viewed as the beginnings of an ethical robotic warfighter. The primary goal remains to enforce international humanitarian law (or the Laws of Armed Conflict (LOAC)) in the battlefield in a manner that is believed achievable, by creating a class of robots that not only comply with the restrictions of international law but in fact outperform human soldiers in their ethical capacity under comparable circumstances. If successful this will result in the saving of noncombatant life and property, ideally without erosion of mission performance. It is too early to tell whether this venture will be successful. There are daunting problems remaining:

- The transformation of international protocols and battlefield ethics into machine-usable representations and real-time reasoning capabilities for bounded morality using modal logics.
- Mechanisms to ensure that the design of intelligent behaviors only provide responses within rigorously defined ethical boundaries.
- The development of effective perceptual algorithms capable of superior target discrimination capabilities, especially with regard to combatant-noncombatant status.
- The creation of techniques to permit the adaptation of an ethical constraint set and underlying behavioral control parameters that will ensure moral performance, should those norms be violated in any way, involving reflective and affective processing.
- A means to make responsibility assignment clear and explicit for all concerned parties regarding the deployment of a machine with a lethal potential on its mission.

Hopefully the goals of these initial and limited efforts will fuel other scientists' interest to assist in ensuring that the machines that roboticists create fit within international and societal expectations and requirements. My personal hope would be that they will never be needed in the present or the future. But mankind's tendency toward war seems overwhelming and inevitable. At the very least, if civilian casualties can be reduced in compliance with applicable protocols of the Geneva Conventions and the ideals enshrined within the just war tradition, the result will have constituted a significant humanitarian achievement, even while staring directly at the face of war.

References

- T. Adams, Future warfare and the decline of human decisionmaking. *Parameters*, U.S. Army War College Quarterly, Winter 2001–2002, pp. 57–71
- J. Adams, US defends unmanned drone attacks after harsh UN report. *Christian Science Monitor*, 5 June 2010
- R.C. Arkin, *Governing Lethal Behavior in Autonomous Systems* (Ft. Lauderdale, FL: Taylor and Francis, 2009)
- R.C. Arkin, P. Ulam, An ethical adaptor: behavioral modification derived from moral emotions, in *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA-09)*, Daejeon, Dec 2009

- R.C. Arkin, A. Wagner, B. Duncan, Responsibility and lethality for unmanned systems: ethical pre-mission responsibility advisement, in *Proceedings of the 2009 IEEE Workshop on Roboethics*, Kobe, May 2009
- B. Bill (ed.), *Law of War Workshop Deskbook* (International and Operational Law Department, Judge Advocate General's School, Charlottesville, 2000)
- J. Bourke, *An Intimate History of Killing* (Basic Books, New York, 1999)
- M. Cook, *The Moral Warrior: Ethics and Service in the U.S. Military* (State University of New York Press, Albany, 2004)
- S. Danyluk, Preventing atrocities. *Mar. Corps Gaz.* **84**(6), 36–38 (2000)
- DARPA (Defense Advanced Research Projects Agency) Broad Agency Announcement 07-52. Scalable Network Monitoring, Strategic Technology Office, Aug 2007
- A. Fifield, U.S. Army suicide rate exceeds national average. *Financial Times*, 30 July 2010, <http://www.ft.com/cms/s/0/2c662840-9b74-11df-8239-00144feab49a.html>. Accessed 21 July 2010
- D. Filkins, Operators of drones are faulted in Afghan deaths. *New York Times*, 29 May 2010, A1
- D. Grossman, *On Killing: The Psychological Cost of Learning to Kill in War and Society* (Little, Brown and Company, Boston, 1995)
- M. Guetlein, Lethal autonomous systems – ethical and doctrinal implications. Naval War College Joint Military Operations Department Paper, Feb 2005
- W. Isaacson, *Einstein: His Life and Universe* (Simon and Schuster, New York, 2007)
- I. Kant, *Perpetual Peace and Other Essays on Politics, History, and Morals* (Trans. T. Humphrey) (Hackett, Indianapolis, 1985)
- S.L.A. Marshall, *Men Against Fire: The Problem of Battle Command in Future War* (William Morrow, New York, 1947)
- R. McLoughlin, Fourth generation warfare and network-centric warfare. *Mar. Corps Gaz.* 15 Sept 2006, <https://feedback.mcamarines.org/gazette/06mcloughlin.asp>. Accessed 30 July 2010
- M. Mount, Army suicide rate could top nation's this year, CNN.com, 9 Sept 2008, <http://www.cnn.com/2008/HEALTH/09/09/army.suicides/>. Accessed 31 July 2010
- W.H. Parks, Crimes in hostilities. Part I. *Mar. Corps Gaz.* (1976a)
- W.H. Parks, Crimes in hostilities. Conclusion. *Mar. Corps Gaz.* (1976b)
- S. Sagan, Rules of engagement, in *Avoiding War: Problems of Crisis Management*, ed. by A. George (Westview Press, Boulder, 1991)
- D. Sevastopulo, US Army suicide cases at record 115. *Financial Times*, 29 May 2008, http://us.ft.com/ftgateway/superpage.ft?news_id=fto052920081802392265. Accessed 30 July 2010
- H. Slim, *Killing Civilians: Method, Madness, and Morality in War* (Columbia University Press, New York, 2008)
- B.W. Sparks, O. Neiss, Psychiatric screening of combat pilots. *US Armed Forces Med. J.* **4**(VII.6), 811–816 (1956)
- R. Strayer, L. Ellenhorn, Vietnam veterans: a study exploring adjustment patterns and attitudes. *J. Soc. Issues* **33**, 4 (1975). As reported in (Bourke 99)
- R. Sullivan, *Drone Crew Blamed in Afghan Civilian Deaths* (New York: Associated Press, 2010)
- Surgeon General's Office, Mental Health Advisory Team (MHAT) IV Operation Iraqi Freedom 05–07, Final Report, 17 Nov 2006
- R. Swank, W. Marchand, Combat neuroses: development of combat exhaustion. *Arch. Neurol. Psychol.* **55**, 236–47 (1946)
- L. Ure, Armchair pilots striking Afghanistan by remote control. CNN.com, 9 July 2008, <http://www.cnn.com/2008/TECH/07/09/remote.fighters/index.html>. Accessed 30 July 2010
- C. Von Clausewitz, On the art of war, in *The Morality of War: Classical and Contemporary Readings*, ed. by L. May, E. Rovie, S. Viner (Upper Saddle River, NJ: Pearson-Prentice Hall, 1832 (2005)), pp. 115–121
- M. Walzer, *Just and Unjust Wars*, 4th edn. (Basic Books, New York, 1977)

Moral Predators: The Duty to Employ Uninhabited Aerial Vehicles

123

Bradley Jay Strawser

Contents

123.1 Introduction..... 2944

123.2 Remote Weapons as Ethically Obligatory..... 2945

123.3 Objection 1: The Move to Independent Autonomous Weapon Systems..... 2950

123.4 Response..... 2951

123.5 Objection 2: UAV Limitations Lead to *Jus in Bello* Violations 2952

123.6 Response..... 2952

123.7 Objection 3: Cognitive Dissonance for UAV Operators 2954

123.8 Response..... 2954

123.9 Objection 4: Targeted Killing by UAVs 2955

123.10 Response..... 2955

123.11 Objection 5: UAVs Create Unjust Asymmetry in Combat 2957

123.12 Response..... 2957

123.13 Objection 6: Reduction of the *Jus Ad Bellum* Threshold 2959

123.14 Response..... 2959

123.15 Conclusion..... 2962

References 2963

Abstract

A variety of ethical objections have been raised against the military employment of uninhabited aerial vehicles (UAVs, drones). Some of these objections are technological concerns over UAVs’ abilities to function on par with their inhabited counterparts. This chapter sets such concerns aside and instead focuses on the supposed objections to the use of UAVs in principle. Several such objections

From the Journal of Military Ethics, vol. 9, no. 4 (2010); reprinted by permission of the publisher (Taylor and Francis, Ltd., <http://www.tandf.co.uk/journals>).

B.J. Strawser
Naval Postgraduate School, Monterey, CA, USA
e-mail: bjstraws@nps.edu

currently on offer are examined and shown all to be wanting. This chapter instead defends the view that society has a duty to protect an agent engaged in a justified act from harm to the greatest extent possible so long as that protection does not interfere with the agent's ability to act justly. UAVs afford precisely such protection. Therefore, a nation is obligated to employ UAV weapon systems if it can be shown that their use does not significantly reduce a war-fighter's operational capability. Of course, if a given military action is unjustified to begin with, then carrying out that act via UAVs is wrong, just as it would be with any weapon. But the point of this chapter is to show that there is nothing wrong in principle with using a UAV and that, *ceteris paribus*, using such technology is, in fact, obligatory.

123.1 Introduction

Lethal employment of unmanned (or, as U.S. Air Force personnel prefer, "uninhabited") aerial vehicles (UAVs) has risen precipitously by a few Western nation-states (most notably the United States) across several theaters of operation (Afghanistan, Pakistan, Yemen, and other locations). The emergence of this technology has sparked widespread debate over the ethical justification of its use. Some claim these drones create a particularly asymmetrical form of warfare that is somehow ignoble or dishonorable. Others contend that UAVs impede certain *jus in bello* principles. Some claim that drones create psychological conflicts for their operators (who are often thousands of miles away) causing unacceptable cognitive dissonance in the mind-set of the warrior. Still others raise concerns over drones carrying out targeted killings by nonmilitary government agencies (such as the CIA) and other concerns over their present employment. There is a worry that UAVs could lead to autonomous weapons that make lethal decisions on their own. Finally, some argue that by removing the pilot from the theater of combat, a degree of asymmetrical warfare is attained such that the risk threshold for a given state is lowered too far – that it becomes too easy for a state using drones to go to war; thus, their use is ethically pernicious.

This chapter defends the unorthodox view that there is an ethical obligation to use UAVs. In principle, there is no need for special ethical concern for this weapon system as opposed to any other more standard weapon technology. All of the concerns just listed either miss their mark and do not challenge the ethical obligation to employ UAVs in principle or else do not rise to the level needed to override the principles which form the basis of ethical obligation for UAV employment. Remotely controlled weapon systems are merely an extension of a long historical trajectory of removing a warrior ever farther from his foe for the warrior's better protection. UAVs are only a difference in degree down this path; there is nothing about their remote use that puts them in a different ethical category. My argument rests on the premise that if an agent is pursuing a morally justified yet inherently risky action, then there is a moral imperative to protect this agent if it possible to do so, unless there exists a countervailing good that outweighs the protection of the agent. Thus, I will contend that, as a technology that better protects (presumably) justified warriors, UAV use is ethically obligatory, not suspicious.

After some preliminaries, I will first present the argument for the ethical obligation to use remotely controlled weapons. Then I'll walk through the various ethical concerns which are supposed problems for UAV implementation and show how each of these worries is misplaced or fails to adequately counter the ethical obligation for their use.

123.2 Remote Weapons as Ethically Obligatory

Media coverage and public debate over the military use of uninhabited remotely controlled weapons is currently *en vogue*. It is surprising then, given such a backdrop, that the case for the ethical obligation to employ UAVs has yet to have been definitively made. That is precisely what I intend to do. First, some distinctions must be made regarding what the target of my claims in this chapter will be. Primarily, I am referencing those aircraft presently employed by the U.S. (and other) militaries commonly known as “unmanned aerial vehicles” or drones. To avoid unnecessary gender bias, I prefer the locution of uninhabited aerial vehicles (UAVs) which I will use throughout. Examples include the General Atomics MQ-1 Predator and the General Atomics MQ-9 Reaper. UAVs have been employed for some time as reconnaissance aircraft, but only fairly recently have such platforms been used for lethal engagement. Critically, when referencing UAVs I only intend those aircraft which are under human control for, in the minimum, any particular lethal action the machine executes. Autonomous weapon systems, which can execute lethal actions apart from a human decision to do so – that can operate “on their own” – will be addressed below in objection 1. Finally, my discussion here regarding the ethical obligation to employ UAVs could be applied, *mutatis mutandis*, to any remotely controlled lethal weapon system, including land- or sea-based remotely controlled weapons.

I contend that in certain contexts UAV employment is not only ethically permissible but is, in fact, ethically obligatory. The basis for this claim rests upon what I call the principle of unnecessary risk (PUR). The PUR proceeds as follows: If X gives Y an order to accomplish good goal G, then X has an obligation, *ceteris paribus*, to chose a means to accomplish G that does not violate the demands of justice, make the world worse, or expose Y to potentially lethal risk unless incurring such risk aids in the accomplishment of G in some way that cannot be gained via less risky means. That is, it is wrong to command someone to take on *unnecessary* potentially lethal risks in an effort to carry out a just action for some good; any potentially lethal risk incurred must be justified by some strong countervailing reason. In the absence of such a reason, ordering someone to incur potentially lethal risk is morally impermissible. Importantly, the PUR is a demand not to order someone to take unnecessary risk *on par* with alternative means to accomplish some goal G. This is what the *ceteris paribus* clause is meant to capture. That is, in some cases, the only possible way to accomplish G will be to order Y to undertake a particular means which exposes Y to potentially lethal risk. In such cases, the PUR is not directly applicable; whether or not the order is justified must be determined on other grounds. The PUR simply demands that no *more* risk than is required for the accomplishment of G (no unnecessary risk) is ordered by X to be incurred by Y.

I take the PUR to be uncontroversial. In fact, it is possible that an even stronger form of PUR could be developed that morally bars not only potentially lethal risk but any risk of bodily harm whatsoever. Further, there may be a reflexive form of the PUR available that could entail self-regarding duties not to incur potentially lethal risk unnecessarily. But some may complain that an individual has the moral permission to incur lethal risk in carrying out act X in pursuit of good A even if the risk in no way aids the accomplishment of A (or some other good B) nor is demanded by justice. To avoid such controversy, I employ here the more modest form of PUR as I have developed it. So even if some wish to contend that it is morally permissible for an individual to take unnecessary potentially lethal risks upon his or herself in accomplishing some good, it still seems that the PUR holds with no problems, focused as it is upon commanding others to action. That is, if some argue that there are no moral prohibitions against recklessly endangering one's own life for no good reason, certainly morality demands that there is a strong moral prohibition against unnecessarily endangering another's life.

I will note here another important approach that could be used to argue for the obligation to employ UAVs over inhabited aerial vehicles. Namely, UAVs are, on par, cheaper to produce and deploy than inhabited planes that accomplish similar missions. Thus, the argument could run; we are obligated to spend as little shared resources as are necessary on any given collective venture (such as a military undertaking), since those resources are scarce and could be used for other worthy goals. A principle of unnecessary waste of scarce resources (PUWSR) could be formulated to capture the normative appeal of such an approach. The PUWSR would contend that by not employing UAVs to the greatest extent possible, militaries are wasting scarce resources and that UAVs should, therefore, be used in place of inhabited aircraft so as to be better stewards of the said shared resources. For, after all, any money not spent on a military venture could be allocated towards other important demands of social justice, such as (say) an egalitarian concern for equal opportunity of welfare. Such a principle, then, could be used to put normative pressure on the financial budgets of Western militaries and demand that efficiency of cost is an important moral issue. I laud such approaches – and I find financial concerns to be particularly relevant in the case of UAV underemployment – but in this chapter I set aside such arguments and focus instead on what I see as the stronger normative principle of unnecessarily risking an agent performing a morally justified act. I do this because appeals to principles like the PUWSR, while legitimate, are often more easily overridden by other competing normative concerns. That is, even a relatively significant cost difference between two competing methods for carrying out a given act could quickly become moot were there any relevant differences warranting moral concern between the two courses of action. Of course, in this case (UAVs versus inhabited aircraft) whether there are such differences will often be an empirical question. And if, as I assume in this chapter, UAVs can carry out similar missions without any significant loss in capability, then concerns over cost would apply just as well. But I still view the PUR as a stronger moral claim – one that demands a higher justificatory bar to override – than principles such as the PUWSR; thus, it is upon the PUR that I base my central claims in this chapter.

Returning then to the PUR, an example may help demonstrate its modest moral demands and *prima facie* appeal. Imagine a group of soldiers are fighting in a just war against an unjust enemy. The (unjust) enemy soldiers are, say, invading the just soldiers' country and committing horrific crimes against humanity in the process. In the defensive effort a group of just soldiers, led by Captain Zelda, engage the enemy who are a short 50 yards away. Assume that engaging these enemy soldiers is a just action in pursuit of some good (in this case the good of defending their homes, families, themselves, and other innocents). Captain Zelda has an idea. She decides to remove her bulletproof vest, throw down her rifle, and charge the enemies with nothing more than a large rock and chutzpa. She turns to the troops under her command and orders them to do likewise. Set aside whether or not such an action is morally permissible for Captain Zelda to pursue individually. Also assume that charging the enemy in this fashion would in no way aid in accomplishing the good of successfully attacking the enemies yet would dramatically increase the lethal risk her troops incur. The PUR says that it is morally impermissible for her to order her fellow troops in her squad to take off their bulletproof vests, throw down their rifles, and charge the enemy with only a rock since there is no good reason to do so. The PUR holds that it is morally impermissible for Captain Zelda to endanger the lives of her troops any more than is necessary for the accomplishment of good A. My argument below for the moral obligation to employ UAVs rests on the PUR as a sound moral principle.

Note that such an action as Captain Zelda's planned foolhardy charge *may* contribute to some other thing, Q, which she takes as a good, such as an adrenaline rush or perceived valor gained by taking such inordinate risks. In such a case, one could try to argue that the act passes the PUR since it aims at some other (purported) good. But the PUR is not in the business of determining whether or not certain goals are goods worthy of being sought. It is a structural principle that functions on permissible commands to others only after it has been determined what the morally proper good to pursue should be. So, granting a proper good, the PUR demands that one commands others to incur lethal risk (or increased lethal risk) only in pursuit of that good (or some equal or greater good) if it is necessary in the way defined. That is, the risk one orders another to incur must track exactly with the necessity of that risk in relation to the accomplishment of the purported good. In this case, we are agreeing that the good sought (or the good that *should* be sought) is the successful attacking of the enemy; hence, Captain Zelda's actions are impermissible by way of the PUR for they do not aid in the accomplishment of the proper good nor are they demanded by justice or some other good. If Captain Zelda engages in reckless warfare and orders others to do likewise not because of necessity but because of some personal pleasure gained by the excitement of risk-taking and combat, then we would conclude her actions to be morally impermissible for other reasons outside of the PUR. That is, it may very well be that Captain Zelda orders her troops to make such a brash charge in the pursuit of something she takes as a good. In that case, the reason her action is wrong is not due to the PUR but because she is mistaken that this is a good worthy of being sought and ordering others to seek (say, the adrenaline rush of risky combat). But, if we agree that the good that *should* be sought

is the attacking of the enemies, her orders are impermissible via the PUR because her commands in no way aid in the accomplishment of attacking the enemies even though they cause her troops to incur (greater) lethal risk.

Granting the PUR then, consider the following claim, OP:

(OP) For any just action taken by a given military, if it is possible for the military to use UAV platforms in place of inhabited aerial vehicles without a significant loss of capability, then that military has an ethical obligation to do so.

The position defended here is that OP is true. It could, of course, very well turn out that OP is only vacuously true because the antecedent is false. This chapter will not primarily be arguing for or against the truth of OP's antecedent, but instead assume it is true and argue that the normative consequent follows. The antecedent of OP could be false for any number of reasons. First, it could turn out to be technologically infeasible to transition some military aircraft into remotely piloted vehicles without a significant loss of capability, for any number of reasons. Or it could be near impossible to do so due to budgetary constraints. Further, it could be that the antecedent of OP is false because remotely controlled weapon systems cannot practice target discrimination as effectively as inhabited vehicles can, and this would constitute a significant loss of capability. Or it could turn out that for some as of yet unforeseen reason remotely piloted weapon systems are not as capable in some other manner as inhabited vehicles. In any such case, the antecedent is false, and OP is vacuously true. There are very good reasons to believe, however, that the antecedent of OP could be true and even likely, as I will mention at points below. The central aim of this chapter, however, is to establish that the normative consequent follows if the antecedent is true. Further, the antecedent of these claims is an empirical question – one that can be tested for its veracity. What I am investigating is whether there is any principled reason for *not* employing UAVs. I contend that there is not and further (based on the PUR) that there is a strong moral obligation to use them in place of inhabited aircraft. If there is such an obligation, then OP follows.

Note that the “in place of” criterion of OP is crucial for its derivation from the PUR. A given commander in a combat context is obligated by the PUR to order her troop to use weapon Z in place of W if and only if Z reduces the risk placed on that soldier in comparison with and as an alternative to W. It is the risk differential between options Z and W that is the source of the obligation. To put it another way, because Z exists and is presently available for the commander to order her troop to use *in place of* W, the commander is obligated *not* to order the use of W so long as Z is a viable alternative that meets the other criteria (such as not violating the demands of justice). That is to say, the ordering to use Z is (presumably) permissible in a just war-fighting context; it becomes obligatory only as an *alternative* to W. But, if only W exists (or is the only option for other reasons, such as the demands of justice), then it could very well be permissible to order the use of W. Both W and the less risky Z must be viable options for the obligation to use Z to instantiate via the PUR.

To build the case for OP's consequent, consider the following scenario. Two small towns, Prudentville and Recklessville, each have a local police force that includes a highly trained “bomb squad.” Each bomb squad has been very successful in disarming and disposing malicious explosive ordinance throughout the years with

few (but some) casualties. Recently, both towns acquired remotely controlled robots that can be used to disarm explosives while being operated from afar. Under the control of a trained explosive ordinance disposal (EOD) technician, these robots are just as capable at disarming bombs as the EOD techs are themselves working “hands-on.” And with the robots, of course, the EOD technicians are not at any risk of injury or death. After some initial experimentation to ensure use of the robots did not cost them any bomb-disarming capability, Prudentville decides to have their bomb squad use the robots in any situation where it was possible to do so. They viewed the decision as a “no-brainer”: Saving the life of even one bomb technician would be well worth the cost of using the robot. Recklessville decides not to have their EOD techs use the robots, even though they have them available and are capable of doing so. Thus, they put their bomb techs at risk for no reason (or no *good* reason, at any rate) and violate the PUR.

Take the above story as a guiding normative analogy for claim OP. If it is possible for the bomb squad to use a robot to defuse the bomb remotely, with no significant or relevant loss of capability, then via PUR the bomb squad has a clear ethical obligation to use the robot in place of a human handling the bomb directly. The situation is relevantly analogous with the current and future use of remotely controlled military aircraft. That is, if it is possible for a state to have its military use remotely controlled weapon systems to carry out combat missions instead of inhabited weapon systems, with no significant or relevant loss of capability, then via PUR (assuming military missions carry potentially lethal risks) the state has a clear ethical obligation to do so. This is simply because by operating at a much greater distance from combat, the operator of the weapon system is exposed to significantly less risk. And if there is no compelling reason to expose a soldier to risk, then it is wrong to do so. Hence, OP.

One important caveat: The justification of remotely controlled weapons in war here assumes that their employment is done as part of a fully justified war effort meeting both *jus ad bellum* and *jus in bello* criteria. Thus, if the military in question is justified in a particular military strike in the first place, they should protect the just warrior carrying out the action to the greatest extent as is possible – up until protecting the warrior impedes her ability to behave justly in combat, as I will argue below. Granted, if a given military action is unjustified, then it is unjustified whether it is done by a pilot flying an aircraft remotely or otherwise. That is, my argument that the employment of UAVs is ethically obligatory follows out of the PUR in that a given military action in question must be a proper good in the first place. If the act is morally unjustified to begin with, then it is morally impermissible for other reasons outside of the scope of the PUR. Notice, for example, that this leaves open the possibility that universal pacifism may be the correct moral outlook towards warfare, and yet OP still holds (although vacuously, because a pacifist will hold that there simply are no justified military actions).

A related point is that some may here object that my analogy between a bomb squad and a military force fails for the bomb squad is trying to disarm a bomb, and thereby *prevent* the loss of life, whereas a military strike is attempting to *take* life. Yet the point of connection for the analogy is not what, specifically, the given action is attempting to carry out (be it disarming a bomb or delivering a bomb)

but simply that a particular action is justified and aiming towards *some* worthy good combined with being inherently risky to the agent engaging in the action. Again, the analogy to UAV use rests on a presumption that a given military strike employing a UAV is justified to start with – if it is not, then the UAV strike is morally impermissible, of course. So the case with the bomb squad is intended to focus on the moral principle of unnecessary risk in the execution of *some* good. The bomb squad, commanded by their town, undertakes morally justified but risky action F aiming to accomplish good goal G. If G can be accomplished just as effectively but with less risk to the bomb squad by a means other than F (such as by using a robot), then there is a moral obligation to use the robot. The same reasoning applies, *mutatis mutandis*, for a given military force. A military, commanded by their state, undertakes morally justified but risky action F aiming to accomplish good goal G. If G can be accomplished equally as effectively but with less risk to the military members (such as by using an uninhabited drone), then there is a moral obligation to use the drone. That G for the bomb squad case is the protection of life while in the UAV case G is the taking of life is not a morally relevant difference for the analogy. What matters is that G is a good worthy of pursuit.

To put the position another way still: Ordering a war-fighter to take on risk in any activity must be justified. If a given action can be equally well accomplished via two different methods, one of which incurs less risk for the war-fighter's personal safety than the other, then a justification must be given for why this safer method is not used. If there is no good reason not to use it, then we are obligated to employ the safer method. For all cases of ordering a war-fighter to undertake any given risky action, there should be a reason that demonstrates why the risk is necessary for the accomplishment of the given objective. If one grants that removing a pilot from the theater of combat by using a UAV instead of an inhabited weapon platform greatly reduces the risk to that pilot, then there should be a presumption for using a UAV (or any remote weapon) whenever it is possible to do so in a way that does not compromise the capability of a given warrior to behave justly. The burden of proof, then, is on those who argue that we should not employ UAVs or a similar remote technology. Such a position needs to justify why we should have pilots take on such risk. As mentioned above, there are a variety of objections usually offered as to why UAV employment is ethically suspicious. I shall now review each of these in turn and show why they fail to overcome the claim that UAVs are, in principle, ethically obligatory.

123.3 Objection 1: The Move to Independent Autonomous Weapon Systems

Some worry that UAVs lead us down a road towards independent autonomous weapons (IAWs), robots that make lethal decisions on their own. Where to draw the line on when a weapon system is “autonomous” is notoriously difficult. For simplicity's sake, one might consider any weapon that makes a decision on its own accord to launch a particular lethal attack as “independently autonomous” (or

“fully” autonomous as is sometimes used). Thus, a cruise missile that guides itself to a target would not be an IAW because a human agent made the decision to launch it and for it to go attack the given target, but a Predator drone programmed so as to make the particular decision to fire on a specific target on its own accord would become an IAW. So long as there is a “human in the loop” (to use the common military parlance) for each particular lethal decision, the system may be considered nonautonomous for the purposes of this chapter. That is, so long as a human agent makes the decision whether or not to employ lethal force, the weapon is not an IAW as I use the term. The objection’s argument against the employment of UAVs runs like this: IAWs are morally impermissible. UAV development will lead to IAWs. Therefore, UAV development is impermissible.

123.4 Response

As an objection against UAV usage goes, this fails to counter the moral obligation for their employment. In fact, we can grant the first premise (that “IAWs are morally impermissible”) but dispatch the objection by arguing that its second premise is presently unsubstantiated (that “UAV development will lead to IAWs”). One could agree with the objection that we should not develop IAWs and that we should not allow development of UAVs to lead us down the road towards IAWs. Indeed, it is plausible that it could be difficult to stop such progression, but it is not true that development of UAVs will *necessarily* lead to the development of IAWs. Thus, we need empirical evidence to show that this is the case. The objection is a kind of slippery slope objection because it assumes that the development and employment of UAVs must lead to the development and deployment of IAWs. Slippery slope objections are problematic because they fail to acknowledge a plausible middle ground stopping point. Namely, this objection misses the possibility of maintaining the employment of UAVs while at the same time working for the banning of IAWs (something I recommend Western nation-states do). Thus, at present, this objection fails as an argument against the ethical obligation to employ (and develop) UAV technology.

I raise this objection first so as to make an important distinction for the scope of this chapter. In this chapter, I am only arguing for the moral obligation to use remote weapons that are explicitly nonautonomous, at least regarding any lethal decisions. On my view, the distinction between IAWs and nonautonomous remote weapons is of paramount importance in this debate and is often neglected. One reason it is so important is that if this distinction is neglected and, even more importantly, if this distinction is not enforced and efforts to develop IAWs are not stopped, then objection 1 stands (assuming that one grants its first premise). That is, to be clear, it is entirely possible that the use of UAVs will in fact lead to the use of IAWs. If this can be shown to be the case and if it cannot be stopped, then, since I do grant the first premise, I see it as a legitimate objection against the employment of UAVs. But my hope is that the development of IAWs can be stopped even while UAVs are employed and developed. I do not here have space to argue against the moral

permissibility of IAWs – that has been done effectively elsewhere (see Sparrow 2007; Asaro 2006, 2007b; Himma 2007).

Some may object that my acceptance of the premise that “IAWs are morally impermissible” is inconsistent with my use of PUR to ground the moral obligation to use UAVs. The objection would contend that many weapon systems which could (arguably) be considered IAWs offer far better protection of a just war-fighter and are thereby obligatory via the PUR. Examples could be weapon systems such as the Phalanx CIWS or the SeaRAM employed by the U.S. Navy when they are used in fully autonomous mode. Without such weapon systems many sailors would potentially be at unnecessary risk, or so this objection claims. But this objection fails to appreciate that the PUR, although a strong *prima facie* moral principle, can be overridden by a strong enough countervailing normative reason. In this case, although I do not argue for it here, I find the principled objections to IAWs to be sufficiently strong such that they override the moral demands of the PUR. That is to say, it is perfectly compatible and in no way logically inconsistent to hold (as I do) that some nonautonomous weapon systems (such as UAVs) are obligatory via the PUR and at the same time hold that IAWs are impermissible on grounds specific to their autonomous nature which overrides the PUR. In any case, regardless of whether or not one accepts the first premise of objection 1, the objection on the whole fails because it is a slippery slope argument that is inadequately substantiated.

123.5 Objection 2: UAV Limitations Lead to *Jus in Bello* Violations

Some grant that remotely controlled weapons better protect the just war-fighter but argue that they do so at the cost of a decreased ability to discriminate combatants from noncombatants and other *jus in bello* compromises.

123.6 Response

Certainly, if an UAV operator engaging the battlefield from thousands of miles away through a video feed is unable to properly adhere to the *jus in bello* principles of discrimination and proportionality, then such drones should not be used. Indeed, if using a UAV in place of an inhabited weapon platform in anyway whatsoever decreases the ability to adhere to *jus in bello* principles, then a UAV should not be used. This is consistent with OP since adhering to principles of discrimination and proportionality is a key aspect of a weapon system’s capability. And the just warrior’s increased protection (which a UAV provides) should not be bought at an increased risk to noncombatants. Martin Cook (2004) makes this point effectively when he discusses the NATO air campaign waged in Kosovo. It seemed to some that by conducting missions at 15,000 ft, NATO was more concerned with force protection than noncombatant discrimination (see Cook 2004: 126–127). Had the combat missions been flown at a lower altitude, they would have put the pilots

at more risk but would have been significantly better at discriminating between, say, an ambulance and a military transport. It is the duty of the just war-fighter, I contend, to take additional risk upon herself if such risk is required in order to better shield innocents from harm. Thus, in arguing for OP, part of the assumption of the antecedent is that the use of UAVs does not hamper the war-fighter's (technical) ability to discriminate between combatants and noncombatants nor make judicious decisions of proportionality. Such a technical weakness would constitute a "significant loss of capability."

However, there is good reason to think just the opposite is true: that UAV technology actually *increases* a pilot's ability to discriminate. For example, the Israeli government-owned Rafael Armament Development Authority claims that with the new Spike ER precision missile, which is designed to be used by UAVs, they have achieved "urban warfare precision" (Rafael Advanced Defense Systems 2010). The missile can be launched in a fire, observe, and update mode (as opposed to a "fire and forget" mode) that "allows the UAV operator to update the missile, aim, point, or steer the missile off course if the intended target turns out to be a civilian" (Rafael Advanced Defense Systems 2010). The report goes on to quote an Israeli pilot who has used the weapon system: "The beauty of this seeker is that as the missile gets closer to the target, the picture gets clearer... The video image sent from the seeker via the fiber-optic link appears larger in our gunner's display. And that makes it much easier to distinguish legitimate from non-legitimate targets" (Rafael Advanced Defense Systems 2010).

And recent studies bear out that UAVs appear to have, in fact, greater technical capabilities at making determinations of combatant status. Avery Plaw (2010) has recently compiled a database combining reports from a variety of sources on the results of U.S. UAV attacks carried out in Pakistan from 2004 to 2007. This data shows that UAV strikes were far better at noncombatant discrimination than all other methods used for engaging Taliban fighters in the region. For example, the UAV strikes resulted in a ratio of over 17 to 1 of intended militant targets to civilian deaths compared with a 4 to 1 ratio for Pakistan SWAT team offensives or a nearly 3 to 1 for Pakistan Army operations in the same region during the same time period. Or, compare the 17 to 1 ratio for the UAV employment to the shocking 0.125 to 1 militant to civilian causality ratio estimate for all armed conflict worldwide for the year 2000 (Plaw 2010). If these numbers are even close to accurate, it seems that there is strong evidence which directly contradicts the central premise of objection 1. That is, UAVs are better, not worse, at noncombatant discrimination.

Regardless, however, whether or not UAVs are as technically capable of making determinations of proper target discrimination is an empirical question. If it turns out that UAVs are not as capable, then OP's antecedent is false, and the claim is vacuously true. At present, however, all available evidence points strongly towards there being no reduction in the technical ability for UAV pilots to discriminate as opposed to inhabited aircraft pilots' ability. But, being that this is an empirical matter, there is no in-principle objection here to UAVs being ethically obligatory for military use.

123.7 Objection 3: Cognitive Dissonance for UAV Operators

This objection worries that the use of drones leads to psychological conflicts for their operators causing cognitive dissonance in the mind-set of the warrior. The worry can manifest two separate ethical concerns, first that it is wrong to do this to UAV operators – for them to kill the enemy from their “desk” at work and then go home to dinner and their child’s soccer match – and that this places an unjust psychological burden on them. The second and greater concern is that this cognitive dissonance will weaken the operator’s will to fight justly in a number of ways (e.g., the operator not taking the warfare as “real” or serious enough but instead viewing it as a video game and the operator suffering mental problems and PTSD which, because of their distance from the battlefield, could go untreated and unrecognized, causing further problems and leading to inappropriate decisions).

123.8 Response

The argument that the ethical justification for UAVs is threatened if UAV operators are more likely to behave unjustly in their combat actions due to this cognitive dissonance is unsound. First, it can be argued that the temptation for the war-fighter to commit *jus in bello* violations would actually lessen, perhaps significantly so, once the war-fighter is not at risk. The remote pilot can take more time in evaluating a target before firing – to ensure that target is an enemy combatant – than they would be able to otherwise; for in the worst case scenario, a machine is lost, not a human pilot. Returning to the bomb squad analogy, in using a robot, the EOD technicians do not experience the same level of stress because there is no danger to themselves; thus, they are not as nervous and, presumably, more successful. The same could hold true for UAV pilots making judicious decisions in combat. Once fear for their own safety is not a pressing concern, one would assume the operator would be more capable, not less, of behaving justly.

But perhaps not. Maybe the distance and disjunct of this level of remote weaponry do create a significant and genuinely new kind of stress on the war-fighter that might compromise their abilities to behave justly. There is significant empirical work here yet to be done. But even if we grant that displaced combat negatively affects UAV pilots’ abilities, first note that there are means of overcoming this problem and, second, that this issue is not a knock against the ethical justification of UAVs themselves. If necessary we could, for example, move all UAV operators much closer to the theater of combat, forcing them to live in a deployed environment, along the same time zone as the combat, and under more standard battlefield conditions and stresses. Further, note that all UAV action has the ability to be recorded and monitored. By default since it is remotely controlled, whatever data feed a UAV pilot received can easily be overseen by many others simultaneously and later for review and critique. This added level of accountability could be used to get, if necessary, further added layers of scrutiny over lethal decision making – even

demanding more than one officer agree to a kill, for example. Indeed, an entire team of officers and human rights lawyers could oversee every single lethal decision made by a UAV, if desired or deemed necessary. The point is that there are a variety of ways to overcome any concerns that the pilots of UAVs would be somehow less judicious on average than inhabited weapon systems would be. All of these argue against this cognitive dissonance problem as being somehow insurmountable, much less negating the ethical obligation for UAV use in principle. Moreover, even if there is some psychological harm done to UAV pilots that we cannot overcome, it certainly seems that such harm would be less damaging than the expected harm that could come about via inhabited flights.

123.9 Objection 4: Targeted Killing by UAVs

Recent press has raised concerns over the use of UAVs for targeted killings, particularly as is currently being done by the CIA in Pakistan, Yemen, and other theaters of operation. The specific objection is that assassinations fall outside the bounds of acceptable just-war theory practice and that UAVs somehow make this practice too easy or contribute to it in some unacceptable manner.

123.10 Response

Although ethical concerns over assassinations and further underlying concerns regarding a nonmilitary government agency carrying out independent lethal operations in a foreign theater are both of great moral significance themselves, none of these concerns are endemic in any significant way to remotely controlled weapon systems. The CIA could be carrying out these same missions with a trained sniper or an inhabited aircraft. It is this particular *policy* that is of proper ethical concern here, not UAV technology or use in general. Some might argue, however, that the UAV makes targeted killing of this sort particularly pernicious because, first, an aerial vehicle flying over airspace is in some principled way different than sending in a ground special ops troop. Second, the objection claims that the battle for the “hearts and minds” of local nationals in a given theater is significantly worsened by what they view as ignoble warfare; UAVs are thought to be “cowardly.” And, third, the objection continues, there are some ways in which UAV technology makes such policies easier to execute because of the abilities unique to current UAV platforms.

As to the first concern, this is admittedly an interesting case that could appear to be peculiar to UAVs. Namely, if a nation-state sends a UAV over another sovereign nation-state’s airspace, they have not sent an actual person or agent over the airspace. This could perhaps leave room for a contrived argument that because no actual person crossed a border, no infringement on national sovereignty occurred. Although I am intrigued by this distinction for UAV weaponry, I do not find it persuasive. For a UAV strike in terms of sovereignty issues amounts analogously to a long-distance artillery shell fired across the border or other forms of attack

that do not involve an agent crossing an actually geographic border such as cyber warfare. In such cases, yes, no actual person violated the territorial integrity of the sovereign state in question, but, of course, all nations would still (rightly) view such acts as a direct violation of their sovereignty. So, contra the worry, UAVs do not create a special class of weapons that can escape traditional just-war theory scrutiny or respect for territorial integrity and national sovereignty through an odd loophole.

As for the second concern, two points are in order. First, I would argue that it is at least possible that if UAVs are used in line with the rules of warfare, and civilian casualties are not increased (and perhaps even lessened) due to their usage, then there might be no *greater* resistance from a local populace than would be encountered for more conventional weapons. There is some empirical evidence (albeit limited) to back up this possibility (Plaw 2010). Further, the possibility has some intuitive plausibility when we note that the majority of negative responses to UAVs by local populations have come, as usual, when they have inadvertently hit civilian targets – but we have seen this same response in other conflicts when similar strikes were delivered from (say) a B-52 flying at altitude dropping munitions. Again, this seems to point to the possibility that the particular platform dropping the bomb (inhabited or uninhabited) is not what generates a negative response from the people below but whether the attack was justified and hits legitimate targets.

But perhaps this response fails. There is, admittedly, some strong empirical evidence suggesting just the opposite: that local populations' particular resistance to UAVs is precisely due to the fact that they are uninhabited. But so be it. For even if the first response fails, recall that my argument for the ethical justification of UAVs requires that there be no reduction in just war-fighting capability. So even if it *does* turn out that in a given theater of operation UAVs do, in fact, cause significantly greater resistance from the local populace as compared to the resistance that similar inhabited vehicles would generate (perhaps because the population thinks they are cowardly or some similar response), then they should not be used on OP grounds. Such a limitation would clearly fall under the "significant loss of capability" clause of OP. And, of course, this is an empirical question, not an in-principle objection to UAVs.

The third concern – that UAV technology makes such actions easier to carry out – similarly does not offer a principled objection to the moral obligation to use UAVs. It is true that the extended ability of platforms such as the Predator to "hover" and stay in a localized area for long hours, even days, and observe targets, is a clear combat advantage. Many inhabited aircraft do not have such capacities. And, further, some of the remote areas where such strikes are carried out by UAVs are such that they would be inaccessible to similar inhabited weapon platforms. But these facts about the superior capabilities of UAVs do not count against OP. Just as the advent of airpower brought with it many new and often superior ways war-fighters could engage in combat (both justified and not), such advantages do not imply anything inherently wrong with airpower *qua* airpower. Further, the mere existence of such advantages does not force policy makers to misuse these capabilities. Certainly, it would be impossible to drop bombs on innocent civilians if planes did not exist. But that some drop bombs on innocent civilians do not make airplanes morally

suspicious but rather those who so use them to drop bombs. The same holds true for the new capabilities brought about by UAVs.

Thus, there is nothing peculiar to UAVs in regard to the ethical concerns over their present use in targeted killings around the globe. It is the morality of the United States' recent policy of targeted killings we must debate here, not the ethical justification of UAVs.

123.11 Objection 5: UAVs Create Unjust Asymmetry in Combat

This objection normally runs as follows: The use of remotely controlled weapons by one force against another force that does not have similar technology crosses an asymmetry threshold that makes the combat inherently ignoble. That is, the extent of asymmetry in the combat abilities between two opposing sides becomes so great when one side employs remote weapons that the fight is intrinsically unfair and that, in turn, makes the use of the said remote weapons morally impermissible. This position is usually held because in such circumstances one side literally does not take *any* life-or-death risks whatsoever (or nearly so, since its war-fighters are not even *present* in the primary theater of combat), whereas the opposing side carries all the risk of combat.

123.12 Response

As an objection against the ethical justification for remotely controlled weapons broadly, and UAVs in particular, this commonly heard argument fails. First, if someone holds that justified combat should be a "fair fight" between sides, at least to some degree, then I would argue that contemporary military engagements crossed that threshold long ago. How fair is the present fight between an F-22 pilot flying at altitude delivering a precision missile and a tribal warrior wielding an RPG? If there is a moral problem here due to asymmetry, it seems to have occurred long before UAV implementation and is not endemic to them. But, second, even *if* the actual removal of the warrior from the theater of combat represents a truly new level of asymmetry in combat (and perhaps it does), this alone is still no argument against doing it. This is because if one combatant is ethically justified in their effort, and the other is not, then it is *good* that the just warrior has the advantage and is better protected.

Here I am following Jeff McMahan's recent work rejecting the moral equality of combatants (see McMahan 2009). That is, the warrior fighting for a just cause is morally justified to take the life of the enemy combatant, whereas the unjust fighter is not justified, even if they follow the traditional principles of *jus in bello* such as only targeting combatants and the like, to kill the justified fighter. Thus, there is no chivalrous reason for a just combatant to "equal the playing field" or "fight fair." If combatant A fights under a just cause, while combatant B fights for an unjust cause, combatant A owes nothing to combatant B by way of exposing herself to

some minimal threshold of risk. Thus, it is *right* for combatant A to reduce their risk in their engagement with the unjust enemy.

But even if one disagrees with McMahan's position and the rejection of the MEC, they still do not have grounds to object to the protecting of a soldier under the "fair fight" objection. An MEC advocate would still presumably agree that a military pursuing a justified action as part of a just war is justified to do all they can to protect their soldier so long as that protection does not hinder the soldier's ability to follow *jus in bello* principles. The only difference is that an MEC advocate will think the unjust aggressor state enjoys the same allowance to protect their war-fighter similarly. That is, even if one thinks that soldiers enjoy a symmetrical position of the right to individual defensive measures in a given conflict, this in no way prevents either side from maximizing their personal defense so long as it is not at the cost of *jus in bello* precepts; indeed, such precepts (under MEC) would explicitly allow it. Thus, again, the argument for a "fair fight" fails on two counts. First, it is already overcome by earlier technological advancements because present military operations are already far from fair even without the asymmetry of UAV weapon systems and thus the issue here is not with UAVs properly speaking. And, second, the desire for a "fair fight" is simply a weak claim in the first place, something akin to an archaic demand of military commanders in the eighteenth-century warfare to line up their troops across from one another for a "dignified battle." There is simply no normatively compelling reason to think a justified military force need have a fair fight anymore than we would think a police force ought not use bulletproof vests to offer dangerous criminals a fair fight.

But perhaps this still does not give the objection its due. Paul Bloomfield once remarked that simply the idea of "being killed by remote control" is powerful and disturbing. The intuition seems to be that killing someone in such a manner is profoundly disrespectful, that a human being deserves to be able to at least point at his or her killers (and condemn them, if they are unjust) even if his or her killers are cruising 20,000 ft above in a plane. The thought is that at least a human being in a plane high above is less of a "faceless" death wrought upon someone than a robot being operated remotely would be. Or consider the sentiment Uwe Steinhoff raises in discussing remote weaponry generally and how the odd risk asymmetry it creates (making the targets of attack "defenseless") does not feel like "honorable" warfare:

To be sure, I do not deny that there is something fishy about attacking the defenseless. What is fishy about it might be captured very well in this passage: 'The pilot of a fighter-bomber or the crew of a man-of-war from which the Tomahawk rockets are launched are beyond the reach of the enemy's weapons. War has lost all features of the classical duel situation here and has approached, to put it cynically, certain forms of pest control.' (Steinhoff 2006: 7).

It must be admitted that there does appear something ignoble or dishonorable in such a vision of warfare as "pest control" that Münkler's quote describes. Perhaps it is that such distance makes warfare seem too clinical or coldhearted. Many will have sympathy with such a sentiment when envisioning UAV warfare – myself included. But whatever this sentiment is, it does not amount to a normative argument; such a "feeling" does not constitute a moral reason for rejecting UAV use.

Something being disturbing does not by itself make it wrong. This sense of the ignobility must be elucidated into a coherent and compelling ethical argument against using UAVs; mere displeasure at imagining their employment does not help us. As Steinhoff writes:

Judged from a traditional warrior's code of honor, a code that emphasizes, among other things, courage, there is nothing honorable in killing off defenseless enemies (whether it is therefore already *dishonorable* is yet another question). But honor and morality are not the same, and honor and the laws of war are not either. In short, the prohibition of assaults upon the defenseless is neither an explicit nor an implicit principle of the laws of war or of just war theory. (Steinhoff 2006: 8).

Steinhoff is certainly right in this. I would add that a crucial element in how one “feels” about imagining such warfare depends on whether or not the precision missile strike in the picture envisioned is justified or not. Is it a military strike as part of a fully justified defense against an aggressing, unjustified, destructive enemy force? Is the strike hitting a legitimate and morally culpable target? If it is, such factors temper our view of the strike considerably and move us away from the “pest control” picture. In such a case, we should desire that the just warrior be well protected from any possible threat that this enemy might proffer – protection that the UAV affords.

123.13 Objection 6: Reduction of the *Jus Ad Bellum* Threshold

The worry here is that the asymmetry in combat abilities created by the advanced technology of UAVs, and in particular by the massive reduction of risk to the UAV pilot, makes it too easy for the nation employing UAVs to go to war. That is, the asymmetry created by UAVs lowers the *jus ad bellum* threshold such that more unjust wars might be conducted because the risks of war to a nation-state could become so minimal.

123.14 Response

This objection, on first glance, may appear to be the strongest objection to the implementation of UAVs. The worry that it will be easier to go to war if we have technology X and thus more tempting to enter into unjust wars (making more unjust wars more likely) is intuitively plausible. But this kind of argument ultimately fails for the objection does not succeed in negating the present moral imperative to use UAVs as derived from the PUR. To see why this is, consider two possible worlds, Alpha and Beta. In Alpha, nation-state Zandar has developed the technology to make bulletproof vests for its military members to wear in combat which significantly decreases the risks they incur in battle. Zandar, in accordance with the PUR, produces these bulletproof vests and has its military members wear them. In world Beta, nation-state Zandar has developed the same technology and has the bulletproof vests available. However, they reason that if they use these bulletproof vests, war

would “cost” them less in terms of risks to their own troops and, thus, be easier (and thus more tempting) to wage. In such circumstances, Beta-world-Zandar worries, more unjust wars are more likely. So they decide not to use bulletproof vests in order to make war more costly to wage (by intentionally increasing the risk to their soldiers) in the hopes that this will lessen the likelihood that they will engage in an unjust war in the future. Aside from this one decision, the worlds Alpha and Beta are identical.

Let us assume that it turns out Beta-world-Zandar’s reasoning was correct. That is, going forward from this juncture there does, indeed, end up being some greater number of unjust wars waged in world Alpha than in world Beta. The use of the bulletproof vests in some way lowered the threshold for going to war for Alpha-world-Zandar enough that it made a positive difference on the total number of wars fought – which included some unjust wars. I still contend that Beta-world-Zandar’s decision was morally impermissible. This is because the normative force of the PUR upon present actions is too strong to overcome such weak predictive calculations of future possibilities.

I will show why this is shortly, but first note that the scope of this issue far exceeds UAVs and bulletproof vests, of course, but strikes at *any* asymmetry in military technological development whatsoever. Any improvement to a given military’s capabilities that gives it an advantage over its potential enemies will face the same objection offered here against UAVs. But that would mean that this objection could be used to block the development and implementation of *any* military technology that creates any asymmetry. Further, the objection could actually be employed to work backwards: that current militaries should intentionally reduce military capabilities in order to make war more costly to them since doing so would place their soldiers at greater risk. Following this logic could even lead to the conclusion that a state should have their militaries throw away their weaponry and all defensive technology, for certainly a neutered military would be less likely to engage in unjust wars in the future. I grant that this worry about asymmetry created by improvements in military technology making it easier to go to war may well be a legitimate concern. But it is a logic that quickly runs to demanding no military technology whatsoever in the hopes of avoiding future unjust wars. Perhaps this is correct. Perhaps there should be no militaries. But notice that we are now a far cry from arguing over UAV technology. We are arguing over the existence of any military weaponry or advancement whatsoever. If so, then this is not actually an objection *specific* to UAVs in principle. Moreover, if objection 6 is correct in this way, then OP still stands – it is just a vacuous claim: as would be *any* claim about the possibility of justified use of military weaponry of any kind.

But the problems with the objection run even deeper. As I alluded to above, the reasoning by Beta-world-Zandar not to use the vests, notice, rests on epistemically dubious calculations that are predictive about *themselves* doing something wrong in the future (“we might be more likely to do wrong action X down the road”) over epistemically solid calculations to protect their own just war-fighters presently (“our soldiers will be safer *today* if they wear the vests”). Notice what odd moral reasoning would be occurring were objection 6 to work: *Because* we will most likely

behave unjustly in the future, we should behave unjustly in the present (by violating the PUR in choosing not to protect our warriors as best we can) in order to try to prevent ourselves from acting unjustly in the future. If that holds, we have a strange account of moral epistemology at work, to say the least. We should forego taking presently morally correct action A in order to help restrain our future selves from the likelihood of committing morally wrong action B. In other words, we should do something wrong now in order to (hopefully) better stop ourselves from doing something wrong in the future.

This seems odd, although there could perhaps be cases where such decisions are the right actions – the lesser of two evils, perhaps. But notice that the Beta-world-Zandar decision is not a straightforward case of present self-restraint to limit future potential wrongdoing for, presumably, *usual* cases of present self-restraint are not acts that are themselves impermissible. For example, imagine a man, Tom, who knows he tends to get very angry and do intemperate things. Tom decides he should lock up his gun in a safe and give the key to it to a trusted friend. Tom does this present act to restrain his future self from doing something wrong. But Tom's locking up his gun is not an impermissible act viewed on its own. Violating the PUR by not protecting just war-fighters is a presently impermissible act viewed in isolation. Thus what makes the reasoning of Beta-world-Zandar's decision so strange: They are intentionally putting their soldiers at greater risk now (which would be considered impermissible in isolation) in order to restrain themselves from doing something impermissible in the future. The comparison back to Tom would be if Tom decides to punch his friend Sam now (which is impermissible) because it will help him not do something worse in the future (such as kill Bob). If that is actually the case, then this *could* be the right thing to do. But notice that we would require a rather high level of epistemic certitude for Tom's knowledge of this scenario – and that there is no other means to avoid killing Bob – in order to deem his act justifiable. That is, if Tom has near *certainty* that the only way to prevent himself from killing Bob in the future is by punching Sam now, then perhaps it is a justified act. But one wonders how Tom could *ever* have such epistemic certitude predicting future acts. The same is true for Beta-world-Zandar.

But perhaps it is still possible that such a decision is justified. This is because, one could argue, I am here equivocating on the moral weight of the present wrong of failing to protect just war-fighters and the future potential wrong of more unjust wars. If they are of vastly different moral significance and consequence, then perhaps it is justifiable to do a lesser wrong now in order to increase even the slightest chance of avoiding a much greater wrong in the future. I grant this possibility. Indeed, one could argue that such a decision is directly in accord with the PUR since the good of avoiding future wars is a greater good than the present protection of just soldiers (i.e., some would argue that such a calculus is demanded by justice).

The trouble with applying this to our present case is the high degree of epistemic uncertainty we have in predicting future states of affairs, particularly future decisions to go to war. That is, even if the wrong of sacrificing the protection afforded to just soldiers is a lesser evil than the possibility of future unjust wars,

we have complete confidence in the *present* wrong occurring but we would be far less than certain of the future wrong occurring. In other words, the *odds* of that future wrong occurring will matter and it is unclear how we could reliably predict such odds. The odds need not be equal between the wrongs, of course. If the greater evil was great enough, we could perhaps need only a relatively small chance of its occurrence to outweigh even the certainty of a lesser evil. But, again, it is unclear if we can even have *that* small level of epistemic confidence that a given weapon technology (be it bulletproof vests, M-16 rifles, or UAVs) would lead to greater instances of unjust wars or at least not the level of confidence we would need to trump the demand not to commit the present wrong.

So even if it turns out that future worlds would be ones with less war were we to intentionally limit our own military technological development, we cannot have enough epistemic certainty in knowing this presently to overcome the demands to protect the just war-fighter. Short of a crystal ball, I cannot imagine how we could ever have the level of certainty for predictive knowledge claims of future group behavior that would be necessary to claim that the future *possible* good should outweigh our present moral duty not to unnecessarily risk others. Perhaps it can be done, but this must be demonstrated before we intentionally bring unnecessary risk upon others. That is, one would have to show how we can have such epistemic confidence that we are not violating the PUR (via the demands of justice override) in not presently protecting just war-fighters. The burden of proof will be on those claiming that we must presently undertake an act that we would usually consider impermissible in isolation in order to avoid a future evil that we do not have complete confidence in. Hence, although it is certainly possible that use of UAVs could lower the costs of going to war for a given state and, thereby, lower the threshold for going to war such that a state might have an increased likelihood of engaging in a war that is unjust, such predictions cannot be the basis for demanding an intentional violation of the PUR given our present epistemic limitations. This is an unhappy conclusion. While I have great sympathy for the worry, it seems the PUR is too strong to overcome with such shaky future predictions as to the unethical decisions a future state would make. And, again, if we allow this block against UAVs, it would set in motion a moral principle that would not stop at UAVs but encompass all military technology – not just its future development but retroactively demand that present military technology creating force asymmetry be intentionally reduced. If this is sound we could eventually be back at demanding that Captain Zelda *should* be required to fight with no bulletproof vest, no rifle, and with only a rock in order to make war “cost us more” so that we would be less likely to engage in an unjust one. But this is absurd. If a war is just, we are obligated to protect the just war-fighters engaging in it. UAVs do precisely that.

123.15 Conclusion

UAVs will have an increasingly large presence in military operations on all levels; this much appears increasingly inevitable. Here I have made the case that any new

technology that better protects the just war-fighter is at least a *prima facie* ethical improvement and is morally required to use unless there are strong countervailing reasons to give up that protection. I have argued that if using UAVs (as a particular kind of remote weapon) does not incur a significant loss of capability – particularly the operators’ ability to engage in warfare in accordance with *jus in bello* principles – then there is an ethical obligation to use them and, indeed, transition entire military inventories to UAVs anywhere it is possible to do so. In fact, I endorse the stronger claim that such a proposed transition would not only be feasible without a significant loss of capability but would actually *increase* weapon systems capability *and* the ability to fight justly. All of the concerns regarding UAVs presently on offer do not negate this ethical obligation to use uninhabited weapon systems and should be properly viewed instead as indictments against mistaken policy decisions and specific instances of force application – not as principled objections against UAVs themselves for none of the concerns are endemic to UAVs in any significant way.

Finally, this chapter defends the ethical obligation to use UAVs for a putatively just military action in the current context, whereas one might object that much, if not all, *actual* UAV employment is part of military actions that are morally questionable or outright impermissible. The particular contemporary circumstances and misuses of UAVs, however, do not trump the moral principles underlying the ethical obligation to employ UAVs for *just* actions. Indeed, this highlights the central point well: The first question for the morally permissible use of *any* weapon technology is, of course, whether the military action itself is morally justified. If it is not a justified undertaking in the first place, then it does not matter if it is carried out via a crossbow, a sniper rifle, or a UAV; it is morally impermissible regardless. If the act is morally justified, however, we are obliged via the demands of the PUR to protect the agent ordered to carry out that action as best we can, be it a police officer apprehending a dangerous criminal, an EOD technician disarming a bomb, or a justified warrior fighting an unjust enemy. Hence, the ethical obligation to employ UAVs.

References

- R. Arkin, *Governing Lethal Behavior in Autonomous Robots* (Chapman and Hall, New York, 2009)
- P. Asaro, What should we want from a robot ethic? *Int. Rev. Inf. Ethics* **6**, 9–16 (2006)
- P. Asaro, How just could a robot war be? Paper presented at 5th European computing and philosophy conference, Twente, June 2007a
- P. Asaro, Robots and responsibility from a legal perspective. Paper presented at IEEE international conference on robotics and automation, Rome, 2007b
- Y. Benbaji, A defense of the traditional war convention. *Ethics* **118**, 464–495 (2008)
- J. Card, Killer machines. *Foreign Policy*, p. 92 (2007)
- M. Cook, *The Moral Warrior* (SUNY Press, Albany, 2004)
- R. Dipert, The ethics of cyberwarfare. Paper presented at international society of military ethics annual conference, San Diego, Jan 2010
- P. Emerton, T. Handfield, Order and affray: defensive privileges in warfare. *Philos. Public Aff.* **37**, 382–414 (2009)

- J.R. Fitzsimonds, T.G. Mahnken, Military officer attitudes toward UAV adoption: exploring institutional impediments to innovation. *Jt. Forces Q.* **46**, 96–103 (2007)
- M. Gross, Assassination and targeted killing: law enforcement, execution, or self-defence? *J. Appl. Philos.* **23**(3), 323–335 (2006)
- K. Himma, Artificial agency, consciousness, and the criteria for moral agency: what properties must an artificial agent have to be a moral agent? Paper presented at 7th international computer ethics conference, San Diego, July 2007
- M. Kaldor, *New and Old Wars* (Stanford University Press, Palo Alto, 1999)
- W.R.P. Kaufman, Rethinking the ban on assassination, in *Rethinking the Just War Tradition*, ed. by M.W. Brough, J.W. Lango, H. van der Linden (SUNY Press, Albany, 2007)
- S. Kershner, The moral argument for a policy of assassination. *Reason Pap.* **27**, 45–67 (2004)
- S. Kershner, Assassination and the immunity theory. *Philosophia* **33**(4), 129–147 (2005)
- D.W. Kolff, 'Missile strike carried out with Yemeni cooperation' – using UCAVs to kill alleged terrorists: a professional approach to the normative basis of military ethics. *J. Mil. Ethics* **2**(3), 240–244 (2003)
- S. Lazar, Responsibility, risk, and killing in self-defense. *Ethics* **199**, 699–728 (2009)
- S. Lazar, The responsibility dilemma for killing in war: a review essay. *Philos. Public Aff.* **38**(2), 180–213 (2010)
- J. Mayer, The predator war: what are the risks of the CIA's covert drone program? *The New Yorker*, Oct 2009. Available at: http://www.newyorker.com/reporting/2009/10/26/091026fa_fact_mayer
- J. McMahan, The ethics of killing in war. *Ethics* **114**(4), 708–718 (2004)
- J. McMahan, Just cause for war. *Ethics Int. Aff.* **19**(3), 1–21 (2005)
- J. McMahan, Debate: justification and liability in war. *J. Politi. Philos.* **16**(2), 227–244 (2008)
- J. McMahan, *Killing in War* (Oxford University Press, New York, 2009)
- H. Münkler, *Die neuen Kriege* (Rowohlt, Reinbek bei Hamburg, 2003)
- J. Mustin, Future employment of unmanned aerial vehicles. *Air Space Power J.* **16**(2), 86–97 (2002)
- National Priorities Project, Cost of war (2010). Available at: http://www.nationalpriorities.org/costofwar_home. Accessed 1 Sept 2010
- Office of the Under Secretary of Defense, *Unmanned Aircraft Systems Roadmap: 2005–2030* (Office of the Under Secretary of Defense, Washington, DC, 2006)
- M. Osiel, *The End of Reciprocity* (Cambridge University Press, Cambridge, 2009)
- M. Phythian, Ethics and intelligence: the implications of the rise of the armed drone. Paper presented at 7th global conference on war and peace, Prague, 30 Apr 2010
- A. Plaw, Sudden justice. Paper presented at 7th annual global conference on war and peace, Prague, 1 May 2010
- Rafael Advanced Defense Systems, Spike ER precision missile (2010). Available at: <http://www.rafael.co.il/marketing/sipstorage/files/0/600.pdf>. Accessed 1 June 2010
- P.W. Singer, *Wired for War: The Robotics Revolution and Conflict in the 21st Century* (Penguin Press, New York, 2009)
- R. Sparrow, Killer robots. *J. Appl. Philos.* **24**(1), 63–77 (2007)
- R. Sparrow, Building a better warbot: ethical issues in the design of unmanned systems for military applications. *Sci. Eng. Ethics* **15**, 169–187 (2009)
- U. Steinhoff, Torture: the case for Dirty Harry and against Alan Dershowitz. *J. Appl. Philos.* **23**(3), 337–353 (2006)
- U. Steinhoff, Jeff McMahan on the moral inequality of combatants. *J. Politi. Philos.* **16**(2), 220–226 (2008)
- W. Wallach, C. Allen, *Moral Machines: Teaching Robots Right from Wrong* (Oxford University Press, New York, 2009)

Killer Robots: Ethical Issues in the Design of Unmanned Systems for Military Applications

Robert Sparrow

Contents

124.1 Introduction 2966

124.2 Building Safe Systems 2967

 124.2.1 Keeping Humans Out of Harm’s Way? 2968

 124.2.2 The Psychological Stresses on Remote Operators 2970

124.3 Designing for the Law of Armed Conflict 2972

 124.3.1 Can Robots Meet the Criteria of Discrimination and Proportionality? 2973

 124.3.2 Locating Responsibility 2974

 124.3.3 “Designing Out” War Crimes 2975

124.4 Conclusion 2979

References 2981

Abstract

Unmanned systems in military applications will often play a role in determining the success or failure of combat missions and thus in determining who lives and dies in times of war. Designers of UMS must therefore consider ethical, as well as operational, requirements and limits when developing UMS. The ethical issues involved in UMS design may be grouped under two broad headings: Building Safe Systems and Designing for the Law of Armed Conflict. This paper identifies and discusses a number of issues under each of these headings and offers some analysis of the implications of each issue and how it might be addressed.

This article was originally published in the Science and Engineering Ethics (June 2009, Volume 15, Issue 2, pp 169-187)

R. Sparrow
Department of Philosophy, Monash University, Canberra, Australia
e-mail: Robert.Sparrow@arts.monash.edu.au

124.1 Introduction

A significant proportion – perhaps even the majority – of contemporary robotics research is funded by the military. The US-led invasion and occupation of Iraq and Afghanistan demonstrated the military worth of – and greatly increased the demand for – uninhabited air vehicles (UAVs) such as Global Hawk, uninhabited combat aerial vehicles (UCAVs) such as Predator and Reaper, and also robots for bomb and IED disposal (Butler 2007; Hockmuth 2007; Office of the Secretary of Defense 2005a,b, 2012; Singer 2009). Unmanned systems (UMS) in the form of UAVs, UCAVs, unmanned ground vehicles (UGVs), unmanned undersea vehicles (UUVs), and unmanned surface vehicles (USVs) are currently being developed and deployed by nations including the U.S., Israel, South Korea, Britain, France, Germany, Denmark, Sweden, China, and India (Card 2007; Kenyon 2006; Legien et al. 2006; McMains 2004; Masey 2006; Office of the Secretary of Defense 2005a, Appendix A; Office of the Secretary of Defense 2005b, pp. 38–40; Peterson 2005). As a consequence, many engineers, computer programmers, and roboticists are now working on systems for military applications.

The use of robots in military applications generates a large number of important ethical issues (Veruggio and Operto 2006, 1517) – indeed, so many that they defy treatment in a single paper. Will these systems make conflict more likely by promoting the idea that wars can be waged without casualties? Will they encourage asymmetric warfare by rendering it impossible for any but the most high-tech of militaries to triumph in conventional battle? What are the implications for just war theory if nations can fight wars without putting the lives of their warfighters at risk and, conversely, what does just war theory have to say about the war being waged in this fashion? Will the development of robot weapons generate an “arms race to autonomy” thus effectively forcing the deployment of weapon systems in “fully autonomous mode”? Who should we hold responsible for killings committed by autonomous weapon systems? The author has dealt with some of these issues elsewhere (Sparrow 2007, 2009, 2011, 2012, 2013).

It is possible that the answers to some of these larger questions may give us reason to doubt the wisdom and the ethics of developing robots for particular military applications. Indeed, while it may not prove a welcome recognition amongst many of the audience for whom this paper is written, there is clearly a question about the ethics of computer scientists and engineers choosing to work on military applications of robotics at all (Sparrow 2012). There are many more urgent human needs than any that will be met by military robots (Anonymous 2007). In the context of contemporary global inequality, widespread poverty, and looming environmental catastrophe, the amount of human ingenuity, labor, and physical resources devoted to the military is, frankly, morally obscene. It is also a live question whether the objective of achieving national security or a just peace could not be better served by nonmilitary spending (Baard 2003). Researchers face a personal choice about whether they wish to participate in and contribute to such an unsatisfactory state of affairs (Mitcham 1989). Whether they should speak out

against their members working for the military is also an important question for professional associations of engineers, computer scientists, and roboticists (Baard 2003). However, as this argument has been made at length elsewhere, it will not be discussed here (Sparrow 2012).

Thus, this paper focuses on a subset of the larger issues raised by the development of robot weapons: those questions that confront those that *have* chosen to work as the designers of these systems. The ethical issues involved in UMS design may be grouped under two broad headings: Building Safe Systems and Designing for the Law of Armed Conflict. This paper identifies and discusses a number of issues under each of these headings and offers some analysis of the implications of each issue and how it might be addressed.

It must be acknowledged at the outset that not all of the issues treated in this paper are unique to UMS. Many of them clearly also arise in relation to the use of long-range and/or precision munitions more generally. Genuinely new issues may arise if future unmanned systems develop the capacity for truly autonomous decision-making due to advances in “Strong AI” (Sparrow 2007). However, it is difficult to assess the likelihood that this will come about; past predictions about the prospects for genuine machine intelligence of the sort necessary to allow machines to constitute autonomous moral agents have repeatedly been shown by events to have been overly optimistic. In the meantime, however, the many important ethical questions involved in the design of existing UMS, which lack the capacity for morally autonomous action, are not any less important or urgent by virtue of not being entirely unique. Moreover, as is noted at several points below, familiar issues may arise in new guises or may arise with increased urgency in relation to the design of unmanned systems as compared to manned systems. The approach in this paper is therefore to examine and highlight the ethical issues involved in the design of UMS without denying that similar issues may arise elsewhere in engineering and especially in the design of other sorts of long-range weapon systems.

124.2 Building Safe Systems

Before it will be ethical to field UMS, they must be safe to operate and maintain and safe to fight alongside. More precisely, they must be as safe as comparable manned systems capable of achieving similar performance in comparable roles. Their routine operation should not pose more of a threat to the health and safety of the operators than the systems they might replace. As far as is possible, they should not expose their operators to enemy line-of-sight fire. They must be capable of combined operations with manned systems during military exercises and wartime and also of being used for training and in peacetime operations alongside civilian systems (Barry and Zimet 2001; Hockmuth 2007, p. 73). This requirement is especially burdensome on UCAVs and UAVs which must be capable of sharing airspace with both military and civilian aircraft (Hockmuth 2007; Kenyon 2006; Kochan 2005; Lazarski 2002; Wise 2007). UMS, their operators, and other military

forces they intended to operate alongside of must be linked by reliable and robust communication systems in order to minimize the possibility of dangerous accidents (Barry and Zimet 2001; Kenyon 2006, p. 44). They must be capable of reliable “friend or foe” identification in order to avoid causing casualties by “friendly fire” (Kainikara 2002; Marks 2006). Less obviously, as is discussed below, UMS must be safe to use around enemy noncombatants (Arkin 2007, p. 4). All of these requirements are likely to become more demanding as the scope of autonomous action available (and allowed) to UMS increases.

Because the relevant safety comparison between manned and unmanned systems involves comparing “systems capable of achieving similar performance in comparable roles,” this comparison actually speaks strongly *in favor* of the application of UMS in roles which currently involve a high risk to human life (Strawser 2010). Thus, for instance, the use of UMS in bomb and IED disposal and mine clearance is not merely ethical but ethically mandated where possible – even with existing systems. Similarly, once they develop the capacity, UMS may quickly become the option of choice for conducting suppression of enemy air defenses (SEAD) and intelligence, surveillance, and reconnaissance (ISR) missions (Barry and Zimet 2001; Mustin 2002; Office of the Secretary of Defense 2005b, Appendix A; Peterson 2005; Sullivan 2006).

Initially, at least, the problem of making robots safe is for the most part a technological rather than an ethical challenge. It will be possible to greatly improve the safety of military robots by improving the technology in ways that can be specified independently of any commitments on controversial ethical questions. However, as the rest of my discussion makes clear, the use of UMS in armed conflict may generate a range of unanticipated problems and consequences involving harm, or the risk of harm, to various parties, the avoidance and/or mitigation of which will require making difficult choices between competing and sometimes incommensurable values. Designing “safe” systems will therefore require making ethical decisions.

124.2.1 Keeping Humans Out of Harm’s Way?

The literature on military robotics typically emphasizes the value of UMS in keeping human beings out of harm’s way and thus saving (“friendly”) human lives (Boland 2007; Chapman 2002; Featherstone 2006; Fielding 2006; Kainikara 2002; Legien et al. 2006; Office of the Secretary of Defense 2005a; Peterson 2005; Scarborough 2005; Sherman 2005; Veruggio 2006, Sect. 7.7.2). Yet, as UMS become more central to war fighting, it is likely that circumstances will arise in which this relationship will be reversed; human beings will start to be placed in harm’s way as a result of the operations of robots.

There are three scenarios in which this might be expected to occur.

First, as soon as robots can play a useful role in military operations, warfighters will rely on them to complete the tasks to which they have been assigned. For instance, where a robotic mine clearer has swept a minefield, warfighters will expect the area to be free of mines. If robots have been used to secure and patrol

a perimeter, warfighters will expect the area to be clear of the enemy. If robots have been sent to conduct reconnaissance, then warfighters will assume that they have reliable knowledge of enemy assets. Consequently, if robots fail in these tasks, these expectations will be confounded and human lives may be placed at risk.

Second, human lives might be placed at risk in order to defend, service, or recover UMS which are threatened in combat or other operations and also to prevent the enemy gaining valuable intelligence if they should capture a UMS. As UMS develop, they are likely to become increasingly sophisticated and expensive systems and consequently important military assets. Where a high-value UMS is in jeopardy due to a malfunction, or to enemy action, there will be a significant temptation for the commanding officer to commit human forces to protect or recover it. If the system is valuable enough and the apparent risk is low enough, the commanding officer may even be obligated to do so (Wall 2006). It seems likely that, given the extent of the operations in which Predator UCAVs have been engaged and the \$4.5-million-plus price tag of each Predator, such circumstances have already arisen in the current conflicts in Iraq and Afghanistan. Of course, similar scenarios may also arise with military systems without a robotic or remote component. However, these “rescue” or “salvage” missions are more likely with the use of UMS because the nominal capability of these systems to carry out operations without incurring friendly casualties is likely to lead them being deployed in circumstances and in pursuit of objectives where manned systems would not be deployed (Mustin 2002).

Third, the availability of UMS may mean that military conflicts are initiated with the intention that they can be completed without placing warfighters in harm’s way only to discover that victory can only be achieved – or the operation aborted without disaster – with the involvement of human beings in the theater of operations. This may occur either because the weapon system malfunctions or because winning victory turns out to be beyond the capabilities of the UMS due to changed circumstances, including enemy action, or because the operation was ill-conceived in the first place. As a result, human warfighters, especially Air Force pilots and Special Operations forces, may find themselves involved in conflicts because robots have failed to achieve their objectives (Mustin 2002).

The obvious way to minimize these risks is to design and manufacture robots that are more reliable, effective, and also flexible, so that they seldom break down or fail in their missions, and to include robust anti-tamper measures to minimize the need to recover weapons that have been lost in enemy territory. A complex and interesting set of trade-offs then arises. More sophisticated robots are likely to be better able to complete their missions but also more expensive and therefore more likely to drag human beings into combat to repair or recover them when they do fail. Less versatile – and therefore less expensive – robots might be more “disposable” and consequently less likely to put human lives in jeopardy as a result of the need to recover them but also less reliable and more likely to involve humans in combat if they fail in their missions. On the other hand, “better” robots may be more likely to be sent on more ambitious missions and have an increased risk of failure as a result. People may also be more inclined to rely more heavily on what they perceive to be “better” robots. More elaborate and/or ambitious missions may also be more likely to require human intervention to complete or “salvage” them. It will be

important for designers of UMS to think through these trade-offs before setting out to build robots that might avoid the dangers highlighted here.

124.2.2 The Psychological Stresses on Remote Operators

An issue, which is related to the safety of UMS, that is worthy of singling out for special attention is the possibility that the deployment of UMS in armed conflict will expose their operators to new forms of psychological stress. Designers will need to consider the impact of operating UMS on their operators when designing them.

At first sight, the use of UMS might be expected to *reduce* stress and trauma in warfighters by taking them off the battlefield. However, while the use of UMS may reduce physical trauma and some sorts of mental and emotional trauma, it also seems likely to generate significant psychological stress related to the operator being simultaneously “present” in and absent from the battlefield. Through their link with the UMS, operators may witness, and may even participate in, events which are psychologically distressing. In particular, operators in UMS may find themselves witnessing events in which they are powerless to intervene because of the limited capacities of the systems they are operating. Thus, for instance, a UMS operator may witness at close quarters the warfighters they were fighting alongside of being killed, massacres of civilians, or other war crimes, yet be helpless to prevent them.

Of course, human beings involved in war have always been subject to the danger that their experiences will scar them psychologically. It is also true that modern-day warfighters, especially pilots, tankers, gunners, and submariners, already experience many aspects of combat through various electronic or other sensing systems, which mediate that experience (Shurtleff 2002). However, what may be new with the application of UMS is a near-complete disjunct between the simultaneous experience of the consequences of war and the experience of the process of fighting it. In the past there has always been a significant gap between the consequences of war and the experience of fighting at, for most of its participants. In the future, however, the operators of UMS may find themselves simultaneously inhabiting a pleasant office space and the chaotic streets of an urban environment in the midst of violent battle. Operating a UGV or UAV may provide them with a “point of view” in the midst of the battlespace that exposes them to a vivid experience of the chaos of modern war and its consequences (Bender 2005). According to Capt. Steven Rolenc, spokesman for Predator operations at Nellis Air Force Base (Quoted in Sherman 2005, p. 35),

(When) “you put your hands on the controls and your eyes on the screens, you feel as though you’re flying over Iraq or flying over Afghanistan. You get yourself into that reality. It’s not a video game. It’s the real deal.”

Similarly, Air Force Major Shannon Rogers (quoted in Donnelly 2005) claims that

Physically, we may be in Vegas, but mentally, we’re flying over Iraq. It feels real.

Yet when they “log off” the operating system for the UMS, the operators may return to a peaceful office environment and perhaps even, later that day, to their families (Shachtman 2005a). This geographic and psychological distance between the operators and the environment in which they are operating may negatively affect the coping mechanisms that warfighters currently use to process the stresses they experience in combat. In ordinary warfare, the larger context of being physically present in the theater of operations may allow warfighters to prepare for combat through a process of anticipation which makes reference to local circumstances and to deal with them afterwards through conversation and interaction with others who may have shared similar experiences. Fighting a war in a country that one has never set foot in, alongside people one has never met, may be uniquely difficult in terms of the opportunities for warfighters to “process” their experiences.

It may well prove preferable to expose operators to these stresses rather than the stresses they would experience if they were actually present in the battlespace. However, armed forces that operate UMS would be well advised to monitor of the impact of any such stresses on the operators of UMS and to be open to the possibility that the combination of close proximity to, and extreme distance from, violence and its consequences made possible by the development of UMS may have a unique affect on the mental and physical well-being of their operators. According to several reports, these effects are already being felt by the U.S. Air Force operators of Predator UCAVs who pilot these vehicles through the skies of Iraq and Afghanistan by satellite link from the U.S. mainland (Donnelly 2005; Kaplan 2006). The result has been physical and emotional exhaustion amongst the operators and also significant stresses on their families (Kaplan 2006).

It seems likely that the extent and consequences of such stresses will be, at least in part, a function of the design of UMS. In particular, two features of systems may play a role in determining the level of stress to which their operators are exposed.

Firstly, the nature of the interface used to operate the system will play an important role in the stress levels of operators. Systems which provide only abstract and mediated images of the battlespace might be expected to induce little, if any, trauma in those who operate them. On the other hand, the development of more sophisticated virtual reality displays, which will immerse the robot’s operator in the robot’s point of view, might be expected to increase the stress to which operators will be exposed. Designers will therefore need to consider these important human factors when designing the human-machine interface for UMS.

Secondly, the capacities of the systems to intervene in the events that they allow the operator to witness are likely to be relevant. However, the relationship between this capacity and the stress on the operators is unlikely to be straightforward. There are, in fact, two different possible types of stress that operating a UMS may produce: stresses relating to being a *witness* of traumatic events and stresses relating to being a *participant* in traumatic events.

The most intensive source of psychological stress on the operators of UMS is likely to be regret arising from what the operator did – or failed to do – during combat operations. Systems that provide *no* capacity to intervene may therefore provoke *less* stress than those that do offer some capacity to shape the outcome of

events in the battlespace. However, even these “passive” systems may traumatize their operators if they are forced to witness atrocities of various sorts; machines that provide the experience of “being there” are likely to be especially problematic in this regard.

Once the system *does* offer its operator some way to affect the battlespace, then the operator is likely to feel responsibility for the exercise of this power and to be prone to regret if they come to believe that they should have done otherwise than they did. It is therefore tempting to conclude that the “best” systems would be those that provided their operators with the largest possible scope of action so that they could prevent or avoid outcomes that might otherwise traumatize them. That is to say, that more powerful systems, which make possible a wider range of actions, would be less likely to expose their operators to “participant” trauma. However, if the operator has more power to affect things, then he/she may also have more to regret. In particular, operators may come to regret what they did do, as much as what they didn’t. Increasing the capacities of systems therefore provides no easy solution to the problem of the stresses to which operators are exposed. However, it is apparent that, in terms of the risk of psychological trauma to operators, the *worst* systems will be those that leave the operator with the sense that they could have altered the outcome of the scenario when in fact they could not have. This conclusion is more significant than first appears because, despite the “hype” surrounding them, many unmanned systems are severely limited in the range of actions that they make possible. A particularly important limitation of such systems in this context is their inability to provide emergency medical care to wounded warfighters and/or civilians where required. It will therefore be crucial to ensure that operators have realistic understandings of the effective capacities of the systems that they are operating, in order to minimize the chance that they will regret circumstances that they were in fact unable to alter.

Because of the difficulties involved in locating unclassified research on the operations of UMS in military roles, these remarks are necessarily, to some degree, speculative. However, at the very least, it is clear that good design in relation to these issues must be guided by the results of empirical research into the appropriate questions in the fields of human-robot interaction and combat psychology.

124.3 Designing for the Law of Armed Conflict

The ethical use of UMS in armed conflict, like that of other weapons, will need to be governed by the law of armed conflict (LOAC) (Gulam and Lee 2006; Lazarski 2002). The details of the application of the LOAC to the operation and design of UMS is a matter for military lawyers (Dalton 2006; Gulam and Lee 2006; Royal Australian Air Force 2004, pp. 74–75). However, several core requirements of the law of armed conflict do place significant constraints on the ethical use and design of these systems and are therefore worth considering here.

124.3.1 Can Robots Meet the Criteria of Discrimination and Proportionality?

It is clear that, at a bare minimum, any ethical use of UMS will need to comply with the principles of discrimination (Bender 2005) and proportionality, which derive from the just war doctrine of *jus in bello* (Schmitt 2005). That is to say, UMS must be capable of discriminating between legitimate and illegitimate targets and of applying force proportionate to the pursuit of legitimate military ends (Arkin 2007, p. 2). These are perhaps the principal design challenges involved in the future development of UMS, especially UMS that are intended to have the capacity to operate autonomously (Marchant et al. 2011).

One possible “work-around” to these problems is to deploy UMS in such a way as to exclude the possibility that they will attack illegitimate targets. For instance, armed sentry robots could patrol inside a perimeter fence with suitable warnings on it to prevent any possibility of noncombatant presence (Arkin 2007, pp. 92–93). However, it may be difficult to reliably prevent robots from acquiring illegitimate targets in this way, especially during wartime. Moreover, limiting the use of UMS to circumstances where they will not come into contact with noncombatants will severely limit their military value. John Canning (2005, 2006) has championed another version of this approach, in which autonomous weapon systems would be tasked with targeting only enemy weapon systems rather than enemy warfighters, thus minimizing the chance of killing noncombatants. Similar problems seem likely to beset this approach. In many circumstances, it will be difficult to distinguish, for instance, an armed tribesman carrying an AK-47 because this is the local cultural practice, from a hostile insurgent. There is also the danger that the number of noncombatants killed when an attack is launched against an enemy weapon system (for instance, a mortar fired from a hospital car park) will be such as to render the force used disproportionate. Unwillingness to risk these outcomes, on the other hand, will significantly limit the context in which automated weapon systems may ethically be used.

Another possible solution – to the problem of *discrimination* at least – is to try to equip the systems themselves with the capacity to make the requisite judgements as to when a target is legitimate (Arkin 2009). Some weapon systems, including antitank, anti-aircraft, anti-ship, and counterfire systems, may be capable of distinguishing between military and civilian targets. If the target recognition algorithms on weapons of this sort are sufficiently reliable, it might appear that there would be fewer ethical barriers to deploying them (Arkin 2009).

However, even with these weapons, there is the possibility that a potential military target may have indicated its desire to surrender or may no longer pose sufficient military threat to be a legitimate target of attack (Fielding 2006). The fact that the principle of discrimination is extremely context dependent suggests that autonomous weapon systems would have to have a very high level of autonomy indeed to be able to make the judgements necessary in order to comply with its requirements (Guarini and Bello 2012). Similarly, decisions about what constitutes

a level of force *proportionate to the threat* posed by enemy forces are extremely complex and context dependent, and it seems unlikely that machines will be able to make these decisions reliably for the foreseeable future. Barring some remarkable breakthrough in artificial intelligence research, it will therefore be necessary to include a human “in the loop” before deploying lethal ordnance from a UMS in order to ensure that the use of UMS does not violate the LOAC in this regard (Gulam and Lee 2006; Fitzsimonds and Mahnken 2007; Kenyon 2006, p. 43).

If the application of UMS is going to rely on a “human in the loop” in order to ensure that it complies with the requirements of discrimination and proportionality, the operation of UMS must provide their operators with sufficient information to be able to make the necessary judgements reliably. Given the limited situational awareness available to operators of UMS (Kainikara 2002; Mustin 2002), this may require that operators have access to independent sources of information about the nature of the targets they are attacking. This, in turn, has implications for the nature and capacities of the communication systems that are required in order to be able to use UMS ethically (Thorton 2005).

124.3.2 Locating Responsibility

As the author has argued more extensively elsewhere (Sparrow 2007), the application of the law of armed conflict to the use of UMS requires that a clear chain of responsibility can be established between the consequences of any such use and the person who is responsible for them (Arkin 2007, pp. 76–83; Asaro 2007; Foster 2006; Roff 2013). If violations of ethical standards occur, it must be possible to identify the sources of violation in order that such violations can be addressed and, if necessary, the relevant party criticized, disciplined, or prosecuted. This will be a significant challenge given that multiple parties may be involved in the operations of UMS (Featherstone 2007; Sullivan 2006).

The need to be able to determine responsibility for the activities of UMS has implications for engineers, roboticists, and computer scientists working on UMS in at least three dimensions.

Firstly, the designers are an obvious and important possible endpoint for the allocation of responsibility for the consequences of the operation of these systems. Thus, for instance, if a design error in a UMS results in the killing of civilians or friendly forces, then the designers will be partially responsible for these deaths. That the designers of UMS might be – and might be held to be – responsible for the deaths of innocents should serve as a reminder of the gravity of their role and an incentive for them to perform this role diligently.

Secondly, a concern for the attribution of responsibility will have implications for the appropriate organization of the process of designing and operating unmanned systems, which should be such as to clearly allocate responsibility for each distinct function of the mechanism and also for the function of the system as a whole. If a problem arises with the operations of an unmanned weapon system, it must be possible to identify those responsible (Marino and Tamburrini 2006). Of course,

this is but one instance of the application of more general principles of good design of complex systems and of good project management. However, this fact does not render their application any less important in this instance. Moreover, the requirement that it must be possible to identify those responsible for system failures will be especially demanding if an unmanned weapon system has the capacity to operate in a “fully autonomous” mode, in which case it will be necessary to *assign* responsibility for the outcomes of the actions of the machine in this mode to appropriate parties (Marino and Tamburrini 2006), which may include the commanding officer and/or those who have designed/programmed the UMS (see Sparrow (2007) and Matthias (2004) for further discussion).

Thirdly, the capacity to sustain a chain of responsibility during the operations of UMS is itself an important criteria of good design of these systems. Thus, for instance, “ethical” systems will have robust communications linking them with their operators and will have mechanisms in place to record telemetry data in order that problems with – and responsibility for – the operations of the systems can be identified when necessary.

124.3.3 “Designing Out” War Crimes

The combination of a number of features of UMS may function to lower the psychological, social, and institutional barriers to the commission of war crimes. In particular, the geographic and psychological distance between the operator and the UMS may make it easy for the operator to perform actions that they would not perform if they were physically present in the battlespace (Cummings 2004, 2006; Graham 2006; Ulin 2005). As a result, they may be more likely to attack illegitimate targets or to use a disproportionate amount of force in attacking legitimate targets.

Of course, this possibility exists with any long-range weapon. Modern warfighters already possess a godlike power to call down destruction from the skies upon their enemies. Telescopic sights, night-vision goggles, and other targeting systems already distance those exercising lethal force from the human beings their actions affect. These existing technologies may therefore already serve to lower some of the psychological barriers to illegitimate killing (Dunlap 1999; Shurtleff 2002).

However, the use of UMS is likely to exacerbate this distancing effect while at the same time increasing “contact” with the enemy (Coeckelbergh 2013). In particular, UAVs may make surveillance ubiquitous on the battlefield and also extend it throughout the entire territories of the warring states. The operators of the Predator UAV can watch people going about their daily activities in real time from half a world away. The Predator UAV flies at such a height and produces so little engine noise that those being tracked via this system may have no idea that they are under observation (Fulghum 2003; Kaplan 2006). Watching targeting video taken from the Predator (available via YouTube!) one is struck by the bizarre sense of intimacy this footage generates (Blackmore 2005, p. 202). One description of the Predator UCAV in action recounts how it is possible for the operators to observe individuals in Afghanistan walking outside to defecate and confirm that they have

done so by the infrared traces of the feces they have left behind (Kaplan 2006). Yet the explosions that inevitably follow in footage of drone strikes seem entirely unreal, flickerings on a cathode ray screen made trivial by their similarity to images we have seen a thousand times before in film and on television. By simultaneously increasing the amount of contact with the enemy while distancing warfighters from them, UMS may contribute to weapon operators coming to see enemy combatants and noncombatants as distant annoyances only, to be destroyed on the merest of whims. This in turn may lead to (more) violations of the requirements of *jus in bello*.

It has been suggested to the author that, on the contrary, by reducing the risks to which their operators are exposed, UMS may also lower the levels of anger, fear, and hatred amongst warfighters. The strong emotions that warfighters in combat feel towards their enemy are in part a product of the fact that enemy often poses an immediate physical threat. As the lives of operators of UMS are not at risk, they may be less inclined to experience these passions. Insofar as anger, hatred, or fear are implicated in the commission of war crimes, reducing the level of these emotions amongst warfighters might also be expected to reduce the number of war crimes.

However, whether or not the operators of UMS are likely to develop more or less compassion and respect for their enemy or be more or less motivated by anger, fear, and hatred is, at the very least, an open question. Anecdotal evidence suggests that the operators of UMS become emotionally engaged in and experience strong emotions in response to events in the battlespace regardless of their geographical distance from it. Bender (2005) quotes one operator reporting that

... "the feeling of anger you get is pretty powerful" when American troops are seen or heard taking fire from insurgents. Others speak of the "adrenaline rush" in the room when the Predator destroys an enemy target. (p. A6)

Another report (Shachtman 2005a) quotes a Major Shannon Rogers describing his experience with a UAV:

"We left their truck one big smoking hole", he remembers. "My heart was pumping as we were doing our business. It felt just as real to me, however many thousands of miles away, as if I were sitting right there in that cockpit."

These reports suggest that the operators of UMS *do* experience those emotions implicated in the commission of war crimes. Yet while these systems are capable of sustaining and generating these powerful negative emotions, it is less clear that they are capable of generating the emotions and moral attitudes that might serve to prevent war crimes. Emotions such as compassion, joy, love, or empathy or moral attitudes such as respect are unlikely to develop or be sustained in a context where warfighters are thousands of miles away from their purported objects. Indeed, a likely consequence of increased use of UMS is a decrease in the number of military personnel involved in conflicts who have ever met or spoken with the people who inhabit the territory in which war is being fought. In the absence of any human relations with those who their actions will affect, warfighters may be less inclined to resist impulses arising out of fear, hatred, or anger where they arise.

As Mary Cummings (2004, 2006) has argued, the alienation of the operators from the persons that their actions affect means that building ethics “into” the user interface represents a key challenge in the design of UMS. The interface is the primary means by which the operators gain access to information on the basis of which they must make life-and-death decisions; it is also the mechanism whereby the operators act on the basis of these decisions. The design of the interface can therefore be expected to exercise a significant influence on the actions of operators. This in turn suggests that the designers of the interfaces must take ethical considerations into account when designing them. The interface for a UMS should facilitate killing where it is justified and frustrate it where it is not. Obviously, it will not be possible for designers to make this discrimination at the level of individual actions; nor will it be possible to prohibit deliberate unethical use of UMS by an ill-motivated operator. However, it should be possible to take into account the morally relevant features of the circumstances in which a UMS is designed to be used and also those of its typical use and to design systems that promote ethical usage in these circumstances. In particular, the designers of UMS should work to discourage and counteract the alienation between operator and those whose lives they affect, noted above.

Perhaps the most important feature of systems in relation to this imperative is their sensors and the information that they provide to operators. “Good” sensors and good interfaces will present the operator with as much of the information that is morally relevant to the decisions that they must make as is possible. This might include information relevant to the identity, intentions, and history, of potential targets, as well as their current location and activities. This in turn is likely to require providing operators with access to available “human intelligence” and to any other relevant information available via other networks or systems. This information should help operators distinguish between legitimate and illegitimate targets according to LOAC and also distinguish situations where the killing of (legally, rather than morally) legitimate targets is justified from situations where it may not be. Moreover, the sensors and information systems should facilitate the operators being able not only to reliably identify targets but also to comprehend what happens to them when lethal force is deployed. That is, as much as is possible, the system’s sensors should communicate the moral reality of the consequences of the actions of the operator. It is essential that operators have a vivid awareness of what is at stake when they make decisions, so that they can learn to make them responsibly and well.

It must be acknowledged that these are demanding goals and that there are likely to be significant limits on designers capacity to achieve them, especially given the concerns expressed above about whether remote systems are capable of transmitting and representing the full range of moral and emotional information relevant to combat and also the budgetary and other design constraints the designers of UMS must operate under. Nevertheless, it is important to clarify and state them in this context in order that they may at least provide some direction in relation to ethical design. It is also worth noting that at least some of the imperatives arising out

of a concern for ethical design align with those arising out of a concern with the operational demands on the systems.

Where possible, UMS should possess a range of capacities beyond the firing of deadly weapons. Lethal force should not be the first and only recourse available to operators. Thus, for instance, UGVs for operations in urban environments should have the capacity to communicate with other persons in the battlespace, including noncombatants. Consideration should also be given to equipping such systems with “non” – or, more accurately, “sub” – lethal weapons. Providing operators with a larger range of options will allow them to respond more appropriately – and hopefully, more ethically – to each particular situation.

A possible *policy* response to the geographical and psychological distance between operators and their (potential) targets, with design implications, would be to return the operators of UMS to the theater of operations in which the systems they operate function, so that they have some experience of the local culture and contact with the people whose lives they are affecting. This would have the obvious disadvantage of placing the operators at higher risk by virtue of being closer to the front line and would also involve expenses associated with the transport and supply of these personnel and associated systems. However, it would go some way to addressing the concerns about “remote control” killing surveyed above. Whether this trade-off is sufficient to justify putting the operators closer to the front line will depend on the details of the particular UMS and the roles for which it is intended, as well as the intensity of the particular conflict. Obviously, the range over which systems are intended to be used will have significant design implications. Armed Forces will therefore need to consider these issues early in the design of such systems.

Another important aspect of the project of ethical design will be working to avoid the creation of *other* types of what (Cummings 2004, 2006) describes as “moral buffers” between the operator and their actions. Many UMS have a good deal of automation built into them, into their sensors as well as into the operations of the system itself. Cummings suggests that the operators may come to over-rely on the automation in these systems, to the detriment of the decisions they make. Moreover, Cummings argues, the presence of this automation may form a (further) moral buffer between the users of the system and their actions, allowing them to tell themselves that “the machine” made the decision. This, in turn, may encourage unethical choices by operators who are thereby able to distance themselves morally and emotionally from the consequences of their actions. Good design of UMS will therefore not only (as discussed above) ensure that we can *identify* those responsible for the consequences of the operation of the system, it will also ensure that those who *are* responsible *feel* responsible – and know precisely what they are responsible for.

A final, important, consideration relating to the ethical design of UMS is that because the operators of the UMS are themselves not in any danger while they are in combat, it seems as though they should err on the side of caution when it comes to making decisions about killing. This means that there is some room in designing these systems for checks and balances, such as the gathering of more

information outlined above, which would not be practical for warfighters actually present in the battlespace. Of course, importantly, the lives of *other* friendly forces in the battlespace may well be at stake, which means that many missions will still be time critical, such that it will still be necessary for the UMS to be capable of responding quickly as required. Nevertheless, the fact that the operators of UMS are safe from harm does alter what it is reasonable to expect from them by way of a concern for the requirements of *jus in bello* and therefore the requirements of ethical design of these systems.

The challenge of ethical design is a challenge that, arguably, it is yet to be adequately met. A number of articles on UMS report that the control units for these systems have deliberately been designed around the controls for the PlayStation game console or around a “Gameboy” controller in order to take advantage of potential operators existing familiarity with these controls (Graham 2006; Hambling 2007; Kenyon 2007; Thorton 2005). Unsurprisingly, this has led to reports that there is a tendency for the operators to mistake their activities for playing computer games.

According to Bender (2005),

For the Predator crews in Nevada, however, the main challenge is simply to remember they are not playing a video game when they step out of their air-conditioned office for a Wendy’s hamburger.

“We have to impress upon them that they are not just shooting electrons,” said Major Sam Morgan, a trainer of Predator pilots. “They’re killing people.”

Shachtman (2005b) also cites an analyst saying, of the operation of these systems,

It’s like a video game. It can get a little bloodthirsty. But it’s fucking cool.

Presuming that we do not want warfighters making life-and-death decisions in the subconscious belief that they are playing a video game, these remarks suggest that much work remains to be done in promoting ethical behavior through the design of UMS.

124.4 Conclusion

The development of UMS for military applications poses ethical as well as technical design challenges for roboticists, engineers, computer scientists, and others involved in the design of these systems.

This paper has argued that one set of ethical issues arises out of a concern for the safety of the operators of the system and of those who will work alongside them. While the development of UMS may keep some warfighters safe from harm, there are also circumstances in which it may place others at risk. Responding to this phenomenon will require negotiating a complex set of trade-offs regarding the capacities – and expense – of these systems. The current discussion has also highlighted the possibility that operation of UMS may expose the operators to unique psychological stresses.

There is also another, arguably more difficult, set of issues that arise out of the need for the operations of UMS to meet the requirements of the law of armed conflict. The profound difficulties involved in enabling fully autonomous weapon systems to distinguish between legitimate and illegitimate targets and assess the proportionate use of force suggest that it will be necessary to retain a “human in the loop” for the foreseeable future. A critical requirement for the ethical design and operation of UMS is that those responsible for each aspect of their design and operations can be clearly identified; this may require attributing responsibility if these systems are used in a “fully autonomous” mode. Finally, ethical design of UMS will require paying attention to the ways in which these systems may facilitate unethical behavior by separating the operators from the consequences of their actions and working to overcome this and other “moral buffers” that may arise in the operation of robotic weapons.

Some of these ethical issues can be “designed out” or at least “designed around.” That is to say, with sufficient ingenuity and more sophisticated technology, engineers can avoid them and/or minimize their impacts. For instance, better neural networks, better sensors, and better target recognition algorithms might mitigate some of the difficulties involved in the problem of discrimination and expand the number of roles in which it was appropriate to employ autonomous weapon systems. Sensors which make possible a more vivid appreciation of the battlespace might also reduce the psychological distance between operators and their targets and thus reduce the extent to which the distance between them acts as a “moral buffer.”

However, it is important to note that there are real trade-offs involved in attempting to address some of these issues, such that efforts to address one will intensify another. Thus, for instance, technologies which make possible a larger scope for autonomous operations of robotic weapon systems exacerbate the problem of locating the responsibility for the consequences of the operations of these systems. Systems with sensors that make the battle more “real” for the operators will also increase the risk that they will suffer psychological stress as a result. Controlling UMS over long distances will decrease the risk to the operators but increase the “moral buffer” between them and their actions. Designers of UMS will therefore need to consider the relative priorities of these issues, the relationships between them, and the trade-offs involved in attempts to address them, in the pursuit of “ethical” design.

Unmanned systems are already playing a key role in contemporary conflicts. Given current enthusiasm for UMS and their obvious utility it seems likely that robotic systems will be used much more widely in military roles in the future (Featherstone 2007; Graham 2006; Hanley 2007; Hockmuth 2007; Office of the Secretary of Defense 2006, 2012; Peterson 2005; Scarborough 2005; Singer 2009). It also seems likely that the scope of autonomous action allowed to UMS will increase as the technology improves. Because the communication infrastructure required to keep a “human in the loop” is a weak point in UMS, systems which dispense with a human operator will be more survivable. As the tempo of battle increases as a result of technological developments, including the development of UMS, systems which rely on human input may be at a substantial disadvantage in

combat against fully autonomous systems. There is therefore a substantial incentive for designers of UMS to provide systems with a capacity for autonomous operations (Adams 2001; Blackmore 2005; Excell 2007; Featherstone 2007; Lerner 2006; Szafranski 2005). These trends make the challenges of ethical design of unmanned systems highlighted here all the more urgent.

Acknowledgments The research for this paper was supported by the Australian Research Council, through the award of an ARC Discovery Grant (DP0770545) to Dr Jessica Wolfendale, Professor Tony Coady, and Dr Robert Sparrow. The author would like to thank Neil McKinnon for assistance with locating sources for this paper and with preparing it for publication. The author would also like to thank Jessica Wolfendale, Linda Barclay, Jim Sparrow, John Canning, and Ron Arkin for comments and discussion which have improved this paper.

References

- T.K. Adams, Future warfare and the decline of human decision-making. *Parameters: U.S. Army War College Quarterly* (Winter, 2001–2), 57–71 (2001)
- Anonymous, Defense sector accused of muscling in on robots. *Prof. Eng.* **20**(3), 8 (2007)
- R.C. Arkin, Governing lethal behaviour: embedding ethics in a hybrid deliberative/reactive robot architecture. Technical report GIT-GVU-07-11 for US Army. Mobile Robot Laboratory, College of Computing, Georgia Institute of Technology (2007). Available at <http://www.cc.gatech.edu/ai/robot-lab/online-publications/formalizationv35.pdf> at 25.10.07
- R.C. Arkin, *Governing Lethal Behavior in Autonomous Robots* (Chapman and Hall, Boca Raton, 2009)
- P.M. Asaro, Robots and responsibility from a legal prospective, in *IEEE International Conference on Robotics and Automation*, Roma, 2007
- E. Baard, Make robots not war: some scientists refuse to get paid for killer ideas. *The Village Voice*, 10–16 Sept 2003, pp. 39–44
- C.L. Barry, E. Zimet, UAVs – technological, policy, and operational challenges. *Defense Horizons* (3) (2001)
- C. Beal, Briefing: autonomous weapons systems – brave new world. *Jane's Def. Wkly.* **33**(6), 22–26 (2000)
- B. Bender, Attacking Iraq, from a Nev. Computer. *Boston Globe*, 3 Apr 2005, A6
- T. Blackmore, Dead slow: unmanned aerial vehicles loitering in battlespace. *Bull. Sci. Technol. Soc.* **25**(3), 195–214 (2005)
- R. Boland, Developing reasoning robots for today and tomorrow. *Signal* **61**(6), 43–46 (2007)
- A. Butler, Global Hawk UAV supports border ops in Iraq. *Aviat. Week Space Technol.* **166**(11), 56 (2007)
- J.S. Canning, *A Definitive Work on Factors Impacting the Arming of Unmanned Vehicles* (Dahlgren Division Naval Surface Warfare Centre, Washington, D.C., 2005)
- J.S. Canning, A concept of operations for armed autonomous systems, in *3rd Annual Disruptive Technology Conference*, Washington, D.C., 6–7 Sept 2006
- J. Card, Killer machines. *Foreign Policy* (May/June): 92 (2007)
- Col. R.E. Chapman, Unmanned combat aerial vehicles: dawn of a new age? *Aerosp. Power J* **16**(2), 60–73 (2002)
- M. Coeckelbergh, Drones, information technology, and distance: mapping the moral epistemology of remote fighting. *Ethics Inf. Technol.* **15**(2), 87–98 (2013)
- M.L. Cummings, Creating moral buffers in weapon control interface design. *IEEE Technol. Soc. Mag.* **23**(3), 28–33, 41 (2004)
- M.L. Cummings, Automation and accountability in decision support system interface design. *J. Technol. Stud.* **32**(1), 23–31 (2006)

- J.G. Dalton, Future navies – present issues. *Nav. War Coll. Rev.* **59**, 17–39 (2006)
- S.B. Donnelly, Long-distance warriors. *Time Magazine*, 4 Dec 2005
- C.J. Dunlap Jr., Technology: recomplicating moral life for the nation's defenders. *Parameters: US Army War College Quarterly (Autumn)* **29**, 24–53 (1999)
- J. Excell, Unmanned aircraft: out of the shadows. *The Engineer*, 18 Jan 2007
- S. Featherstone, The coming robot army. *Harper's Magazine*, 43–52 (2007)
- M. Fielding, Robotics in future land warfare. *Aust. Army J.* **3**(2), 1–10 (2006)
- J.R. Fitzsimonds, T.G. Mahnken. Military officer attitudes toward UAV adoption: exploring institutional impediments to innovation. *Joint Force Q.* **46**, 96–103 (2007)
- Lt. Col. J. Foster, Ricochets and replies: first rule of modern warfare. *Air and Space Power J.* (Spring 2006)
- D.A. Fulghum, Predator's progress. *Aviat. Week Space Technol.* **158**(9), 48–50 (2003)
- S. Graham, America's robot army. *New Statesman* **135**(4796), 12–15 (2006)
- M. Guarini, P. Bello. Robotic warfare: some challenges in moving from noncivilian to civilian theaters, in *Robot Ethics: The Ethical and Social Implications of Robotics*, ed. by P. Lin, K. Abney, G.A. Bekey (MIT, Cambridge/London, 2012), pp. 129–144
- H. Gulam, S.W. Lee, Uninhabited combat aerial vehicles and the law of armed conflict. *Aust. Army J.* **3**(2), 1–14 (2006)
- D. Hambling, Military builds robotic insects. *Wired Magazine*, 23 Jan 2007. Available at <http://www.wired.com/science/discoveries/news/2007/01/72543>
- C.J. Hanley, Robot-aircraft attack squadron bound for Iraq. *Aviation.com*, 16 July 2007
- C.M. Hockmuth, UAVs – the next generation. *Air Force Magazine* Feb 2007, pp. 70–74
- S. Kainikara, UCAVs probable lynchpins of future air warfare. *Asia-Pac. Def. Rep.* **28**(6), 42–45 (2002)
- R.D. Kaplan, Hunting the Taliban in Las Vegas. *Atlantic Monthly*, 4 Aug 2006
- H.S. Kenyon, Israel deploys robot guardians. *Signal* **60**(7), 41–44 (2006)
- H.S. Kenyon, Airborne testbed opens new possibilities. *Signal* **61**(9), 47–49 (2007)
- A. Kochan, Automation in the sky. *Ind. Robot* **32**(6), 468–471 (2005)
- A.J. Lazarski, Legal implications of the uninhabited combat aerial vehicle. *Air Space Power J.* **16**(2), 74–83 (2002)
- W. Legien, C.-J. Andersson, G. Hansen, UUV and USV: which 'unmanned' for what task? *Nav. Forces* **27**(3), 44–51 (2006)
- P. Lerner, Robots go to war. *Pop. Sci.* **268**(1), 42–96 (2006)
- G. Marchant, R. Arkin, E.T. Barrett, J. Borenstein, L.M. Gaudet, O. Kittrie, P. Lin, G.R. Lucas, R. O'Meara, International governance of autonomous military robots. *Columbia Univ. Rev. Sci. Technol. Law* **12**, 272–315 (2011)
- D. Marino, G. Tamburrini, Learning robots and human responsibility. *Int. Rev. Inf. Ethics* **6**, 47–51 (2006)
- P. Marks, Armchair warfare. *New Sci.* **192**(2575), 24 (2006)
- J.P. Masey, Towards an unmanned navy: the growing world of UAV, UCAV, USV and UUV. *Nav. Forces* **27**(6), 23–31 (2006)
- A. Matthias, The responsibility gap: ascribing responsibility for the actions of learning automata. *Ethics Inf. Technol.* **6**, 175–183 (2004)
- J.W. McMains, The marine corps robotics revolution. *Mar. Corps Gaz.* **88**(1), 34–37 (2004)
- C. Mitcham, The spectrum of ethical issues associated with the military support of science and technology, in *Ethical Issues Associated with Scientific and Technological Research for the Military*, ed. by C. Mitcham, P. Siekevitz. Volume 577 of *Annals of the New York Academy of Sciences* (New York Academy of Sciences, New York, 1989)
- J. Mustin, Future employment of unmanned aerial vehicles. *Air and Space Power J.* **16**(2), 86–97 (2002)
- Office of the Secretary of Defense, *Joint Robotics Program Master Plan FY2005: Out Front in Harm's Way* (Office of the Undersecretary of Defense (AT&L) Defense Systems/Land Warfare and Munitions, Washington, D.C., 2005a)

- Office of the Secretary of Defense, *Unmanned Aircraft Systems Roadmap: 2005–2030* (Department of Defense, U.S. Government, Washington, D.C., 2005b)
- Office of the Under Secretary of Defense, *Development and Utilisation of Robotics and Unmanned Ground Vehicles: Report to Congress* (Office of the Under Secretary of Defense, Acquisition, Technology and Logistics, Portfolio Systems Acquisition, Land Warfare and Munitions, Joint Ground Robotics Enterprise, Washington, D.C., 2006)
- Office of the Secretary of Defense, *Unmanned Systems Integrated Roadmap FY2011–2036* (Department of Defense, U.S. Government, Washington, D.C., 2012)
- G.I. Peterson, Unmanned vehicles: changing the way to look at the battlespace. *Nav. Forces* **26**(4), 29–38 (2005)
- H.M. Roff, Responsibility, liability, and lethal autonomous robots, in *Routledge Handbook of Ethics and War: Just War Theory in the 21st Century*, ed. by F. Allhoff, N.G. Evans, A. Henschke (Routledge, Abingdon/New York, 2013), pp. 352–364
- Royal Australian Air Force, *AAP 1003 – Operations Law for RAAF Commanders*, 2nd edn. (Air Power Development Centre, Canberra, 2004)
- M.N. Schmitt, Precision attack and international humanitarian law. *Int. Rev. Red Cross* **87**(859), 445–466 (2005)
- R. Scarborough, Special report: unmanned warfare. *Washington Times*, 8 May 2005
- N. Shachtman, Attack of the drones. *Wired Magazine* **13**(6) (2005a). Available at http://www.wired.com/wired/archive/13.06/drones_pr.html, at 25.08.05
- N. Shachtman, Drone school, a ground's-eye view. *Wired Magazine*, 27 May 2005b. Available at <http://www.wired.com/science/discoveries/news/2005/05/67655> at 17.9.07
- J. Sherman, The drone wars. *Bull. At. Sci.* **61**(5), 28–37 (2005)
- D.K. Shurtleff, The effects of technology on our humanity. *Parameters: US Army War College Quarterly* (Summer), 100–112 (2002)
- P.W. Singer, *Wired for War: The Robotics Revolution and 21st Century Conflict* (The Penguin Press, New York, 2009)
- R. Sparrow, Killer robots. *J. Appl. Philos.* **24**(1), 62–77 (2007)
- R. Sparrow, Predators or plowshares? Arms control of robotic weapons. *IEEE Technol. Soc.* **28**(1), 25–29 (Spring 2009)
- R. Sparrow, Robotic weapons and the future of war, in *New Wars and New Soldiers: Military Ethics in the Contemporary World*, ed. by J. Wolfendale, P. Tripodi (Ashgate, Surrey/Burlington, 2011), pp. 117–133
- R. Sparrow, Just say no' to drones. *IEEE Technol. Soc.* **31**(1), 56–63 (2012)
- R. Sparrow, War without virtue? in *Killing by Remote Control*, ed. by B.J. Strawser (Oxford University Press, Oxford/New York, 2013), pp. 84–105
- B.J. Strawser, Moral predators: the duty to employ uninhabited aerial vehicles. *J. Mil. Ethics* **9**(4), 342–368 (2010)
- J.M. Sullivan, Evolution or revolution? The rise of UAVs. *IEEE Technol. Soc. Mag.* **25**(3), 43–49 (Fall 2006)
- Col. R. Szafranski, The first rule of modern warfare: never bring a knife to a gunfight. *Air and Space Power J.* **19**(4), 19–26 (2005)
- R.L. Thorton Jr. Captain, The case for robots in the SBCTs (now). *Infantry* **94**(1), 33–41 (2005)
- D.L. Ulin, When robots do the killing. *Los Angeles Times*, 30 Jan 2005
- G. Veruggio, *EURON Roboethics Roadmap* (European Robotics Research Network, Genoa, 2006)
- G. Veruggio, F. Operto, Roboethics: social and ethical implications of robotics. *Springer Handbook of Robotics*, eds. B. Siciliano, O. Khatib (Springer, Berlin, 2006), pp. 1499–1524
- R. Wall, Reality check. *Aviat. Week Space Technol.* **165**(5), 33 (2006)
- J. Wise, No pilot, no problem. *Pop. Mech.* **184**(4), 64 (2007)

George R. Lucas Jr.

Abstract

The chapter contains a transcript of testimony presented by the author and Handbook Field Editor to a select ad hoc committee convened by the National Academy of Sciences (USA) in response to a request from the U.S. Defense Advanced Research Projects Agency (DARPA) to evaluate the “Ethical and Societal Implications of Advances in Militarily Significant Technologies that are Rapidly Changing and Increasingly Globally Accessible.” The testimony was delivered at the inaugural meeting of the committee on 31 August 2011 at the NAS Beckman Center, on the campus of the University of California – Irvine.

For the first time this past year (2011), the Air Force recruitment and training pipeline turned out more RPV/UAV operators than it did conventional fighter pilots. This comes as a dual shock to the fighter pilots. First, it was painful to be asked to leave the cockpit and set aside years of education, training, and experience and to sit at a control station thousands of miles removed from the aircraft itself. Pilots resisted this change, until they saw that they were simply leaving themselves out of the fight: better to be flying Predators themselves, they concluded, than to be replaced by teenagers with joysticks! But then came a further blow: the highly skilled fighter pilots were less adapt at learning the ropes and operating Predators than were teenagers with barely a high school education but a lot of video and computer game experience. This is humiliating enough. But one day, one can predict, the Air Force Chief of Staff will be chosen from the ranks of RPV operators. That will provide

G.R. Lucas, Jr.
United States Naval Academy, Annapolis, MD, USA
Naval Postgraduate School, Monterey, CA, USA
e-mail: grlucas@nps.edu

the culmination of a cultural sea change among members of the military aviation community. How will that community respond? How will it define and uphold its core values in this new mechanized, automated, “virtual” milieu?

Meanwhile, the U.S. Army has spent millions of dollars to stand up a new “Army Center for the Professional Military Ethic.” No service has worked harder, both to define itself as a community of professionals and to investigate just what the core competencies, skill sets, and core moral values are that define that professional community. But what sense does it make to speak of the Army’s professionalism and professional ethic, if an autonomous and lethally armed ground combat robot can do the same job as well or better? Those are the kinds of questions and challenges that new military technologies are presenting to military services now, and indicative of the transformative effect they will one day doubtless have on civilian communities as well.

With the advent of these exotic new military technologies and the challenges they present, one hears a familiar refrain:

- “Our traditional concepts of warfare and its justification are outmoded.”
- “Existing laws of war and moral constraints regarding conventional combat are useless.”
- “Our conceptions of both must either be cast aside or radically reformulated.”

It can be described as a familiar refrain, because:

- One heard this same complaint earlier, with the wars of genocidal violence and the humanitarian military responses launched in the 1990s.
- One heard this in the aftermath of the terrorist attacks of 9/11 and the subsequent rise of simmering “small wars” of counterinsurgency and other varieties of “irregular” warfare over the past decade (2001–2011).
- And one hears this concern expressed now with the advent of exotic new war-fighting technologies, particular the prospect of lethally armed and autonomous robotics.

There is some rhetorical value to this stratagem. It serves to startle, alarm, and even terrify those who otherwise don’t seem to take the new and very real challenges seriously enough. There is a real danger, however, of threat inflation, exaggeration, and hysteria, as though we are somehow not well equipped to deal with the new challenges, once we have finally awakened to them.

These ethical challenges are very real and complex, but after 2–3 years of sustained reflection, one can predict that the patterns seen here will duplicate the discoveries of the preceding two decades: to wit, conceptions of morally justifiable resort to war, and of the moral and legal constraints on its conduct, are *not* outmoded.

- One still ought not to resort to force, for example, without a very strong moral justification, in terms of either the threat of harm posed or the harm actually inflicted, sufficient to justify the risk and destruction that will inevitably follow.
- Moreover, any such resort to force ought to be undertaken only after reasonable alternatives to a military conflict have been attempted without success.

Likewise, in the conduct of ensuing hostilities, whether carried out with conventional forces or alternatively with robots, cyber weapons, nonlethal weapons or by biologically and/or psychologically enhanced warriors:

- One should *still* not deliberately target noncombatants and civilian infrastructure, and
- One should *still* not deliberately use more force than is necessary to attain legitimate military objectives.

These four points are defined in the long-standing moral discussion of war as “just cause” and “last resort” (with respect to the decision to engage in armed conflict) and “discrimination” and “proportionality” with respect to the conduct of hostilities. All four are basic and widely shared ethical principles long enshrined in moral discourse and reflected for the most part in international law (LOAC).

The problem instead is that the new military technologies show us that *we need to think harder about what these basic concepts mean* and how they translate and apply in these unusual new circumstances.

Here are three examples:

1. First, consider the case of UAV/RPV systems, unmanned, but remotely piloted. Suppose that a given mission in pursuit of insurgents would be permissible under law and morality using conventional manned aircraft. Such a mission would not suddenly be transformed in its legal or moral status, simply by removing the pilot from the cockpit. Either the mission is permissible or it is not. The position of the pilot with respect to the aircraft is not the determining factor in this judgment. Yet many persons think otherwise, presuming that there is something inherently illegal or immoral about such a mission, simply because it is carried out by UAVs. Morality and legality, however, are properties of the mission, not of the particular technology used to carry it out (provided, of course, that the means or weapons technology utilized is not specifically banned by treaty, as causing “superfluous injury or unnecessary suffering”).

But if this point is acknowledged, then it must also follow that the mission’s legal and moral status is likewise not inherently transformed by removing the human pilot from the equation altogether. Providing the autonomous unmanned system has been proven as safe and reliable in its operation (including demonstration of immunity to cyber attack, loss of control, or unpredictable “emergent behaviors”), then the mission in which it is employed is no more or less legal or moral, simply on its account. Yet their “intrinsic” legality or morality is among the most contested issues with respect to the further development and use of unmanned systems. Clearly, everyone involved in evaluating their development and use needs to think harder and more clearly about this issue.

2. Secondly, cyber weapons are, or have been, routinely designed to target civilians and civilian infrastructure (e.g., as unleashed in alleged Russian attacks on Georgia and Estonia or in the alleged CIA “logic bomb” used in the early 1980s to destroy Soviet-era oil pipelines). *That’s not “new,” that’s wrong.* That this deliberate, intentional cyber targeting of civilians and their property has

only recently been highlighted and criticized is again attributable to a cultural anomaly: cyber strategy is largely an extension of espionage, psychological warfare operations, and sabotage of the sort routinely carried out by covert intelligence operatives, rather than combatants. These communities operate according to different rules and assumptions. Their cover actions are always considered domestic crimes within local jurisdictions, rather than acts of war.

The newly promulgated U.S. cyber strategy has been widely summarized in the news media, in the words of an unnamed military expert: “if you launch a cyber attack against our power grid, we may just put a missile down your smokestack.” But *whose* smokestack (given the difficult problem of attribution)? And *how many* missiles down how many smokestacks? And what SORT of smokestacks (defense industries? civilian factories?) Indeed, is a conventional or “kinetic” response of this sort even appropriate as a response to a cyber attack? Would it matter what kind of cyber attack it was: DDOS? Damage to civilian infrastructure? Draining your 401k? And who is a “noncombatant” in a cyber war, during a DDOS, for example, when personal computers of the world over are hijacked in numerous and massive botnets that are collectively carrying out the strike? Clearly, all concerned need likewise to think harder and more clearly about this new technological wrinkle as well.

3. Finally, in quite another context, does the concept of “proportionate use of force” that has always governed the Law of Armed Conflict (LOAC) and specific rules of engagement for combat now dictate that nonlethal weapons should always be used before resorting to lethal weapons? Should militaries use such weapons under existing rules of engagement for lethal weapons (thereby lowering the threat of destruction and damage from mistakes), or should military and security personnel instead expand the regime (or broaden the prevailing norm) permitting the use of force more widely, in light of the presumed “nonlethality” of these alternative weapons? At present, nonlethal weapons threaten to widen the scope of permissible targets, to include civilian noncombatants as well as enemy combatants. *This is not new or revolutionary: this is a mistake*, stemming from failure to think this through. Such an unfortunate widening of the rules of engagement for the use of force has already taken place with disturbing and increasing frequency in domestic law enforcement: the rules governing the engaging of suspects with force has expanded, and so have unintended injuries and deaths stemming from use of nonlethal weapons in situations in which no police officer would have ever drawn his pistol let alone fired it.

These are some of the questions that decidedly challenge *but do not at all “overturn” or obviate* our moral conventions regarding the declaration and conduct of war. To suggest otherwise is extremely unwise and robs us of the resources required to reflect on these (frankly, quite interesting) questions.

Often such extreme declarations about the “end of war and morality as we have known it” are made by those who know little about the moral conventions surrounding war or who secretly thought them specious, nonsensical, or irrelevant

to begin with. Even military personnel, whose underlying professional conduct these moral principles reflect, are sometimes confused by all this or regard such underlying principles as superfluous, irrelevant, externally imposed constraints upon a deadly and dirty business best carried out with ruthless and brutal efficiency by those (like themselves) whom we empower to engage in armed conflict on our behalf.

What goes unrecognized in these attitudes is that military ethics is a *species of professional ethics*. That is, just like communities of physicians and health care professionals, or lawyers, or journalists, members of the military profession can also be portrayed as perpetually immersed in an ongoing dialogue regarding the nature of their practice, including aspirations for best practice, as well as professional probity and rectitude, defining the boundaries of unacceptable practice.

To those who think the “international law of armed conflict” is an external, legal constraint imposed by lawyers wearing fancy neckties in Geneva (who presumably therefore know nothing of the actual experience of war), it is important to emphasize in response that, in essence, the conventions that such lawyers may finally enshrine in some legal regime actually reflect centuries of philosophical reflection upon justified use of force and reflect the views and intuitions that we find warriors themselves discussing, for example, in Plato’s *Republic*, Shakespeare’s *Henry V*, or Sun Tzu’s classic, *The Art of War*. Such works, and the discussions of warriors they embody, attempt to define what a warrior is about and what he, or she, is trying to accomplish; what he or she is willing to do in order to accomplish it; and, perhaps most importantly, what he or she is unwilling to do in pursuit of otherwise legitimate military objectives. This characterization is true, even in a work like Baron Karl von Clausewitz’s *On War*, that seems otherwise to warn us against allowing ethical considerations to influence our judgment about strategy. This is because, at this level, as Michael Walzer observed (1977), ethics is inseparable from strategy: ethics is, ultimately, a mode of strategic thinking about goals and objectives and the best means of attaining them. Military ethics, just war doctrine, and the laws of armed conflict incorporate the consensus of warriors themselves about the nature and limitations of their professional practice.

In the military case, a reasoned, reflective equilibrium of perspectives among people of action has not been easy to come by. And its results are therefore fragile and easily threatened. It is simply not true that warriors cannot discern acceptable from unacceptable conduct, or differentiate justified from unjustifiable and wrong behavior. The consternation reported among many members of the Syrian armed forces over being forced to target fellow citizens during the beginning of their terrible civil strife, or earlier, among the Israeli Defense Forces in Gaza, illustrated this discernment in practice, as do the essays on military ethics written by Georgian, Moldavian, Turkish, Greek, and other allied officers for ethics classes taught at the Naval Postgraduate School. Whatever a soldier is, military students find themselves agreeing that the soldier exists to protect his homeland and fellow citizens, not to turn his weapons upon them. And in protecting both, he does not rain indiscriminate

and disproportionate violence haphazardly on a helpless noncombatant population, not at least without coming eventually to question the moral legitimacy of his undertaking.

Our technological innovations ought not to shatter this fragile consensus, either through their impact on military culture or through their own design and deployment at the hands of those unfamiliar, or unsympathetic, with this way of regarding public security and the rule of law.

Section XXII

Epilogue

***George J. Vachtsevanos and
Kimon P. Valavanis***

Kimon P. Valavanis and George J. Vachtsevanos

Contents

| | | |
|---------|---|------|
| 126.1 | Introduction | 2994 |
| 126.2 | The Future Economic Impact of Unmanned Aviation | 2996 |
| 126.3 | Challenges and Issues in UAS Used for Military Applications | 2997 |
| 126.3.1 | Autonomy | 2999 |
| 126.3.2 | Interoperability | 3000 |
| 126.3.3 | Airspace Integration | 3000 |
| 126.3.4 | Communications | 3000 |
| 126.3.5 | Training | 3001 |
| 126.3.6 | Propulsion and Power | 3001 |
| 126.3.7 | Manned–Unmanned (MUM) Teaming | 3001 |
| 126.3.8 | Human–Machine (UAV) Interface Issues | 3001 |
| 126.4 | Challenges and Issues in UAS Used for Civilian Applications | 3002 |
| 126.5 | The Road Ahead | 3008 |
| | References | 3009 |

Abstract

This chapter attempts to bring closure to the *Handbook of UAVs*. This, by itself, is an extremely challenging and difficult task; it should not be, but it is! After all, the reason is so simple that it was almost entirely ignored and overlooked during the course of preparing the handbook. Unmanned aviation has witnessed

K.P. Valavanis (✉)
John Evans Professor and Chair, Department of Electrical and Computer Engineering, Daniel
Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver,
CO, USA
e-mail: kimon.valavanis@du.edu; kvalavan@du.edu

G.J. Vachtsevanos
Professor Emeritus, School of Electrical and Computer Engineering, The Georgia Institute of
Technology, Atlanta, GA, USA
e-mail: gjv@ece.gatech.edu

exponential growth, worldwide, amidst controversy and difficulties, with new potential applications regularly surfacing. And although several federal agencies and multiple governments, with perhaps competitive agendas, are involved in accomplishing the common goal of integration of unmanned aviation into civilian airspace, everything seems to move so fast, that when one considers that a specific subject is completed, new changes and modifications are in effect. However, it looks like that 2013 will be “the critical year” for a new jump-start in unmanned aviation, with efforts converging and roadmaps being in place towards the ultimate goal of integration of unmanned aviation into civilian and commercial airspace. This goal remains intact and it has not changed over the years. Therefore, this chapter aims at, simply, summarizing the state of the art, the potential of unmanned aviation, referring to challenges that still need to be resolved and overcome, commenting on economic impact, and highlighting anticipated future developments. At the same time, this epilogue may also be viewed as justification for enhancing the handbook publishing a second edition.

Let's think the unthinkable, let's do the undoable, let's prepare to grapple with the ineffable itself, and see if we may not eff it after all.

Douglas Adams
(Dirk Gently's Holistic Detective Agency, 1987)

126.1 Introduction

The field of unmanned systems (US) and the related US enabling technologies has witnessed a meteoric rise over the recent past. Although “unmanned objects” of the flying or ground variety have been reported from the ancient times on, advances in sensing, communications, computing, controls, and aerospace systems over the past years have provided the impetus for the explosion being experienced in the development and utility of US across multiple domains.

The area of unmanned aviation has seen, for obvious reasons, the largest and most rapid increase in these technologies. The number of fielded systems and the range of missions supported by unmanned aviation, regardless of their acronym (UAVs, UAS, RPA, or RPAS), continue to grow at a dramatic rate. The military unmanned systems market should remain of higher value than the civilian market for the foreseeable future due to higher allocation of funding drawn from national government defense budgets, although tightened spending could have an effect on some programs and procurement. Industry has provided additional funding towards research and development (R&D) costs of national unmanned aviation programs or has developed their own systems with the hope of attracting domestic and international orders.

Although unmanned aviation technologies have expanded rapidly, worldwide, major challenges still remain to be addressed if the ultimate goal is the full utilization of such systems for a wide spectrum of applications, coupled with the

ability for unmanned–manned systems to share commercial airspace. However, in order to justify the statement “full utilization of such systems for a wide spectrum of applications,” one needs to first start by trying to answer the question(s): “*Why UAVs? What do they offer? Why are they useful, if at all?*”

The answers, nowadays, are more than straightforward and obvious. This seems hard to believe since the lack of a human pilot raises eye brows and makes the public suspicious, if not skeptical, against “those things that fly alone in the sky,” to say the least. However, the benefit of the doubt should be credited and granted to the general public. One should not forget what happened in the field of robotics about 60 years ago; it took more than 30 years fighting for survival before robotics was accepted in the USA as a viable, even better, alternative to completing a wide range of applications, which were tedious for humans. Needless to mention that when robotics was accepted in the USA, automakers in Japan had already automated their car assembly lines, and research and development on humanoids had already began. Even European industries used robots before it happened in the USA. The rationale for making the above statement stems from the fact that “those robots” were not pilotless and they did not fly alone. Therefore, it is natural to assume that it will be much more difficult for the public to accept unmanned aviation and their use in civilian applications. However, this is what the future holds, and unmanned aviation is on its way to entering everyday life. Before long, the “*Robots are here to stay*” expression, which already has been on magazine covers, will be accompanied by the “*Unmanned aviation is here to stay*” expression. The economic impact, worldwide, will be major, to say the least.

However, one should not overlook the fact that perhaps the biggest challenges related to full UAV utilization are not only technical. Instead, public acceptance, regulatory and insurance issues, and ethical and legal issues, among others, top the list. A relevant question is how might due process laws governing manned surveillance flights apply to UAS. Privacy advocates fear that UAVs could be used for surveillance of individuals; thus, privacy concerns demand a meaningful framework for discussion. The most controversial domestic use is by law enforcement agencies interested in using such vehicles for surveillance and to fight crime, a prospect that has privacy advocates and other citizens on edge. Ethical issues related to the use of UAVs are also the topic of recent discussions among technologists and social scientists. Data security is also a challenge as a regulation framework must be adapted for protection of data collected by UAS. Controversies surrounding this topic will continue to proliferate in the future as military and security operations impact the lives of noncombatants or result in privacy intrusions and other legal or social concerns.

Regardless, research activities are bound to expand in several directions motivated and supported by government and private industry. The aim will be to improve the autonomy, reliability, and safety of unmanned platforms while reducing life-cycle costs and operator workload. Among other initiatives, the National Robotics Initiative (NRI) is setting the stage for an expanded role of university research in the UAV arena. On August 3, 2013, the Office of Science and Technology Policy announced that the Department of Defense is also supporting the National

Robotics Initiative through the *Defense University Research Instrumentation Program* (DURIP). This \$40 million program, supported by the Army Research Office (ARL), the Office of Naval Research (ONR), the Air Force Office of Scientific Research (AFOSR), and the Office of the Secretary of Defense, strengthens the capability of universities to conduct research and educate scientists and engineers in areas that are important to national defense. Universities worldwide are beginning to introduce academic offerings addressing the design and operation of novel UAV concepts. About ten universities already have formally introduced degree programs in UAVs, while others offer certificates in the UAV field.

126.2 The Future Economic Impact of Unmanned Aviation

The global unmanned aviation market is expanding rapidly with projections falling behind almost on a daily basis. Several forecasts have been published, which project that the U.S. and the global unmanned aviation markets will witness a strong growth during the next 10 years.

As reported in (CRS, 30 January 2013) “A forecast of global UAS demand by the Teal Group shows worldwide annual spending on research, development, testing, and evaluation (RDT&E) and procurement rising from \$6.6 billion in 2013 to \$11.4 billion in 2022. Total spending for the decade is projected to amount to \$89.1 billion.” It is also stated that throughout the forecast period, it is anticipated that the U.S. share of RDT&E will account for 62 % of worldwide spending, while U.S. procurement will amount to 55 % of worldwide spending. According to the Teal Group, “UAS procurement will mirror demand for high-tech arms procurement in the Asia-Pacific region and in Europe.”

In March 2013, the AUVSI (with more than 500 corporate members in 2011 and still growing, representing a significant number of U.S. companies with a stake in UAS manufacturing activities) published its economic impact report (AUVSI, March 2013) documenting the economic benefits to the U.S., once UAS are integrated into the NAS. The report states that from the wide spectrum of potential civilian applications of UAS, precision agriculture and public safety will be the most promising commercial and civil markets, comprising approximately 90 % of the known potential markets for UAS. In further detail, the report concludes that (AUVSI, March 2013):

1. The economic impact of the integration of UAS into the NAS will total more than \$13.6 billion in the first 3 years of integration and will grow sustainably for the foreseeable future, cumulating to more than \$82.1 billion between 2015 and 2025.
2. Integration into the NAS will create more than 34,000 manufacturing jobs and more than 70,000 new jobs in the first 3 years.
3. By 2025, total job creation is estimated at 103,776.
4. The manufacturing jobs created will be high paying and require technical baccalaureate degrees.

5. Tax revenue to the states will total more than \$482 million in the first 11 years following integration (2015–2025).
6. Every year that integration is delayed, the USA loses more than \$10 billion in potential economic impact. This translates to a loss of \$27.6 million per day that UAS are not integrated into the NAS.

According to (CRS, 30 January 2013), in April 2012, the U.S., DOD reported that it had more than 7,100 UAS in its inventory. Further, a June 2011 analysis by the Congressional Budget Office (CBO) examined DOD UAS acquisition costs based on DOD's FY2012 budget request and reported procurement costs for the 2011–2020 period would amount to nearly \$37 billion. U.S. manufacturers with the largest share of the global UAS market include General Atomics (20.4 %), Northrop Grumman (18.9 %), Boeing (1.5 %), and AAI (1.2 %).

Analysts predict that Europe, China, and Japan will also press ahead with significant UAS development programs. Japan, for example, has already addressed safety and airspace regulatory issues, allowing civilian use of unmanned aircraft for certain applications, such as aerial spraying of pesticides. The Teal Group expects that investments by China and Japan could be larger than its forecast suggests, because less is known about those countries' intentions and programs. Forecast International projects that, while Europe will account for almost 15 % of worldwide UAS procurement, European companies will produce only 3.9 % of global production value during the next decade, slightly more than the 3.7 % share that Israel is expected to claim (CRS, 30 January 2013).

In summary, integrating unmanned aviation into the global airspace will have major economic impact, worldwide, as it will create new frontiers to civilian and public domain applications. However, the economic impact due to the growth of civilian market applications is rather long term, as opposed to the economic impact due to military applications, which will continue to be strong.

126.3 Challenges and Issues in UAS Used for Military Applications

The focus on and the perspective of UAS used for military applications will continue to grow, since UAS are preferred over manned aircraft not only because of downsizing risk and increasing confidence in mission success avoiding at the same time the human cost of losing lives if the mission is unsuccessful, but also because unmanned vehicles have better and sustained alertness over humans during dull operations. The statement that UAS are and will be best suited for “dull, dirty, and dangerous” missions has merit and it is supported because:

- Dull operations that require more than 30 or 40 hour missions are best carried out using UAS, since crew members in manned aircraft are used to much shorter duty cycles. Before the 1990s, crews were used to 4 hour sorties and missions. Fatigue and mission duration compromise proficiency and functionality of crew members; thus, the UAS alternative prevails.

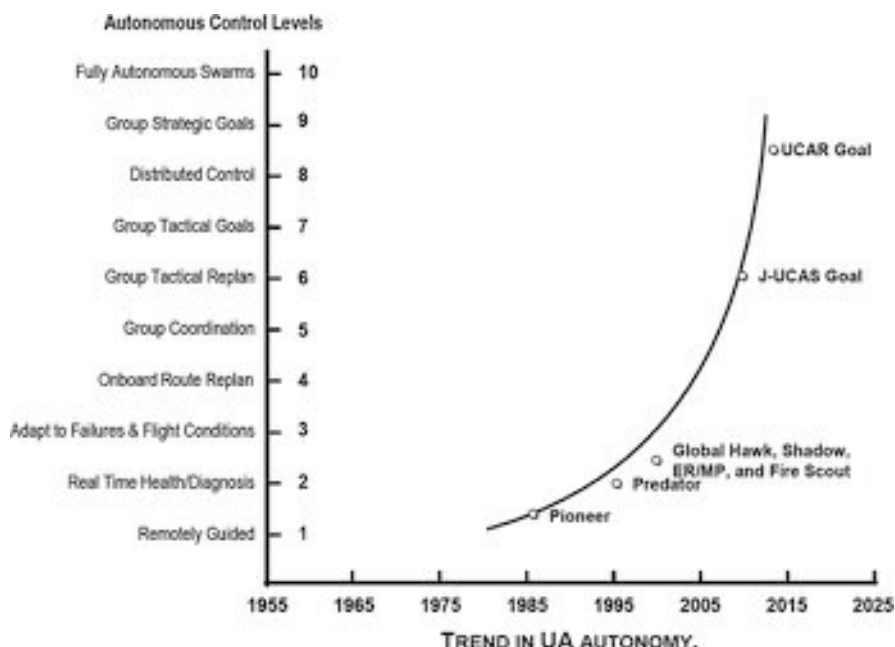


Fig. 126.1 Tentative time table trend in unmanned aircraft autonomy – U.S.

- Dirty operations may require that UAS fly into nuclear clouds (as happened in 1946–1948) immediately after bomb detonation, a mission that is clearly dangerous to human crews and threatens human lives.
- Dangerous operations like reconnaissance over enemy territory may result in loss of human lives; thus, UAS are preferred.

All recent U.S. DOD unmanned systems roadmaps envision an Air Force based mostly on unmanned planes. This becomes more evident since future urban warfare, search and rescue, border patrol, homeland security forest fire detection, traffic monitoring, and other applications will utilize an unprecedented level of automation in which human-operated, autonomous, semiautonomous air and ground platforms will be linked through a coordinated control system to perform complex missions.

The main challenge in future operations will relate to networked UAS with adaptable operational procedures, planning and asset deconfliction, and increased situation awareness, coupled with cutting-edge technologies to realize autonomous collaborative operations.

In general, technical challenges will stem from real-time sensing, computing, and communication requirements; environmental and operational uncertainty, hostile threats; and the emerging need for improved UAS and UAS team autonomy and reliability. Significant challenges will also relate to inter-UAS communications, links to command and control, contingency management, etc. The requirement and emerging need for improved UAS autonomy is depicted in Fig. 126.1 that shows

the tentative time table for autonomous control level trends. In essence, Fig. 126.1 tabulates unmanned aircraft sophistication levels from the U.S. DOD perspective, which cover the whole spectrum from the teleoperated and preprogrammed flight of a single aircraft to self-actuating and fully autonomous group vehicle flights.

Challenges increase significantly as one moves up the hierarchy of the chart from single to multi-vehicle coordinated control. Only moderate success has been currently reported in meeting the lower echelon challenges, leaving open the whole field for subsequent developments.

Technically, to meet stated challenges, innovative coordinated planning and control technologies such as distributed artificial intelligence (DAI), multi-agent system (MAS) theory, computational intelligence and soft computing, generalized system theory, as well as game theory and dynamic optimization, coupled with sophisticated hardware and software architectures, will be needed. Even though related approaches and methodologies have been investigated intensively in recent years, most work has been focused on solving particular problems, such as formation control and autonomous search, while less attention has been paid to the overall “system architecture” concept, especially from an implementation and integration point of view.

In addition to all of the above, NATO members, as part of the STANAG 4586, have been working towards “plug and play” capabilities for UAS, attempting to improve the flow of information among shared systems and enhance battlefield efficacy. STANAG 4586 includes the 4586 standard and four volumes of supporting documentation towards establishing interoperability of present and future UAS in a NATO Combined/Joint Service Environment. The STANAG 4586 Annex B is composed of three appendices, which detail Data Link Interface, Command and Control Interface, and Human–Computer Interface requirements.

When focusing on the U.S. DOD *Unmanned Systems Integrated Roadmap FY2011–2036*, the following specific major challenges in UAS technologies need to be addressed and overcome.

126.3.1 Autonomy

The proliferation of unmanned systems and the simultaneous operation of manned and unmanned systems have created a manpower burden. With limited manpower resources to draw upon, the military in the USA are seeking ways to improve the efficiency of operations. The introduction of greater degrees of autonomy promises to enable one operator to control more than one unmanned system, thus, reducing the manpower requirements. The U.S. Army, for example, aims at the “one operator-four UAVs” concept, in essence reversing the currently used “four operators-one UAV.” Additional benefits include the reduction of high bandwidth communication needs and the decrease in decision cycle time.

Improved efficiencies will be gained by automating the tasking, processing, exploitation, and distribution (TPED) of data collected by unmanned systems. Autonomy can help extend vehicle endurance by intelligently responding to the

surrounding environmental conditions (e.g., exploit/avoid currents) and appropriately managing onboard sensors and processing (e.g., turn off sensors when not needed). Increased autonomy will also assist in reducing the detrimental impact of fault/failure modes experienced by UAVs.

However, implementing a higher degree of autonomy faces the following challenges:

- Investment in science and technology (S&T) to enable more capable autonomous operations
- Development of policies and guidelines on what decisions can be safely and ethically delegated to the UAV and under what conditions
- Development of new verification and validation (V&V) and testing and evaluation techniques to enable verifiable systems

126.3.2 Interoperability

To achieve the full potential of unmanned systems, the systems must operate seamlessly across the domains of air, ground, and maritime and also operate seamlessly with manned systems. Robust implementation of interoperability tenets will contribute to this goal while also offering the potential for significant life-cycle cost savings.

126.3.3 Airspace Integration

The unmanned aviation community must continue to work with the Federal Aviation Administration (FAA) to ensure UAS have routine access to the appropriate airspace needed within the National Airspace System (NAS), to meet training and operations requirements. Similar efforts must be leveraged for usage of international airspace.

126.3.4 Communications

Unmanned systems rely on communications for command and control (C2) and dissemination of information. C2 and data transmission require a portion of the electromagnetic spectrum, which is already in very short supply. UAV users must continue to address frequency and bandwidth availability, link security, link ranges, and network infrastructure to ensure availability for operational and mission support of unmanned systems. Planning and budgeting for UAS operations must take into account realistic assessments of projected SATCOM bandwidth, and the community must move towards onboard preprocessing to pass only critical information.

126.3.5 Training

An overall U.S. DOD strategy is needed to ensure continuation, and joint training requirements are in place against which training capabilities can be assessed. Such a strategy will improve basing decisions, training standardization, with the potential to promote common courses resulting in improved training effectiveness and efficiency.

126.3.6 Propulsion and Power

The rapid development and deployment of unmanned systems has resulted in a corresponding increased demand for more efficient and logistically supportable sources for propulsion and power. In addition to improving system effectiveness, these improvements have the potential to significantly reduce life-cycle costs.

126.3.7 Manned–Unmanned (MUM) Teaming

Today's force includes a diverse mix of manned and unmanned systems. To achieve the full potential of unmanned systems, UAV users must continue to implement technologies and evolve tactics, techniques, and procedures (TTP) that improve the teaming of unmanned systems with the manned force.

126.3.8 Human–Machine (UAV) Interface Issues

With the operator in the loop, automated systems for sensing, piloting, targeting, etc., must be designed to provide maximum and reliable information to the operator (sensor, pilot) in the ground control station (GCS). A number of technical and human engineering issues must be addressed if this critical interface will function in an optimum capacity.

Remaining technical challenges also include the following nonexclusive list:

- A significant improvement in response times of single UAVs or UAV swarms.
- A significant improvement in the reliability of such systems, leading to increased trust by humans.
- Improved friend-or-foe capabilities, assessing damage and automatic IED detection.
- Improved and more accurate ability to identify a person in an urban environment from a range of (at least) 200 m.
- Improved self-learning and intelligence gathering capabilities.
- A significant boost to endurance – up to 10 additional hours for large UAVs, 80 h for medium, and 50 h for small UAVs.

- Integrated cyber-attack capabilities, coupled with improved defenses against cyber-attacks.
- Development of aerial robots with biological components.
- Transforming UAVs to perform as “soft robots,” that is, develop systems that could, for example, shrink in order to enter a blocked building through small openings and then expand again.
- Develop a perfectly camouflaged UAV, invisible to human eyes while in the air.
- Develop self-replicating robots.

Several of the above challenges, when overcome, may create transformative technology that may be applicable to civilian UAVs as well.

126.4 Challenges and Issues in UAS Used for Civilian Applications

On December 2, 2013, Amazon CEO Jeff Bezos said on *60 Min* that “Amazon.com is testing delivering packages using drones.” The idea is to use octocopters that will pick up packages in small yellow buckets at Amazon’s fulfillment centers and fly through the air to deliver items to customers after they hit the buy button online at Amazon.com. Although the technology has not yet reached such level of maturity (and despite the claim that it may be available in 4 years), is this *science fiction or reality*? Is such a “futuristic” application going to actually happen, pending “mature technology” and the FAA’s framework for rules and regulations?

As unimaginable as the above statement may sound, and it may be, it is not the first one! A few years ago, AUVSI Board Member Kitty Higgins, during an NTSB 2-day seminar on “UAS in the NAS” in Washington D.C., joked that one day “...our pizzas will be delivered by UAS.” At the same time, Douglas Davis who was at that time the head of the FAA’s Unmanned Aircraft Program Office made it clear that “no one should look to the sky for pizza just yet.” However, and this is very important, the FAA, recognizing that for some special cases UAS may be very useful and effective, did partner with police departments in Miami and Houston to evaluate the use of small UAS for surveillance (Davis 2008a,b). Along the same lines, in AUVSI’s Unmanned Systems Magazine, in a two-part interview with Margaret Gilligan, Associate Administrator for Aviation Safety with the FAA then (Gilligan 2010), she stated that “We are working hard on the rulemaking for small UAS and 2013 continues to be our goal.” However, she also stated that larger systems are more of a challenge, although there are standards. And if UAS meet these standards, then they can fly in the NAS.

A rather interesting but 6-year-old article (Cooney 2008) presents accurately what was going on in 2008 in the U.S., which is also valid today despite progress made. Because of its importance, it is presented in its entirety next:

A ton of work needs to be done by military, federal and civil aviation groups if the rapidly growing unmanned aircraft community is allowed routine access to public airspace. In a wide-ranging report on the impact of unmanned aircraft on the country’s commercial airspace, congressional watchdogs at the Government Accountability Office today called

on Congress to create an overarching body within Federal Aviation Administration to coordinate unmanned aircraft development and integration efforts.

The GAO also called on the FAA to work with the Department of Defense, which has extensive unmanned aircraft experience to issue its program plan. In addition, the Department of Homeland Security (DHS) assesses the security implications of routine unmanned aircraft access to commercial airspace, the GAO said. Even if all issues are addressed, and there are a number of critical problems, unmanned aircraft may not receive routine access to the national airspace system until 2020, the GAO concluded.

But such access is certainly on the minds of the unmanned aircraft community. That's mainly because the market for government and commercial use unmanned aircraft could explode in the coming years. Federal agencies such as the DHS, the Department of Commerce, and NASA alone use unmanned planes in many areas, such as border security, weather research, and forest fire monitoring. Researchers at the Teal Group said in their 2008 market study estimates that UAV spending will more than double over the next decade from current worldwide UAV spending of \$3.4 billion annually to \$7.3 billion, totaling close to \$55 billion in the next ten years. The forecast also indicates that the U.S. could account for 73% of the world's research and development investment unmanned flight in the next decade.

Still, routine unmanned aircraft access to the national airspace system poses technological, regulatory, workload, and coordination challenges, the GAO said. A key technological challenge is providing the capability for unmanned aircraft to meet the safety requirements of the national airspace system. For example, a person operating an aircraft must maintain vigilance so as to see and avoid other aircraft. However, because the airplanes have no person onboard, onboard equipment, radar, or direct human observation must substitute for this capability. No technology has been identified as a suitable substitute for a person on board the aircraft in seeing and avoiding other aircraft, the GAO report stated.

Additionally, the aircraft's communications and control links are vulnerable to unintentional or intentional radio interference that can lead to loss of control of an aircraft and an accident, and, in the future, ground control stations – the unmanned airplane equivalent to a manned aircraft cockpit – may need physical security protection to guard against hostile takeover, the GAO said.

There are other issues as well, the GAO report stated, including Cooney (2008):

- Many unmanned airplanes, particularly smaller models, will likely operate at altitudes below 18,000 ft, sharing airspace with other objects, such as gliders. Sensing and avoiding these other objects represents a particular challenge for unmanned aircraft, since the other objects normally do not transmit an electronic signal to identify themselves and FAA cannot mandate that all aircraft or objects possess this capability so that the aircraft can operate safely. Many small unmanned do not have equipment to detect such signals and, in some cases, are too small to carry such equipment. The Aircraft Owners and Pilots Association, in a 2006 survey of its membership, found that unmanned aircraft's inability to see and avoid manned aircraft is a priority concern.
- The effort to develop the Traffic Alert and Collision and Avoidance System (TCAS), used widely in manned aircraft to help prevent collisions, demonstrates the challenge of developing detect, sense, and avoid capability for unmanned airplanes. Although FAA, airlines, and several private sector companies developed TCAS over a 13-year period, at a cost of more than \$500 million, FAA officials point out that the designers did not intend for TCAS to act as the sole means of avoiding collisions and that the onboard pilot still has the

responsibility for seeing and avoiding other aircraft. FAA officials also point out that TCAS computes collision avoidance solutions based on characteristics of manned aircraft and does not incorporate unmanned aircraft's slower turn and climb rates in developing conflict solutions. Consequently, FAA officials believe that developing detect, sense, and avoid technology that unmanned aircraft would need to operate routinely in the national airspace system poses an even greater challenge than TCAS did. FAA officials believe that an acceptable detect, sense, and avoid system for airplanes could cost up to \$2 billion to complete and is still many years away.

- The lack of protected radio frequency spectrum for unmanned operations heightens the possibility that an operator could lose command and control of the plane. Unlike manned aircraft, which use dedicated, protected radio frequencies, unmanned aircraft currently use unprotected radio spectrum and, like any other wireless technology, remain vulnerable to unintentional or intentional interference. This remains a key security vulnerability for unmanned aircraft, because in contrast to a manned aircraft where the pilot has direct, physical control of the aircraft, interruption of radio frequency, such as by jamming, can sever the plane's only means of control. One of the experts we surveyed listed providing security and protected spectrum among the critical airplane integration technologies.
- Unmanned aircraft have the capability to deliver nuclear, biological, or chemical payloads and can be launched undetected from virtually any site. In response to the events of September 11, 2001, entry doors to passenger airplane cockpits were hardened to prevent unauthorized entry. However, no similar security requirements exist to prevent unauthorized access to unmanned aircraft ground control stations – the unmanned system equivalent of the cockpit. Security is a latent issue that could impede unmanned airplane developments even after all the other challenges have been addressed, according to one study.
- Although U.S. DOD has obtained benefits from its unmanned operations overseas, the agency notes in its Unmanned Systems Roadmap that unmanned aircraft reliability is a key factor in integrating unmanned systems into the national airspace system. Our analysis of information that U.S. DOD provided on 199 military unmanned airplane accidents, of varying degrees of severity, that occurred over 4.5 years during operations Enduring Freedom and Iraqi Freedom, indicates that reliability continues to be a challenge. About 65 % of the accidents resulted from materiel issues, such as failures of aircraft components. FAA officials noted that unmanned aircraft today are at a similar stage as personal computers in their early years before newer, more user-friendly operating systems became standard.
- The variety of ground control station designs across unmanned aircraft is another human factors concern. For example, pilots of the Predator B control the aircraft by using a stick and pedals, similar to the actions of pilots of manned aircraft. In contrast, pilots of the Global Hawk use a keyboard and mouse to control the aircraft. Differences in unmanned system missions could require some variation

among control station designs, but the extent to which regulations should require commonalities across all ground control stations awaits further research.

- Because unmanned aircraft have never routinely operated in the national airspace system, the level of public acceptance is unknown. One researcher observed that as unmanned aircraft expand into the nondefense sector, there will inevitably be public debate over the need for and motives behind such proliferation. One expert we surveyed commented that some individuals may raise privacy concerns about a small aircraft that is “spying” on them, whether operated by law enforcement officials or by private organizations, and raised the question of what federal agency would have the responsibility for addressing these privacy concerns.

Feedback from European agencies was pretty much the same and along the same lines of thought. D. Hughes’ article (2008) stated that “Unmanned aircraft need lots of flying hours to be ‘civilized.’” The article talks about the Eurocontrol plans to develop a roadmap by the end of 2009 outlining how unmanned aircraft will be integrated into the European ATM system. A key issue is harmonization with what the FAA, ICAO, EUROCAE WG-73, and RTCA SC-203 are deciding. According to H. Matthiesen, Senior Specialist in ATM (Hughes 2008), *“What the industry needs is flying hours and experience and lessons [learned] to build confidence in these machines. Airworthiness requires lots of flying hours, especially to support airworthiness on the civil side.”*

Further, M. Bennett (2008) stated that the European Defense Agency (EDA) awarded in January of 2008 a contract to a consortium of defense and aerospace companies to develop a detailed roadmap for integrating UAS into European airspace so that they can fly routinely with other air traffic by 2012 or “the end of 2015 at the latest.”

The previously stated information reveals a very important pattern: The “thought process” in U.S. and European agencies, despite possible differences, has been similar if not identical. Therefore, what appears to be obvious as the aftermath of the same thought process that also includes valid concerns and skepticism is the publication of the roadmaps for integration of UAS into the NAS (Final Report – European RPAS SG, 2013) (Report-FAA, 2013), which lay the foundations for unmanned aviation to be eventually widely used for a multitude of civil and public domain applications.

However, all of the above notwithstanding, there is consensus on a number of challenges that need to be met before UAS fly routinely in civilian airspace. The following are true:

- Safety, safety, and more safety, with all prerequisites and aftermath attached to it. The public will not tolerate accidents; it is as simple as that. For UAS to achieve an equivalent level of safety (ELOS) to that of manned aircraft system, reliability needs to be increased and the probability of midair collisions needs to be reduced. Collision avoidance has been a major area of research since most of the corresponding manned aviation procedures and technology rely primarily on pilots and secondarily on ATC instruction.

- Detect, sense, and avoid technology. The NTSB members have expressed particular interest in the ability of UAS to handle contingencies beyond close encounters in shared airspace. This is perhaps the most important challenge to be overcome, as the FAA states that UAS should function as if there were a pilot onboard. In addition, redundancy is a must, and as such, it is obvious that sense and avoid technology should include both visual and nonvisual sensor components. Such a system should be capable of operating under various weather conditions and situations and, as autonomy increases, with limited operator involvement. The latter is indeed important since different requirements can be expected depending on whether a pilot will be in the decision loop or not. In any case, a successful detect, sense, and avoid (and see and avoid) system will likely entail information fusion from multiple sensors to avoid the “blind spots” of a single sensor. It will also be necessary for the system to interface with ground-based sensors (e.g., radar) for supplementing its own information and with other sensors and onboard systems to be able to make successful decisions for collision avoidance and execute them with precision and timeliness. Based on the principles of fairness and equivalence, it would be reasonable to demand that such systems achieve a performance comparable to that of human pilots. Nevertheless, both research and practice has shown that pilots are not very good at detecting potential collisions. Moreover, currently manned aviation pilots can only monitor a limited part of the airspace around them, mostly in front of the aircraft. As a result, the development of detect, sense, and avoid systems that are capable of exceeding pilot capabilities would be beneficial for the manned aviation sector, as well. In any case, minimum requirements will be mandated that will include minimum detection distances and aircraft separation margins. The “reaction time” of the system should also be at least comparable to that of a pilot, based on the FAA AC 90–48C report. Also, although research has been focused on duplicating pilot vision, efficient deconfliction algorithms will also need to be developed to replace pilot action in the case of a possible collision.
- Sensors development. From a regulation standpoint, sensor capabilities are an issue, since they will determine detect, sense and avoid, and see and avoid performance, which in turn is a major factor affecting safety of operations. Related research has investigated electro-optical, acoustic, and microwave sensors. Acoustic sensors are not very susceptible to problems from the environment and are quite robust but have low resolution and exhibit problems tracking multiple targets. Electro-optical sensors have good resolution and allow multiple target tracking but are more susceptible to environmental interference (sun glare, temperature, vibration). Microwave sensors can give good range estimates but are typically too large, heavy, and expensive especially for smaller UAS, although research is being done on ultra-lightweight radar systems for this class of UAS. When combined, these sensors offer unique characteristics that enable a UAS to detect and in some cases track one or more targets in difficult conditions like fog, glare, or darkness. Besides developing sensors with higher resolutions and improved capabilities, efficient fusion algorithms will also be needed.

Finally sensor miniaturization will play an important role in the integration of small UAS.

- **Bandwidth regulation.** The United Nations International Telecommunications Union (ITU) is working with the International Civil Aviation Organization (ICAO) to provide a safe, global communications infrastructure for UAS operations. One option presented to ITU by the U.S. government – with industry support – is the use of fixed satellite service, which is in abundance and can safely support the projected growth of the UAS market for years to come.
- **Lost link procedures:** In all cases, the UAS must be provided with a means of automatic recovery in the event of a lost link. There are many acceptable approaches to satisfy the requirement. The intent is to ensure airborne operations are predictable in the event of lost link.
- **Flight termination system (FTS):** It is highly desirable that all UAS have system redundancies and independent functionality to ensure the overall safety and predictability of the system. If a UAS is found to be lacking in system redundancies, an independent flight termination system that can be activated manually by the UAS pilot in command may be required to safeguard the public.
- **Autonomous operations:** At first only those UAS that have the capability of pilot intervention, or pilot-in-the-loop, shall be allowed in the NAS outside of restricted, prohibited, or warning areas. UAS that are designed to be completely autonomous, with no capability of pilot intervention, are going to be the last to be authorized for operations in the national airspace system.
- **Technology testing and evaluation.** In parallel with standards and regulations development, other efforts are required to streamline the integration process of UAS in the NAS. Existing regulations severely limit access to airspace so that UAS developers can evaluate their systems. As a result, the FAA announced the establishment of six test sites (December 2013) to evaluate UAS and their subsystems for both R&D as well as certification purposes. Similar centers are planned or have already been built in France, Spain, Canada, Australia, and New Zealand. Such centers can be used to certify UAS hardware and software components as well as provide UAS crew training, as is the goal for the ARCAA in Australia.

In addition, new modeling techniques will be required to capture coupling between individual system/sensor dynamics, communications, etc., with “system of systems” behaviors. Hybrid system approaches will play a key role in capturing complex behaviors and defining the means to represent and manage uncertainty, including spatial–temporal models of distributed sensors to integrate system and motion dependencies, contingency planning, and situation awareness.

Intelligent and hierarchical/distributed control concepts must be developed and expanded to address “system of systems” configurations. Game theoretic approaches and optimization algorithms running in real time to assist in cooperative control and adversarial reasoning will be needed. Comprehensive approaches to control of networks of dynamic agents will be essential to tackle coordination and cooperation issues.

Networking and communications will deviate from traditional communication protocols and standards, requiring novel solutions to overcome jamming. Security metrics will need to be redefined.

Sensors and sensing strategies will need innovative technologies and solutions that will include wireless communications. This will be coupled with building improved, reliable, and cost-effective sensor suites, as well as “smart” sensors, leading to better sensing strategies for massive data processing, data mining, and sensor fusion.

The need for defining new “system of systems” performance and effectiveness metrics for verification, validation, and assessment of networked systems is more than obvious.

126.5 The Road Ahead

During the last almost 20 years, we have witnessed significant advances in the unmanned aircraft and unmanned systems state of the art.

Major research and development projects, sponsored primarily by the military sector, have contributed towards development and deployment of aerial, under-sea, and ground unmanned systems with new designs surfacing at an increasing rate. Common objectives and goals of any such designs are improved autonomy through innovative sensing and control strategies, enhanced reliability and endurance through fault-tolerant control, advanced materials, and high-efficiency power plants.

Yet, new and novel concepts and technologies are required for a more widespread use of such critical assets, not only for military but also for commercial and other applications such as homeland security, rescue operations, forest fire detection, and delivery of goods, to name just a few applications.

Federated systems consisting of multiple unmanned aerial vehicles performing complex missions present new challenges to the control community. UAS must possess attributes of autonomy in order to function effectively in a “system of systems” configuration.

Coordinated and collaborative control of UAS swarms demands new and novel technologies that integrate modeling, control, communications, and computing concerns into a single architecture. Typical application domains include reconnaissance and surveillance missions in an urban environment, target tracking and evasive maneuvers, search and rescue operations, homeland security, etc.

Major technological challenges remain to be addressed for such UAS swarms, or similar federated system of systems configurations to perform efficiently and reliably. Excessive operator load, autonomy issues, and reliability concerns have limited thus far their widespread utility. The systems and controls community is called upon to play a major role in the introduction of breakthrough technologies in this exciting area.

References

- Aerospace Industries Association (AIA), UAS: perceptions and potential, report.
- Association for Unmanned Vehicle Systems International (AUVSI), The economic impact of unmanned aircraft systems integration in the US, report, Mar 2013
- M. Bennett, Schedule for UAS flights in Europe slips, but hope stays on track. *Unmanned Syst. Mag.* **26**(5), 30–31 (2008)
- Congressional Research Service (CRS), UAS: manufacturing trends, 30 Jan 2013
- M. Cooney, Unmanned aircraft pose myriad problems to US. *Airspace*, GAO reports (2008), <http://www.networkworld.com/community/node/27876>. Accessed June 2008
- B. Davis, Small UAS to take to the skies first, FAA says, but hurdles remain. *Unmanned Syst. Mag.* **26**(1), 32–33 (2008a)
- B. Davis, UAS in the national airspace: the NTSB takes a look. *Unmanned Syst. Mag.* **26**(6) (2008b)
- P. Gilligan, Unmanned aircraft in civil airspace: a chat with the FAA. *Unmanned Syst. Mag.* **28**(8), 17–21 (2010)
- D. Hughes, UAV road map for Europe. *Aviat. Week Space Technol.* **168**, 78–79 (2008)
- Integration of Civil UAS in the National Airspace System (NAS)*, 1st edn. (FAA, 2013)
- A. Kirschbaum, Plug and play for UAS: STANAG 4586 shows the way. *Unmanned Syst.* **26**(5), 28–29 (2008)
- Roadmap for the Integration of Civil Remotely-Piloted Aircraft Systems into the European Aviation System, Final report from the European RPAS Steering Group, June 2013
- TEAL Group, World unmanned aerial vehicle systems, 2011 market profile and forecast
- US Department of Defense Office of the Secretary of Defense, OSD UAV roadmap 2002–2027 (2002)
- US Department of Defense Office of the Secretary of Defense, Unmanned aircraft systems roadmap 2005–2030 (2005)
- US Department of Defense Office of the Secretary of Defense, Unmanned systems roadmap 2007–2032 (2007)
- US Department of Defense Office of the Secretary of Defense, Unmanned systems integrated roadmap FY2011–2036 (2011)
- K.P. Valavanis (ed.), *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*. Intelligent Systems, Control and Automation: Science and Engineering, vol. 33 (Springer, Dordrecht, 2007)

Index

A

- Acceptable level of safety (ALoS), 2238–2239
- Accident(s), 2200, 2202–2210, 2215–2218, 2220–2223, 2225, 2226, 2231, 2239, 2242, 2244–2247, 2249–2251, 2253, 2256, 2261, 2265, 2267, 2269, 2270
- Acquisition, 1312, 1314, 1326
- Action, 859, 876, 880, 889, 890, 893, 900, 911–913, 915, 916, 922, 925–930, 942
- Activity, 872, 880, 883, 885
- Actuator saturation, 617, 629
- Adaptive control of unmanned aerial vehicles, 613–671
- Adaptive two-stage extended Kalman filter (ATSEKF), 1169, 1175, 1177–1179
- Adaptive two-stage Kalman filter (ATSKF), 1131, 1134, 1143–1145, 1147, 1150–1153
- Additive white Gaussian noise (AWGN), 1909
- Ad hoc, 783, 786, 789, 791, 793, 794, 796, 797, 802, 806, 808
- Ad-hoc aerial relay networks, 753–754, 773
- Administrative Procedures Act, 2170
- ADS-B. *See* Automatic Dependent Surveillance-Broadcast (ADS-B)
- Advisory Circulars (ACs), 2170, 2173
- Aerial images, 2646, 2660, 2661
- Aerial manipulation, 213
- Aerial measurements, 2670–2672
- Aerial sensor networks, 754, 760, 770–773
- Aerobatic miniature helicopter, 280
- Agent, 856, 858, 859, 890, 891, 905, 906, 914, 920, 930–933
- Aided-inertial localisation, 434, 435, 439, 443–453, 458
- Airborne Collision Avoidance System (ACAS), 1868–1870
- Aircraft, 2161–2169, 2171, 2172, 2174
- design, 170
- sizing, 167–171, 173–180
- specifications, 115
- Air Force Research Laboratory (AFRL), 2509, 2513, 2515, 2517, 2519
- Air safety, 2342, 2343, 2345–2349
- Airspace, 1842, 1844–1847, 1851, 1852, 1855
- classes, 2130–2132
- issues, 2724, 2728–2730
- Air Support Operations Center (ASOC), 2513–2517, 2521, 2522
- Air-to-Air Communication, 751, 764
- Air-to-ground communication, 751, 763, 764, 777
- Air traffic control, 2137, 2138, 2142, 2146, 2148–2156
- Airworthiness, 1843, 1844, 1852, 2118, 2119, 2122–2126, 2128, 2130, 2132, 2179, 2180, 2185–2187, 2192, 2278, 2279, 2284, 2289, 2290, 2339–2358
- Alliger, G.M., 2519
- American Society of Testing & Materials (ASTM), 2170, 2174
- Amplitude comparison monopulse (ACM), 1906
- AMS sensor, 2734, 2736, 2746, 2748
- Annex 2, 2164–2168, 2170
- Annex 6, 2167, 2168
- Annex 7, 2170
- Annex 11, 2164, 2165
- Anomaly detection, 1006
- Antenna, 1417, 1430–1437
- Approximate model inversion based adaptive control, 616, 617, 634–636
- Aquatic weeds, 2815, 2819–2825, 2830, 2832
- Architecture(s), 849–948, 953–972, 1725–1746
- Article 1 of the Convention, 2164
- Article 2 of the Convention, 2164
- Article 3 of the Convention, 2163

Article 8 of the Convention, 2163
 Article 12 of the Convention, 2164, 2165
 Assignment problems, 2081
 Assignments, 2054, 2055, 2066–2072
 Asynchronous optimization, 1750
 Athena Vortex Lattice (AVL) computational fluid dynamics software, 114
 Attitude, 348–359, 361, 363, 364, 366, 367, 369
 Auctions, 1478, 1482
 Auftragstaktik, 1274
 Aural displays, 2457
 Automated contingency management, 1001–1002, 1012–1015, 1019–1024
 Automated design analysis, 1050
 Automatic, 1277, 1278, 1283, 1284, 1286
 operation, 1860, 1868
 plane, 60
 Automatic Dependent Surveillance-Broadcast (ADS-B), 1897, 1965–1972, 1974–1978, 2372
 Automation, 2142–2146, 2148, 2151, 2152, 2155, 2156
 Autonomy(ous), 1273–1297, 1306, 1309, 1726–1728, 1730, 1732–1737, 1739, 1740, 1742–1744, 1746
 agents, 752
 flight, 114
 maneuver, 1845
 UAV, 1603
 Autopilot, 350, 359, 361, 364, 367, 369, 372, 377, 1859, 1860, 1868
 Aviation Rulemaking Committee (ARC), 2174
 Avoidance manoeuvre, 1860, 1868, 1869

B

Bandwidth, 1281, 1282, 1284, 1285, 1291
 Barracuda UAS, 1859–1861, 1864, 1866
 Battery weight fraction, 175
 Bayesian, 465, 466, 469, 471, 473, 474
 Bayes' law, 1091
 Beamforming, 1434–1437
 Beard, R., 2519
 Bearing-only SLAM, 401, 403, 415–423
 Behaviors, 1726–1737, 1741–1744, 1746
 Bell-Hiller stabilizer bar, 1221
 Bennett, W., 2519
 Best practices, 2294, 2333
 Biological inspiration, 1308
 Biologically, 1385–1412
 Biomimetic, 1308, 1310
 Brain activity, 2478, 2480–2483, 2497
 Budget, 1425, 1435, 1436

C

Cacti, 2815, 2817, 2827, 2829–2832
 Canons of good governance, 2892
 Cantwell, H.R., 2510, 2511
 Cascaded control systems, 617
 Causal factor, 1819, 1820, 1823–1828, 1830–1838
 Causal model, 1819, 1824–1828, 1837
 Centralized, 977–991
 Centralized planning, 1457, 1481
 Certificate of Authorization (COA), 2179–2197
 Certification, 1850–1852, 2119, 2122–2127, 2132, 2277–2290, 2361–2380
 Certification process, 2279–2285, 2290
 14 CFR. *See* Title 14 Code of Federal Regulations (14 CFR)
 CFR Part 43, 2172
 CFR Part 61, 2172
 CFR Part 91.1, 2168
 CFR Part 91.111, 2166
 CFR Part 91.113, 2167
 Chicago Convention of 1944, 2162
 Chinese postman problem, 1511, 1517–1519, 1543, 1544
 Civil Aviation Authorities (CAA), 2162, 2166, 2169, 2171, 2173
 Civilians, 2881, 2886–2889, 2891, 2896, 2898
 CK. *see* Control Kernel (CK)
 Clap-and-fling, 1371–1373, 1378–1380
 Close Air Support (CAS), 2511, 2513
 Cluster Breathing, 754, 766, 767, 770
 Coalition formation, 2021–2046
 Coaxial helicopter, 1219–1223, 1225–1239, 1241, 1245, 1246, 1252, 1255
 Cognitive assessment, 2483–2492, 2495
 Cognitive performance, 2477, 2492
 Cognitive radio, 813–843
 Colegrove, C.M., 2519
 Collective pitch, 1219–1221, 1224, 1229
 Collision avoidance, 390–394, 397, 1549–1552, 1561–1564, 1572, 1574, 1778, 1789–1792, 1956–1959, 1961, 1962, 1964–1969, 1972–1974, 1976–1978, 2362, 2364, 2365, 2367, 2369–2371, 2373–2376, 2378
 Column generation, 1603–1605, 1609
 Command and control (C2), 781–809, 1273–1297, 2506, 2507, 2515, 2788, 2791, 2792, 2797, 2801, 2805, 2808
 Command, control, and communication, 814, 819, 2294, 2312–2322

- Communication, 715–748, 1306, 1307, 1309,
1310, 1415–1438
constraints, 1632–1637
cost minimization, 1755
delays, 1752, 1753, 1755–1757,
1760–1763, 1766–1769, 1771
relays, 1512, 1621, 1622, 1643–1649
Communication-Aware Networking
Algorithms, 766–771
Communication-Aware Potential Fields, 757,
769–771, 773
Computational complexity, 1605,
1615, 1616
Computational unit, 904
Concept of operations, 2339–2358
Concurrent plan, 914–918
Conflict recognition and resolution, 1888,
1892
Congress, 2160, 2161
Consensus, 1473–1479, 1482, 1491–1506
Constraint analysis, 168–174, 176, 177
Constraints, 858, 859, 865, 884, 889, 891–893,
895–900, 904, 906, 912–918,
922–931, 947
Contingency management, 2325, 2326,
2329, 2333
Continuous wave (CW), 1899–1902
Contra-rotating rotors, 1220, 1221
Control(s), 2136, 2137, 2140–2157,
2443–2469
derivatives, 330, 331, 338–340, 345
formula, 914–918
layer, 853, 855–857, 862–871, 884, 911
station, 2136, 2138–2148, 2150, 2152,
2153, 2155, 2156
system, 111, 127, 1703–1718
Control Kernel (CK), 851, 857, 862–865, 877,
878, 884–888, 893, 894, 900, 937,
943, 947
Convoy protection, 2049–2075
Cooperative control, 1620, 1634–1636,
1649, 2080
Cooperative intruder, 1859, 1865
Coordinate frames and transformations,
245–246, 253–254
Coordination, 2050, 2061, 2065
algorithms, 1632
network, 1602, 1603
Corrugation, 1353–1356
Coverage control, 1751, 1752,
1764–1771
Criminal negligence, 2871–2873
Crowley, D., 2508
Cruising altitude, 112
Cyber warfare, 2988
Cyclic pitch, 1219–1221, 1224, 1229, 1231,
1236–1238
- D**
Damage threshold, 1060
Data alignment method, 2657
Data fusion, 424–425, 429, 430, 434,
446, 458
Data Interface (DI), 885, 888, 904, 909
Data routing, 783, 789–791, 799
Decentralized, 977–991
autonomous control, 1577–1598
control, 2085
data fusion, 462–470, 472–477, 483, 487
planning, 1457, 1481
Decision making, 1438, 1439, 1510, 1512,
1531–1534, 1544
Deductive reasoning, 1821
Delegation, 859, 925, 926, 931
DelFly, 1377–1380
Deliberative layer, 853–855, 864, 866, 870,
884, 893, 910–930, 946, 947
Deliberative/Reactive Computer (DRC), 861
Department of Defense (DoD), 2506–2510
Design approval, 2279
Design synthesis, 165–180
Detect and avoid, 1873–1893, 1955–1979
Detection, 1071–1119
DI. *See* Data Interface (DI)
Diagnosis/Diagnostics, 1006, 1007, 1009,
1019, 1020, 1024, 1029, 1030,
1038–1040, 1042, 1047, 1050
Dielectric resonant oscillator (DRO), 1904
Differential flatness, 314–317, 319, 322
Direct current (DC), 1901, 1905, 1915
Direct measurement, 1239, 1240, 1255
Discrimination, 2973–2974, 2977, 2980, 2987
Displacement transfer functions (DTF), 128,
129, 136
Display(s), 2138–2141, 2144, 2146–2148,
2150, 2154, 2443–2469
Distributed auction, 1494
Distributed blackboard, 1289–1291,
1293, 1294
Distributed Mission Operations (DMO), 2515,
2516, 2518, 2519
Distributed optimization, 1603, 1749–1772
6DOF motion, 246
Doppler, 1901–1903, 1909, 1912, 1913, 1916
Dragonfly, 1389
DRC. *See* Deliberative/Reactive Computer
(DRC)

Drone, 58, 60–68, 978, 1703–1718
 Dual simplex ascent, 1604, 1605, 1610, 1611
 Dual unscented Kalman filter (DUKF),
 1157–1180
 Dubins airplane, 1677–1699
 Dubins car, 1679, 1688–1691, 1693–1695
 Dubins path, 1511–1517, 1519–1522, 1525,
 1526, 1528, 1530, 1531, 1539,
 1543, 1544
 Ducted fan, 1330, 1344–1346
 Due care, 2872, 2873
 DyKnow, 856, 858, 885, 903–906, 908, 909,
 914, 918, 947
 Dynamic co-fields (DCF), 1291–1294
 Dynamic vehicle routing, 2081–2084,
 2104

E

Electric aircraft, 173–180
 ELOS. *See* Equivalent Level of
 Safety (ELOS)
 Embedded controls, 1402
 Embedded fiber bragg gratings (FBGs),
 128, 134
 Emergency procedures, 2179, 2180,
 2187–2191, 2193
 Emergency recovery, 2294, 2322–2333
 Emergent behavior, 1306, 1309
 Endurance, 111, 112, 121
 Energy density, 175, 177, 178, 1306–1308,
 1310
 Energy efficiency, 1387, 1395, 1412
 Enewold, S.L., 2508
 Engineering design specifications, 2873
 Entomopter, 1308
 Entropic drag, 1280–1282, 1286
 Environmental robotics, 2832
 EO camera, 1965
 Equivalent level of safety (ELOS), 1875,
 1878, 1886–1888, 1892, 1893,
 2200–2205, 2238–2243, 2260
 ESD. *See* Event sequence diagram (ESD)
 Ethics, 2865–2875, 2886, 2888, 2893, 2903,
 2904, 2966, 2977
 EUROCONTROL, 2162
 European Aviation Safety Agency (EASA),
 2162
 Evasive target, 1776, 1777, 1782, 1787,
 1797–1801, 1806
 Event, 871, 874–895, 900–903, 910
 Event-driven, 1749–1772
 Event sequence diagram (ESD), 1825–1828,
 1837

Excitation signals, 1099, 1100, 1103, 1107,
 1109–1111, 1114
 Executor, 854, 858, 890, 893–895, 900, 914,
 921, 947
 Experimental validation, 1123–1153
 Extended information filter (EIF), 403,
 423, 424
 Extended Kalman filter (EKF), 403, 411, 412,
 414, 416, 417, 420, 421, 423, 426,
 428, 429, 434, 443–459
 External forces and moments, 267, 268, 276
 Eye-in-the-sky, 2649

F

Failure, 1072–1077, 1086–1088, 1090,
 1094–1100, 1102, 1108, 1114,
 1116
 Failure modes and effects analysis (FMEA),
 1028, 1030–1047, 1049, 1050, 2577,
 2579, 2580, 2592
 Failure prediction, 1010
 False alarms, 2651, 2653, 2662
 Fault, 1071–1119
 diagnosis, 1183–1210
 mode, 1054, 1056
 reasoning, 1001
 Fault detection and diagnosis (FDD),
 1123–1153, 1157–1180
 Fault detection and isolation, 1184–1195,
 1198–1210
 Fault-tolerant control (FTC), 599–603, 606,
 1124, 1126, 1128–1130, 1136
 FCL. *See* Flight Control Language (FCL)
 FDI (EMMAE-FDI, SMAC-FDI), 1074–1077,
 1080, 1084–1086, 1093, 1095–1118
 Feature analysis, 2653
 Feature points, 2655
 Federal Aviation Administration (FAA), 1897,
 1930, 1937, 1948–1950, 2136–2139,
 2147, 2149, 2152, 2154, 2156, 2162,
 2169–2171, 2173, 2174, 2178–2183,
 2185–2188, 2191–2197, 2508, 2511
 Federal Aviation Regulations (FAR), 2161,
 2166, 2168, 2172, 2173
 Fiber optic sensors, 128
 Fiber optic strain sensing technology (FOSS),
 127–136
 Finite element model (FEM), 120, 122,
 125, 132
 Fire sensors, 2736
 First order rotor flapping, 280, 290
 Fixed-pitch coaxial helicopter, 1220–1223,
 1225, 1227–1239, 1252

Fixed wing control, 656
 Flapping mechanisms, 1401, 1411
 Flapping wing, 329–345, 1331–1334, 1336, 1337, 1340–1343, 1354–1356
 Flapping-wing propulsion, 1359–1382
 Flexible wing, 1331, 1333, 1335–1343, 1354, 1355
 FLight AlaRM (FLARM), 1897
 Flight control, 677, 678, 709, 1389, 1402–1404, 1407, 1411
 Flight control architecture, 1680
 Flight Control Language (FCL), 855, 884–886, 943
 Flight control system (FCS), 110, 136–139, 1860, 1864, 1866, 1870
 Flight data analysis, 1255
 Flight demonstration, 1858, 1865
 Flight dynamics, 677–681
 Flight experiment, 1195, 1198–1210, 1568–1575
 Flight Information Regions (FIRs), 2165, 2166, 2168, 2169
 Flight operations, 2342, 2343, 2349–2351, 2357
 Flight safety, 817, 819, 843
 Flight termination, 2294, 2322–2333
 Flight vehicle modeling and control, 548–549
 Flow sensor mimicking hairs, 1348–1353
 Force closure, 208–213, 231, 236
 Force multiplication, 2869–2872
 Formation control, 955, 957, 963–965, 972
 Formation flight, 1705, 1718
 Frames of reference, 478, 480
 Frequency agile radio, 816
 Frequency bands, 720, 729–731, 742, 745–747
 Frequency-modulated continuous wave (FMCW), 1901, 1902
 Frequency-shift-keyed continuous wave (FSKCW), 1901, 1902, 1904, 1909–1928
 Frequency sweep, 1246, 1250, 1251
 Fuel balancing, 2066, 2069
 Fuel fraction, 168, 171–173
 Functional magnetic resonance imaging (fMRI), 2479, 2481
 Functional Near Infrared Spectroscopy (fNIR), 2475–2498
 Functional-reasoning, 1034
 Fuselage drag, 1231–1232, 1234

G

Gap analysis, 2293–2334
 Gaussian processes, 1656–1658, 1660, 1674

Gertler, J., 2506
 Global harmonization, 2174
 Global Hawk, 2506, 2507, 2509, 2510, 2521
 Global Positioning System (GPS), 434, 435, 453–459
 Glyph symbology, 2450–2452
 Governance, 2879–2908
 Government Accountability Office (GAO), 2508
 GPS, 352–357, 359, 361
 Graphical User Interface (GUI), 796, 799–802, 804, 807
 Greedy randomized adaptive search procedures, 1590
 Ground control station (GCS), 127, 1315, 1318, 1320, 1323–1326, 1864, 1866
 Ground effect, 1371, 1372
 Ground impact risk, 2221
 Ground target tracking, 1580
 Guard, 871, 874, 876, 884
 Guderian, Heinz, 1274, 1280, 1286

H

Hardware-in-the-loop (HIL), 136
 Hardware schematic, 860, 861
 Harm, 2968–2970, 2979
 Hazard(s), 1818–1820, 1822–1828, 1830–1838, 2238–2245, 2247–2256, 2261, 2262, 2264–2266
 Hazard causal events, 2171
 HCSM. *See* Hierarchical Concurrent State Machines (HCSM)
 HDRC3, 849–948
 Headlock gyro dynamics, 1233
 Health monitoring, 1703–1718
 Health state, 1057, 1062
 Helical path following, 1679, 1682
 Hemoglobin, 2486
 Hierarchical Concurrent State Machines (HCSM), 855, 858, 864, 870–884
 Hierarchical system, 1551
 High-Altitude Long-Endurance (HALE), 2731–2732
 History of regulation, 2160–2163
 House Permanent Select Committee on Intelligence, 2508
 Human cognitive factors, 2497
 Human factors, 2135–2157, 2367–2368, 2497
 Human factors engineering (HFE), 1312–1315, 1321, 1322, 1326
 Human interaction, 1309
 Human Machine Interface (HMI), 1860–1863
 Human performance, 2136, 2150, 2153

Human-swarm interactions

- algorithms, 2017
- models, 1987–2017
- performance, 2017

Human systems integration (HSI), 1312–1316,
1321, 1322, 1324, 1326, 2135–2157

I

ICAO Doc 7030, 2165

ICAO UAS Study Group, 2170, 2174

IFR. *See* Instrument Flight Rules (IFR)

ILS elements, 2607–2610

Inductive reasoning, 1821

Inertial, 401–430, 433–459

- navigation, 442
- sensor, 1183–1210
- SLAM, 403–430

Inertial localization system (ILS), 2567, 2570,
2573, 2579–2581, 2583, 2585, 2592

Inertial measurement unit (IMU), 1185, 1188,
1193, 1195, 1197–1199, 1201, 1203,
1207, 1210

Inertial navigation system (INS), 352–357,
359, 361

Information analysis, 2431–2434, 2437

Information fusion, 465

Information routing, 1625, 1630–1631,
1643, 1646

Information-theoretic path planning, 1657,
1662–1669

Information theory, 1279, 1282, 1286

Inner loop control, 350, 359–369, 374, 377

Inner-outerloop control of an unmanned
helicopter, 616, 634

Insect-sized MAV, 1329–1356

Instructor operator station (IOS), 2515

Instrument flight rules (IFR), 1819–1828,
1831, 1833, 1834, 1836, 1837

Integer programming, 1448–1456, 1477, 1483

Integrated Combat Operations

- Training-Research Testbed
(ICOTT), 2505–2522

Integrated Logistics Support, 2606–2613

Integration, 145

Integration concept, 1873–1893

Intelligence, surveillance, and reconnaissance
(ISR), 2506, 2511

Intentional cooperation, 955, 958, 959, 965,
967–969

Interface, 2444, 2446, 2448, 2452, 2453, 2456,
2458, 2463–2469

Interference-Aware Positioning of Aerial
Relays, 757, 771

International Civil Aviation Organization
(ICAO), 2162–2170, 2174

International law, 2866, 2871, 2872, 2874,
2875

International treaties, 2908

Interoperability, 1726, 1727, 1730, 1732,
1744

Intersection-based, 1554–1561, 1564–1566

Inter-swarm communication, 772

Invasive plants, 2814, 2832

IR camera, 1965

Isolation, 1074, 1085, 1090–1094,
1098–1100, 1102, 1103, 1110,
1114, 1117, 1118

J

Joint Chiefs of Staff, 2508, 2513

Joint Terminal Attack Controllers (JTAC),
2512–2517, 2521, 2522

Joint Unmanned Aircraft System Center of
Excellence, 2510

Just cause, 2987

Just war theory, 2989

K

Kalman filters, 1075, 1080, 1084, 1094,
1107, 1118

Kármán vortex street, 1368, 1369, 1371

K-band, 1902, 1903

Killer robots, 2866, 2871

Kinematics, 243–277

Kinematics and rigid-body dynamics,
1224–1227, 1255

Knoller-Betz effect, 1360

Knowledge process, 904, 905, 909

L

Language, 855, 856, 870–872, 874, 875, 877,
878, 884, 885, 889, 892–893, 908,
912, 913, 924, 926, 927, 930, 931,
944, 946, 947

Last resort, 2987

Latency, 1281, 1284, 1285

Law, 2884, 2886, 2887, 2891–2898,
2901–2908

Law of armed conflict, 2967, 2972–2980,
2989

Level of Repair Analysis (LORA), 2582,
2584, 2592

Liability, 2886, 2887, 2896

Licensed spectrum, 817, 834

Life cycle cost analysis (LCCA), 2582, 2592
Life cycle management, 2601–2634, 2789, 2790, 2806–2809
Lift-to-drag ratio, 176
Light-sport Aircraft (LSA), 2581, 2582, 2592
Limited communication, 2021–2046
Linear and angular momentum, 244, 245, 276
Linearized equations of motion, 538
Linear quadratic Gaussian control (LQG), 566, 568, 573
Linear quadratic regulator (LQR), 560–566, 573
Line of sight (LOS), 1918, 1920, 1928, 1934–1939, 1942
Link budget, 732–735, 743, 744
Load transportation, 955, 961–963, 969
Localization, 433–459
Logistics, 2601–2634
Long-term prediction, 1059, 1062
Long-wave infrared (LWIR), 2725–2727, 2734, 2735, 2737–2740, 2742
Low-Altitude Long-Endurance (LALE), 2732
Low-Altitude, Short-Endurance (LASE), 2724, 2732
LP relaxation, 1606

M

Machine autonomy, 2868, 2870–2872, 2875
Maintenance, 2605, 2608–2612, 2615–2627, 2630–2634
Manned aviation, 2118, 2132
Manufacturing techniques, 110, 111, 114
Maritime operations, 2787–2809
Market-based approaches, 1477–1480
Markov decision processes, 1448, 1449, 1456–1464, 1480–1483
Matching errors, 2656, 2663
Matching process, 2652, 2656
MAV. *See* Micro aerial vehicle (MAV)
Maximum takeoff weight (MTOW)
 medium altitude, 85, 86
 very high altitude, 86
 very low altitude, 85
Mean Shift algorithm, 2652
Mechanical bird, 58, 59
Medium-Altitude, Long-Endurance (MALE), 2724, 2731–2732, 2747
Mental workload, 2476–2478, 2495, 2497
Micro aerial vehicle (MAV), 329–345, 1305–1327, 1385–1412
Microbat, 1373–1375

Midair collision, 1819, 1823–1828, 1831, 1834, 1837
Midair collision risk, 2219, 2222
Mid-wave Infrared (MWIR), 2725, 2727, 2734, 2735, 2737, 2740, 2742
Military ethics, 2986, 2989
Military profession, 2989
MIMO, 1434–1437
Mission analysis, 168, 171–173
Mission Essential Competency (MEC), 2519–2522
Mission planning, 1509–1544
Mixed integer linear programming (MILP), 1512, 1519–1521, 1525, 1526, 1543, 1544
Model-based reasoning, 1028–1030
Model formulation, 1219, 1220, 1222–1239, 1248, 1255
Modeling, 1257–1270
Model inputs, 1224, 1252
Model inversion based control, 590–594, 598
Model predictive control, 579, 588–590, 606
Model reference adaptive control, 579, 594–598, 622
Modulation, 1418–1427, 1437
Monitor formula, 919–924
Monitor formula progression, 923–924
Monopulse, 1906, 1907
Moral autonomy, 2871
Motor dynamics, 1223, 1243
Moving targets, 2023, 2024, 2028, 2032, 2033, 2037, 2046
Mulrine, A., 2507
Multi-agent scenario, 1549, 1551, 1572
Multi-mission/multi-purpose air system, 111
Multi-mission UAV, 110
Multi-operator control, 2420–2421
Multiple UAV management, 2410
Multi-UAV systems, 956, 957, 963, 966, 969
Multi-UAV teams, 2049–2075
Multi-vehicle SLAM, 403, 423, 427, 428

N

Nano Hummingbird, 1379–1381
NASA GTM (Generic Transport Model), 1157–1180
National Airspace, 2177–2197, 2508, 2511
National Airspace System (NAS), 2171
National Transportation Safety Board (NTSB), 2162, 2173
Navigation, 347–379
Nearest insertion, 1512, 1521–1522, 1544
Near infrared, 2481

- Negentropy, 1280
 Negotiation-based, 1549, 1551–1554,
 1566–1568, 1574
 Network, 781–809
 architectures, 1624–1626, 1630, 1646
 resilience, 1990
 Networked robots, 2082
 Networked UAVs, 2022–2024, 2034, 2046
 Networking, 721
 Neuroimaging, 2479, 2492, 2495
 Node, 856, 858, 889–895, 898–900, 913–915,
 917, 918, 921, 931, 943, 947
 interface, 890, 892
 parameter, 890, 891
 Non-convex supply-demand formulation, 1605
 Non-Gaussian, 462, 463, 469, 487
 Nonlinear control, 322–323, 579, 605
 Nonlinear program, 1581
 Nonlinear simulation, 282
 Nonsegregated airspace, 1955–1979
 Notice to Airmen (NOTAM), 2182, 2183,
 2192–2195, 2197
- O**
- Observe, Orient, Decide, Act (OODA)
 loop, 1274
 Oceanic airspace, 2166, 2168
 Operational and logistic scenario, 2567–2571
 Operation rules, 2127–2130
 Operator performance, 2497
 Operator workload, 980
 Optical brain imaging, 2475–2498
 Optimal control, 2055–2058
 Optimal path planning, 1579
 Organic Persistent Intelligence Surveillance
 and Reconnaissance (OPISR), 1283,
 1287–1292, 1294–1297
 Organisational approval, 2288, 2289
- P**
- Parameter identification, 281, 291, 295,
 1218–1220, 1227, 1239–1252, 1255
 Parametric study, 112, 113
 Parity space, 1184–1190, 1193, 1200, 1202,
 1203, 1207, 1210
 Partial loss fault in control effectiveness, 1174
 Partially observable Markov decision processes
 (POMDP), 1776–1783, 1785–1787,
 1789, 1805
 Particle filter, 1057, 1059, 1062, 1068
 Particle filtering, 2652, 2653, 2661
 Path following, 1681, 1682
 Path planning, 1509–1544, 1547–1576
 Payload, 110–112, 115, 119
 budgets, 146–149
 design, 143–162
 management, 2407, 2414–2416, 2420
 subsystems, 1, 46, 145, 155, 157, 162
 Payload and Perception Control Language
 (PPCL), 855, 856, 871, 877, 878,
 884–888, 890, 937, 944, 946
 Performance-Based Logistics, 2613–2615,
 2618
 Personal remote sensing, 2339–2358
 Peterson, N., 2508
 PFC. *See* Primary Flight Computer (PFC)
 Phase comparison monopulse (PCM), 1906,
 1907
 Physics-based models, 1055
 Pilot, 2137–2156
 Pilot-in-command, 2166, 2167
 Pilot performance, 2496
 PITVANT, 2525–2560
 Planning architectures, 1468–1477
 Platform server, 855, 856, 858, 871, 877, 878,
 884–885, 890, 893, 894, 900, 904,
 909, 946, 947
 Platoon matching method, 2654, 2662
 Policy, 904–906
 Policy-based radio, 823
 Polynomial-time, 1605, 1609, 1615, 1616
 POMDP. *See* Partially observable Markov
 decision processes (POMDP)
 Portable Collision Avoidance System
 (PCAS), 1897
 Porto University College of Engineering, 2526
 Portuguese Air Force (PoAF), 2526, 2527,
 2540, 2541, 2550
 Position, 348, 350–359, 361, 370, 374,
 376, 377
 Potential field, 1549, 1550, 1561–1564,
 1572, 1574
 Power line inspection, 394–397
 PPC. *See* Primary Perception
 Computer (PPC)
 PPCL. *See* Payload and Perception Control
 Language (PPCL)
 Predator MQ-1, 2509, 2521
 Predator Research Integrated Networked
 Combat Environment (PRINCE),
 2517, 2519
 Primary Flight Computer (PFC), 860, 861, 863,
 864, 877–880, 884, 885, 900, 909
 Primary Perception Computer (PPC), 9, 861,
 864, 877–879, 884, 885, 909
 Probabilistic, 462, 470–472

- Procedure for Air Navigation Services (PANS),
 2162, 2168, 2170
 Professional ethics, 2986, 2989
 Prognostics, 1053–1069
 Prognostics and health management
 (PHM), 1002, 1007, 1027–1050,
 1054, 1069
 Prognostics horizon, 1069
 Prohibitions on use, 2900–2901
 Propeller efficiency, 176, 178
 Proportionality, 2973–2974, 2987
 Propulsion, 261, 262, 264, 268–270, 2294,
 2295, 2298–2306, 2330, 2333
 Pseudocontrol hedging, 625, 663
 Psychological stress, 2970–2972,
 2979, 2980
 Public Law 109-364, John Warner National
 Defense Authorization Act for
 Fiscal Year 2007, 2507
 Publish/Subscribe, 808, 809
 Pulsed, 1899–1902
 Pulse repetition frequency (PRF), 1900
 Pursuit guidance, 370, 372
- Q**
- Qball-X4 quadrotor UAV, 1132
 Quadro-rotor MAV, 1345–1346
 Quadrotor, 307–327, 1258–1267,
 1269, 1270
 Quadrotor dynamics, 307–327
 Qualitative modeling, 1033
 Quaternions, 534–535, 537, 548,
 555–559
- R**
- RADAR. *See* RAdio Detection And Ranging
 (RADAR)
 Radar, 1965–1972, 1974–1978
 Radio, 717–721, 724, 726, 728–732, 734, 737,
 739–745
 Radio control, 61
 Radio-controlled model aircraft, 2163
 RAdio Detection And Ranging (RADAR),
 1895–1952
 RAMS, 2589, 2595
 Rapidly-exploring random tree (RRT), 1656,
 1657, 1674
 Reactive layer, 853–855, 858, 870–893, 947
 Real-time data, 2645, 2651, 2659
 Reaper MQ-9, 2509, 2517
 Regional Air Navigation (RAN),
 2165, 2168
 Regional Supplementary Procedure (SUPPS),
 2162, 2168, 2169
 Regulation, 2235, 2238–2242, 2261, 2268,
 2270, 2271
 Regulation study, 2173
 Regulatory, 721, 724, 729, 730, 741, 748,
 2293–2334
 Regulatory-based Causal Factor
 Framework (RCFF), 1820–1825,
 1837
 Regulatory risk controls, 1824, 1825,
 1828–1837
 Reliability, 2201, 2202, 2215, 2220–2223,
 2225, 2226, 2872–2874
 Remotely Operated Video Enhanced Receiver
 (ROVER), 2515
 Remotely piloted aircraft (RPA), 2169,
 2505–2522
 Remote sensing, 387, 389, 394–397
 Required communications performance (RCP),
 717, 722, 725–729, 732, 743,
 746, 747
 Residuals, 1079–1085, 1087, 1088, 1090,
 1092–1094, 1103–1112, 1114,
 1117, 1118
 Resolution Advisory (RA), 1860, 1862, 1863,
 1866, 1868–1870
 Rigid body, 244–246, 250–252, 256, 260–268,
 272, 274, 277
 Risk, 2229–2271
 management, 2229–2271
 modeling, 2199–2227
 perception, 2236, 2237
 Road detection, 2648–2649, 2661
 Road following, 2648, 2657, 2663
 Road geometry information, 2656
 Road-network search, 1511, 1512, 1517–1531,
 1543
 Road traffic data, 2648, 2660
 Robot morality, 2874
 Robust and adaptive control, 675–708
 Robust communication, 463, 467, 469
 Role-based Connectivity Management,
 776–777
 Roll-pitch coupling, 1219, 1248
 Rommel, E., 1274
 ROS, 856–859, 885, 888–890, 908–909
 Rotary wing, 1331, 1332, 1344–1346
 Rotorcraft flight control and guidance, 616
 Rotor flapping dynamics, 1220, 1229–1231,
 1234, 1236–1237, 1245, 1246,
 1248, 1249
 Rotor thrust and torque, 1229, 1234
 Royal Air Force (RAF), 2511, 2512

- RRT. *See* Rapidly-exploring random tree (RRT)
- RTCA, 2170
- RUL, 1059, 1069
- Rules of the Air, 2164–2166, 2168, 2170
- S**
- Safety, 719, 721, 722, 727–729, 731, 732, 735, 740, 741, 743, 747, 2229–2271, 2869, 2872–2874, 2967, 2968, 2970, 2979
analysis, 1878–1880, 1882, 1886–1893
baseline, 1817–1838
- Safety Management System (SMS), 1819, 1822, 1823
- Satellite positioning, 433–459
- Scientific method, 2171, 2172
- See and avoid* (SAA), 1818, 1819, 1823–1838, 1844–1846, 1852, 1875, 1876, 1878, 1882, 1887–1890, 1892, 1893, 2165–2167, 2169
- Self-separation, 2365, 2369–2371, 2373–2376
- Sense-and-avoid (SAA), 1817–1838, 1841–1854, 1857–1859, 1861, 1895–1952, 2294, 2306–2312, 2361–2380
- Sense-reasoning Gap, 856, 903
- Sensor, 2365, 2370, 2372–2374, 2378
networks, 463, 465, 469
suite, 1957, 1960, 1961, 1963–1966, 1970, 1974
- Separation provision, 1956–1959, 1961, 1962, 1964, 1966–1969, 1972–1974, 1976–1978
- Service, 851, 852, 855, 856, 863, 870, 933–942
- Service discovery, 794–798, 800–802, 806, 808, 809
- Shrinking error-bound, 2657
- Signal to noise ratio (SNR), 1909, 1922
- Simmons, N., 2509
- Simulated annealing, 1590, 1591
- Simulation, 1731, 1741–1742
- Simultaneous localization and mapping (SLAM), 401–430
- Simultaneous state and parameter estimation, 1169, 1172, 1179
- Situational awareness, 1312, 1318–1320, 1326
- Skew-configuration, 1183–1210
- Sleeper weeds, 2830–2832
- Slung payload, 1257–1270
- Small UAS, 2527, 2553
- Small UAV, 143–162
- Small Unmanned Aircraft System (SUAS), 2340, 2341, 2343–2347, 2351, 2356, 2357
- Society of Automotive Engineers (SAE), 2170
- Solar aircraft, 174
- Source, 874, 904, 947
- Spectrum, 720–725, 729–731, 742, 745–747
management, 815–817
policies, 816, 836, 837
- Speech-based control, 2446, 2458, 2461, 2462
- Spherical air vehicle, 1347–1348
- Split-cycle, 334
- Stability axes, 531, 532
- Stability derivatives, 341–343
- Standardization, 2792, 2803, 2805, 2807
- Standards, 2160, 2162, 2164, 2166, 2168–2171, 2174, 2175, 2361–2380
- Standards or Recommended Practices (SARPs), 2165, 2168
- Starting vortex, 1364–1366, 1368
- State, 858, 864, 865, 870–875, 877–885, 888, 896, 898, 906, 907, 909, 914, 915, 917–924
aircraft, 2163
space, 1279, 1280
- State-space models, 115
- Statistical errors, 2655, 2663
- Statistical profile, 2644, 2649–2650, 2661
- Stepped frequency continuous wave (SFCW), 1901
- Stochastic stability, 1704, 1711, 1712
- Straight-line path following, 1678, 1679, 1698
- Strategic, 2728, 2730–2737, 2749, 2751, 2752
- Stream, 856, 858, 861, 864, 865, 903–910, 924, 934, 937, 947
- Stream generator, 904, 905
- Structural health monitoring (SHM), 111, 127–130
- Supervisory control, 2406–2410, 2413, 2420–2421, 2444–2446, 2448, 2452–2454, 2456–2458, 2462
- Support vector machine (SVM), 2825, 2829
- Sustainment, 2602–2606, 2609, 2610, 2612, 2616–2632
- Swarm(s), 955, 957, 958, 961, 965–967, 972
interaction, 1989
manageability, 1988, 1989, 2017
- Swashplate, 1219–1221, 1229, 1238
- Symptom generation, 1044, 1045
- System architecture, 1727–1728, 1740–1741
- System design, 2425–2438
- System dynamics, 1264, 1268
- System reconfiguration, 111, 127, 128
- System safety analysis, 2173

Systems design, 2965–2981
 Systems engineering, 2791
 System trade-offs, 144

T

Tactical, 2728, 2730–2733, 2737–2740, 2744, 2749, 2751, 2752
 Tactile displays, 2456–2457
 Tail-sitter MAV, 1346–1348
 TAL. *See* Temporal Action Logic (TAL)
 TALplanner, 858, 911, 913–918, 941, 942, 947
 Target assignment, 1603, 1605, 1614, 1615
 Target classification, 2651
 Targeted killings, 2866, 2867
 Target prosecution, 2032–2033, 2035
 Target tracking, 1776, 1777, 1796, 1806
 Task, 840, 843, 852, 854–856, 858, 859, 863, 865, 870, 883–885, 889–893, 910, 913–918, 925, 926, 930, 931, 946, 947
 allocation, 1601–1616, 1620, 1622, 1625, 1647–1649, 2022, 2023, 2080, 2081
 analysis, 2425–2438
 assignment, 1547–1576
 Task-based control, 990
 Task Specification Tree (TST), 854, 855, 858, 859, 871, 889–893, 925, 928, 931, 937, 943, 946
 TCAS. *See* Traffic Alert and Collision Avoidance System (TCAS)
 (European) Technical Standard Order ((E)TSO), 2284–2286, 2288
 Technological trends, 2790, 2802–2806, 2808, 2809
 Technology survey, 2293–2334
 Tele-operated, 1278, 1283, 1284, 1286
 Teleoperation, 1277, 1283, 1309
 Temporal Action Logic (TAL), 858, 892, 893, 911–915, 917, 919, 923, 925, 927, 928, 947
 Temporal density, 2653
 Temporal interface, 2452
 Territorial airspace, 2168, 2169, 2174
 Test bench experiment, 1239–1246, 1249, 1255
 Test flights, 1873–1893
 Testing, 2897, 2899–2900, 2905
 Throughput, 722–724, 743
 Thrust-specific fuel consumption (TSFC), 120
 Tip path plane (TTP), 1219, 1227, 1231, 1237, 1245, 1247

Title 14 Code of Federal Regulations (14 CFR), 1820–1824, 1828, 1831, 1834, 1835, 1837
 Topic, 856, 859, 888, 922
 Tornado, 2178, 2180–2182, 2195
 Track swap, 1776, 1777, 1782, 1787, 1801–1806
 Traffic advisory (TA), 1862, 1863, 1866
 Traffic Alert and Collision Avoidance System (TCAS), 1857–1871, 1897, 1965–1972, 1974–1978, 2369
 Traffic detection, 1888–1892
 Traffic flow, 2646, 2653, 2662
 Traffic flow data, 2656, 2663
 Traffic management, 2646, 2648, 2649
 Traffic monitoring, 2643–2665
 Traffic parameters, 2649, 2654, 2656, 2663
 Training, 2476–2478, 2485, 2492, 2497, 2498, 2505–2522
 Training Rehearsal System (TRS), 2515–2517, 2522
 Trajectory tracking control, 617, 640
 Transition, 853, 855, 856, 870–872, 874–876, 880, 882, 889, 901, 902, 911, 946
 Truth maintenance system (TMS), 1289, 1291
 TST. *see* Task Specification Tree (TST)
 Two-stage Kalman filter (TSKF), 1131, 1134, 1139–1145, 1168, 1169, 1175, 1177–1179

U

UASTL RMAX, 857–862, 864, 868, 877–880, 882, 884–886, 888, 890, 893, 896, 898, 900, 902, 909, 924, 934, 937, 943, 945–947
 Uncertainty management, 1056
 Unlicensed spectrum, 834
 Unmanned, 2506–2511
 Unmanned Aerial System (UAS). *See* Unmanned aircraft system (UAS)
 Unmanned aerial vehicle (UAV), 143–162, 715–748, 978, 988, 989, 1123–1153, 1257–1270, 2022–2029, 2031–2046, 2079–2107, 2601–2634
 channel model, 764
 configuration, 244, 256
 design specifications, 112–122

- networks, 1989, 1991, 2008
 - path planning, 1776–1778, 1805
 - payload, 386, 391
 - system, 2602–2607, 2611–2613, 2617–2620, 2623, 2624, 2626, 2627, 2631, 2632
 - Unmanned aircraft, 618, 2757
 - Unmanned Aircraft Program Office (UAPO), 2180, 2183, 2191, 2192
 - Unmanned aircraft system (UAS), 143–162, 1874–1888, 1891–1893, 2340–2353, 2355–2358, 2723–2752
 - Circular 328, 2170
 - concepts of operation, 2527, 2543–2550
 - development, 2554
 - for environmental monitoring, 2668
 - logistics, 2587–2592
 - operations, 2527
 - operator certification, 2342, 2350, 2352
 - regulations, 2340
 - service platform, 754–756
 - training, 2525–2560
 - transportability, 2566, 2583–2587, 2592
 - Unmanned helicopter, 639
 - Unmanned systems, 2881, 2882
 - Unpredictability, 2887–2888
 - Unscented Kalman Filter (UKF), 1130, 1131, 1134, 1136–1139, 1159, 1163–1169, 1174, 1178
 - U.S. Air Force (USAF), 2507, 2508, 2510, 2511, 2514
- V**
- Validation of design, 2280, 2289, 2290
 - Variable-pitch coaxial helicopter, 1220–1223, 1225, 1227, 1234–1239, 1255
 - Vector field guidance model, 1679
 - Velocity, 348, 351–359, 361, 370, 372
 - Vision-based sense and avoid, 392
 - Visual syntax, 872, 873, 877
 - Volcanic environments, 2667–2690
 - VORTEX2, 783, 791, 803–808, 2178, 2180–2183, 2186, 2188, 2191, 2193–2195, 2197
 - Vulnerability, 1705, 1706, 1717, 2210–2214, 2221, 2224, 2225
- W**
- Wagner function, 1365
 - War crimes, 2970, 2975–2979
 - Wavelet transform, 1185, 1186, 1190–1193, 1200, 1203, 1207, 1208, 1210
 - Weed classification, 2825
 - Weed detection, 2814, 2815, 2824–2825, 2832
 - Weed management, 2815, 2821, 2823, 2825, 2832
 - Weed surveillance, 2816–2818, 2824–2831
 - Weis-Fogh effect, 1371, 1373, 1378
 - White spaces, 816, 817, 823, 826, 834, 835, 840
 - Wildfire, 2724, 2727–2736, 2738, 2741–2752
 - Wind compensation, 1778, 1787–1789, 1805
 - Wireless, 1423, 1432, 1434
 - Wireless channel models, 1620, 1621, 1623, 1626–1631, 1641, 1649
 - Woody weeds, 2815, 2823–2830, 2832
 - Work analysis, 2429–2432, 2435
 - Workload, 1312, 1314, 1316–1321, 1324, 1326, 2137, 2142–2144, 2149–2151, 2153
 - Workstation, 2153–2155