Control, Dynamics and Robotics

SIRCA-2526

Jorge Martins

Instituto Superior Técnico
Department of Mechanical Engineering
Center of Intelligent Systems – idMEC
Human Robotics Group
JorgeMartins@tecnico.ulisboa.pt

Outline

Control

- A brief **History**
- Foundations: Concept of **Signals** and **Systems**
- Representation: **Input-Output** and **State-Space**
- Concepts of **Feedback** and **Feedforward**
- **Stability** vs **Performance**: a trade-off
- Control Techniques: **PID, Optimal, Nonlinear**

Robotics

- Kinematics, Dynamics and Control
- Machine Intelligence

A brief History

Main sources:

IEEE

IEEE Control Systems Magazine, 16(3) 1996 – Dedicated to the History of Control

IFAC
INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

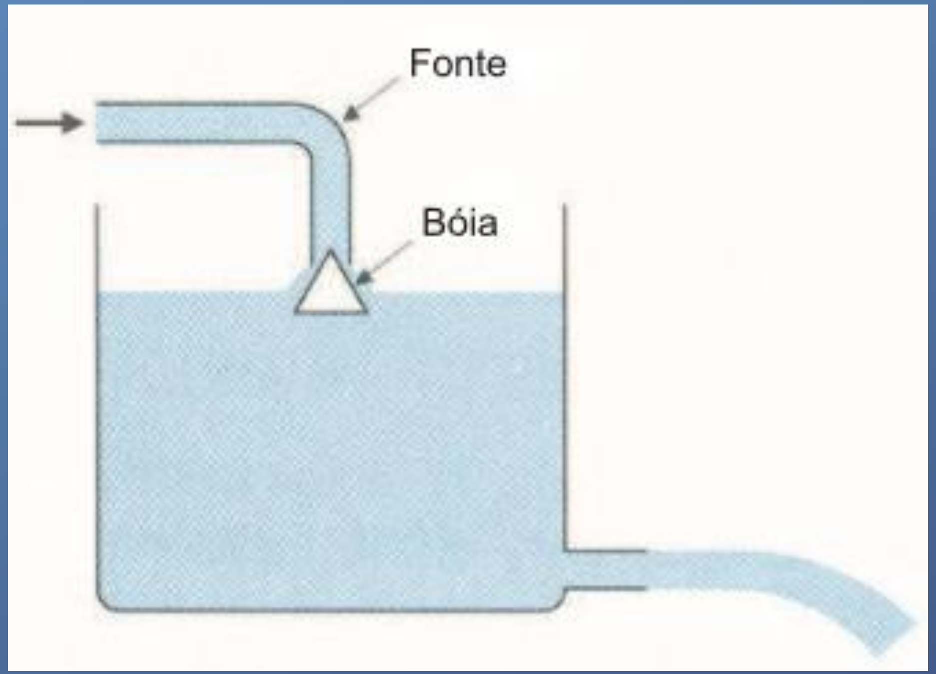
IFAC Control Resources

The IFAC Control Resources (ICR) Publications – History of Control

A brief History

From an engineering perspective

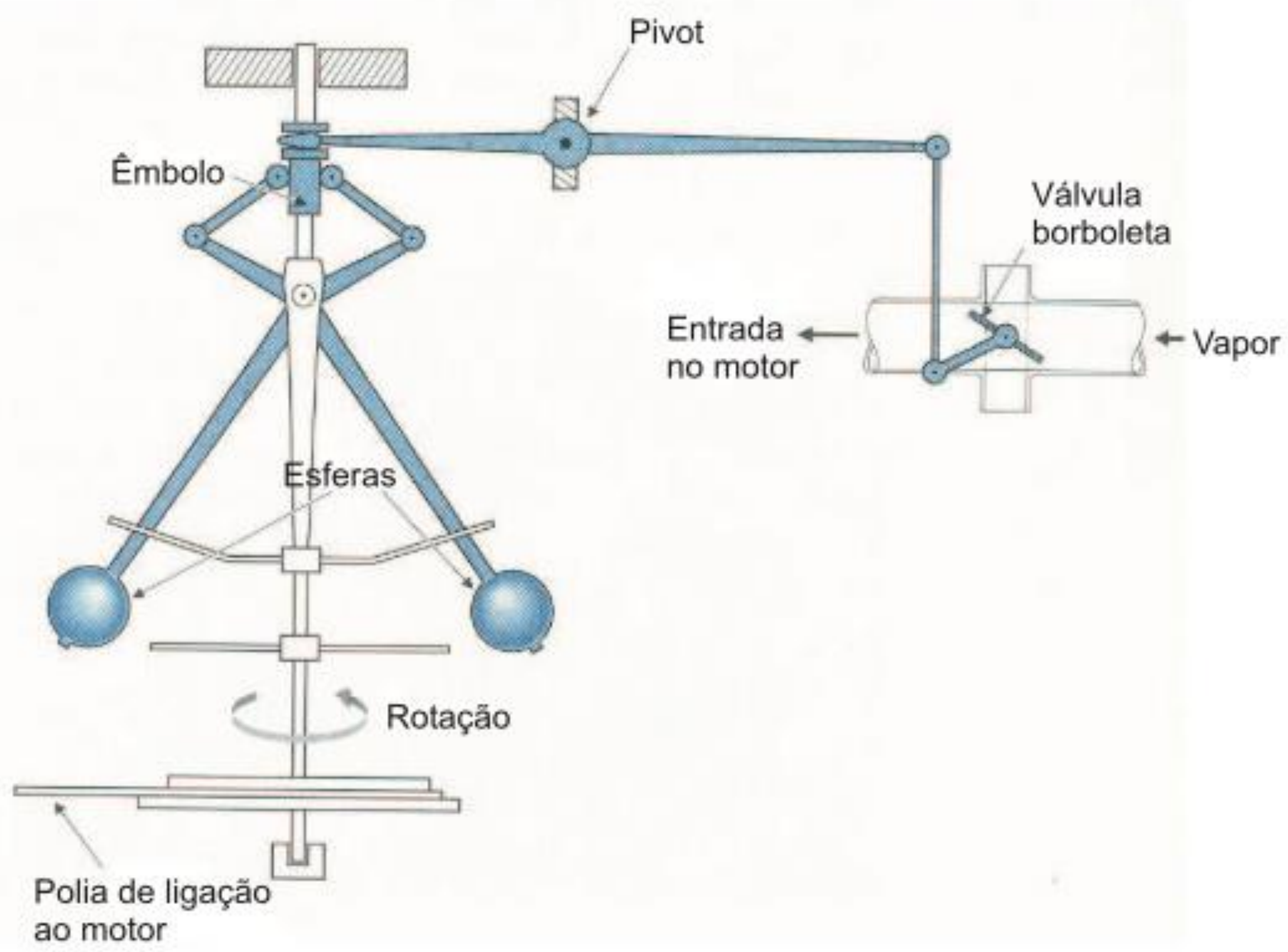
- Probably the oldest application of control:
*Maintaining the liquid level (in a **wine barrel**), independently of the consumption*

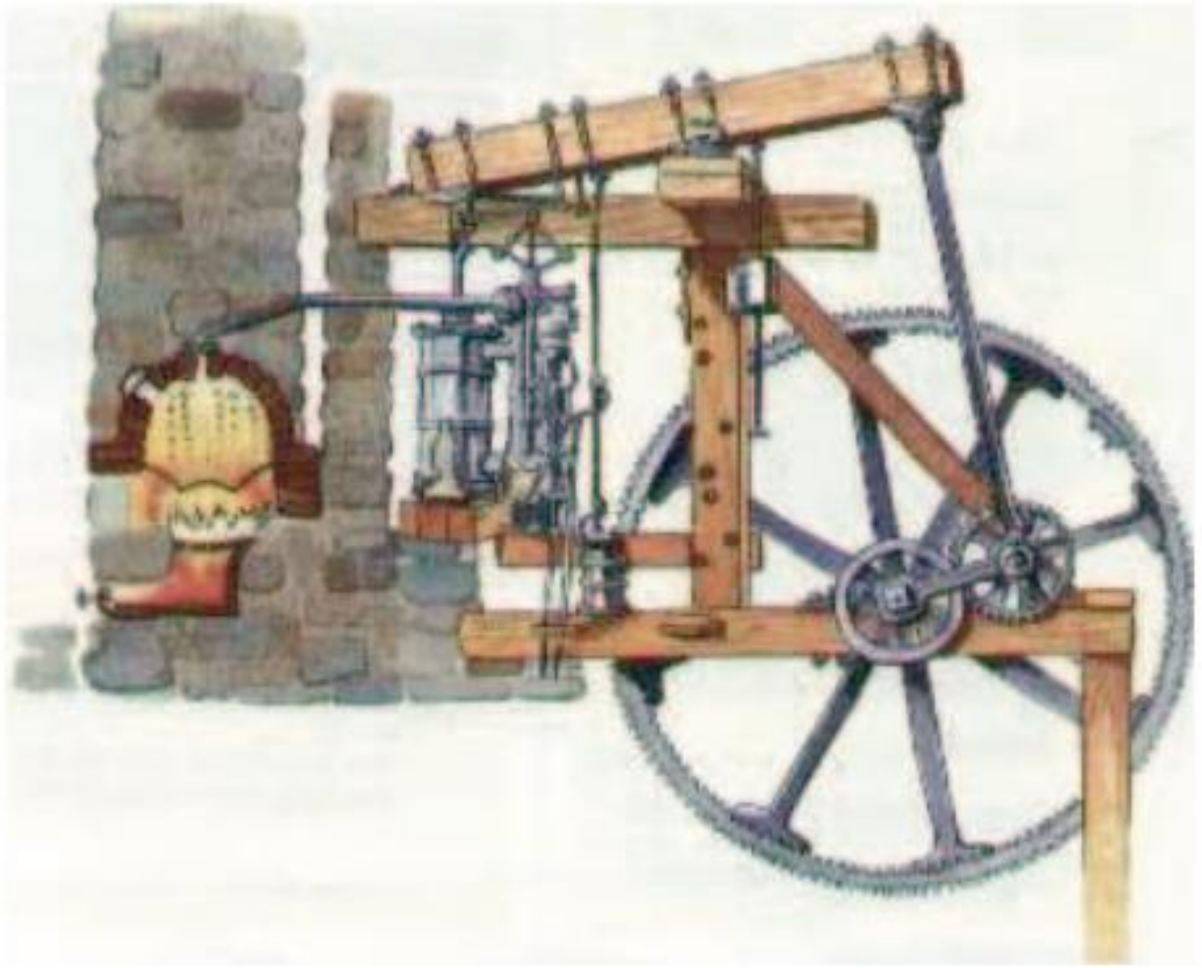


- The Water Clock – **Ktesibios, 270 A.C.**

A brief History
From an engineering perspective

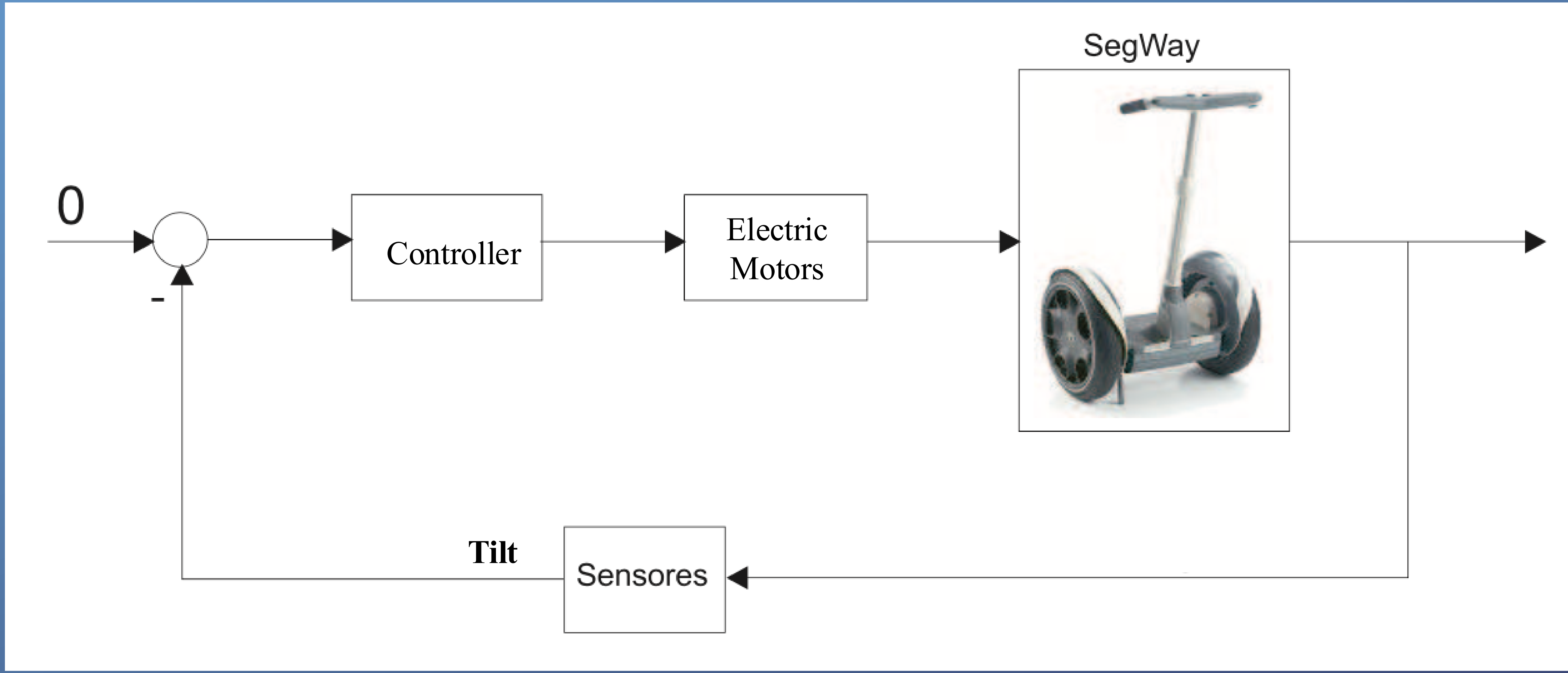
- ***James Watt, in 1788**, developed a system to regulate rotational speed: **Fly-Ball governor**, or Watt's regulator*





A brief History

From an engineering perspective



A brief History
From an engineering perspective

- ***J. Maxwell, 1868***: *"On Governors" – First systematic mathematical analysis of the stability of the fly-ball governor: roots of the characteristic polynomial must have negative real parts*

- ***E.J. Routh, 1877***: *Routh-Hurwitz polynomial stability criterion*

- ***H. Nyquist, 1932***: *Nyquist Stability Criterion – Frequency response*

- ***Callender et. al, 1936***: *Concept of the classical PID controller (Proportinal-Integral-Derivative)*

- ***H. Bode, 1945***: *Based on Nyquist, developed methodologies for the design of amplification systems through feedback – Bell Labs*

A brief History

From an engineering perspective

- ***W. Evans, 1948***: *Root Locus Method - navigation and control of aeronautical systems.*

- ***N. Wiener, 1948:*** *Cybernetics: or the Control and Communication in the Animal and the Machine*

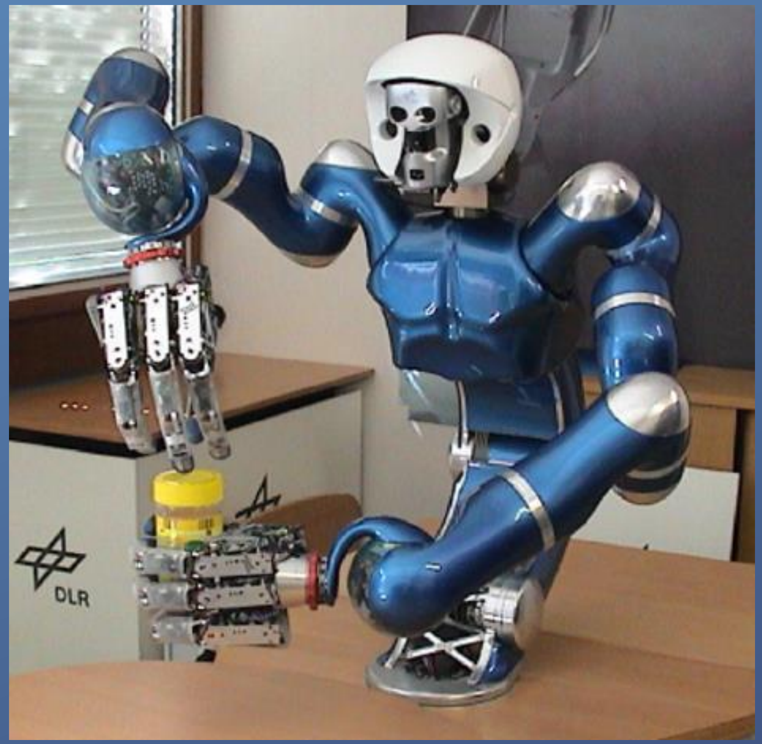
- ***R. Kalman, 1950-60***: *Shift from classical control based on transfer function representation to modern control based on state-space representation. The space exploration era and the digital era.*

- ***1960-1980***: *Based on State-Space representation: Optimal Control, Robust Control and Predictive Control. Based on transfer function representation: Adaptive Control and Stochastic Control.*

A brief History
From an engineering perspective

- ***1980-2000***: *Development of Non-Linear control systems using Lyapunov stability theory. Artificial neural networks. Fuzzy systems. Impedance Control (PCs in 1983, Matlab in 1984)*

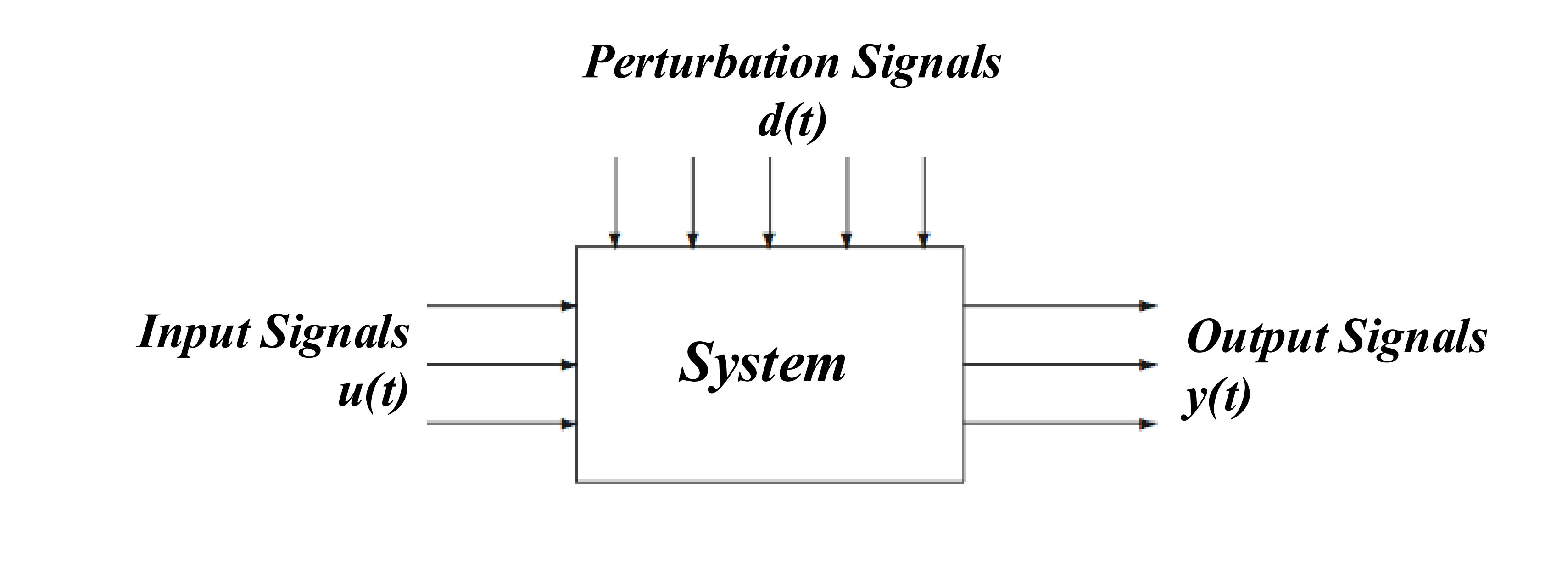
- ***2000-***: *Control of distributed and networked systems, hybrid systems, smart materials, biological systems, learning systems, etc...*



Example - Robotics

Example - Hydroponics

Foundations: Concept of **Signals** and **Systems**



Linear, time invariant, single input - single output system: LTI-SISO

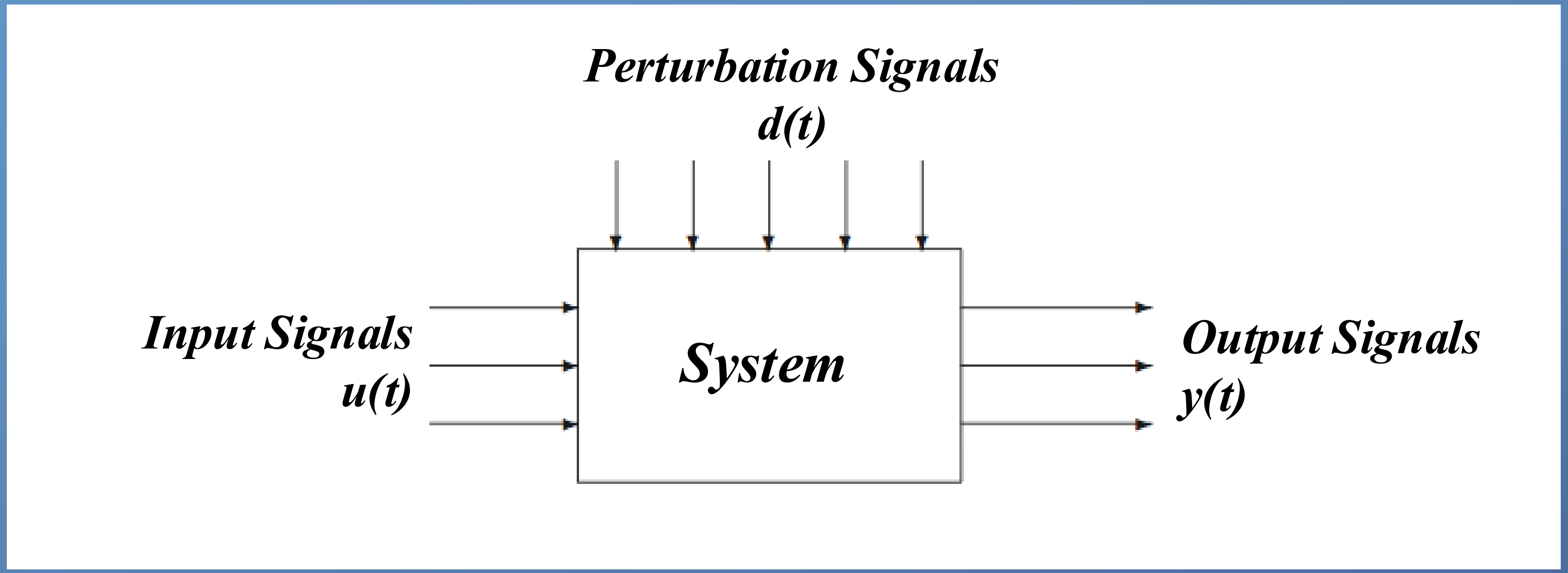
$$\sum_{k=0}^{n} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{m} b_k \frac{d^k u(t)}{dt^k} \qquad n \geq m \quad \text{to be causal}$$

Transfer Function:
(Laplace Transform)

$$\frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} = G(s)$$

escarda.tech

Foundations: Concept of **Signals** and **Systems**



Linear, time invariant, multi-input multi-output system: LTI-MIMO

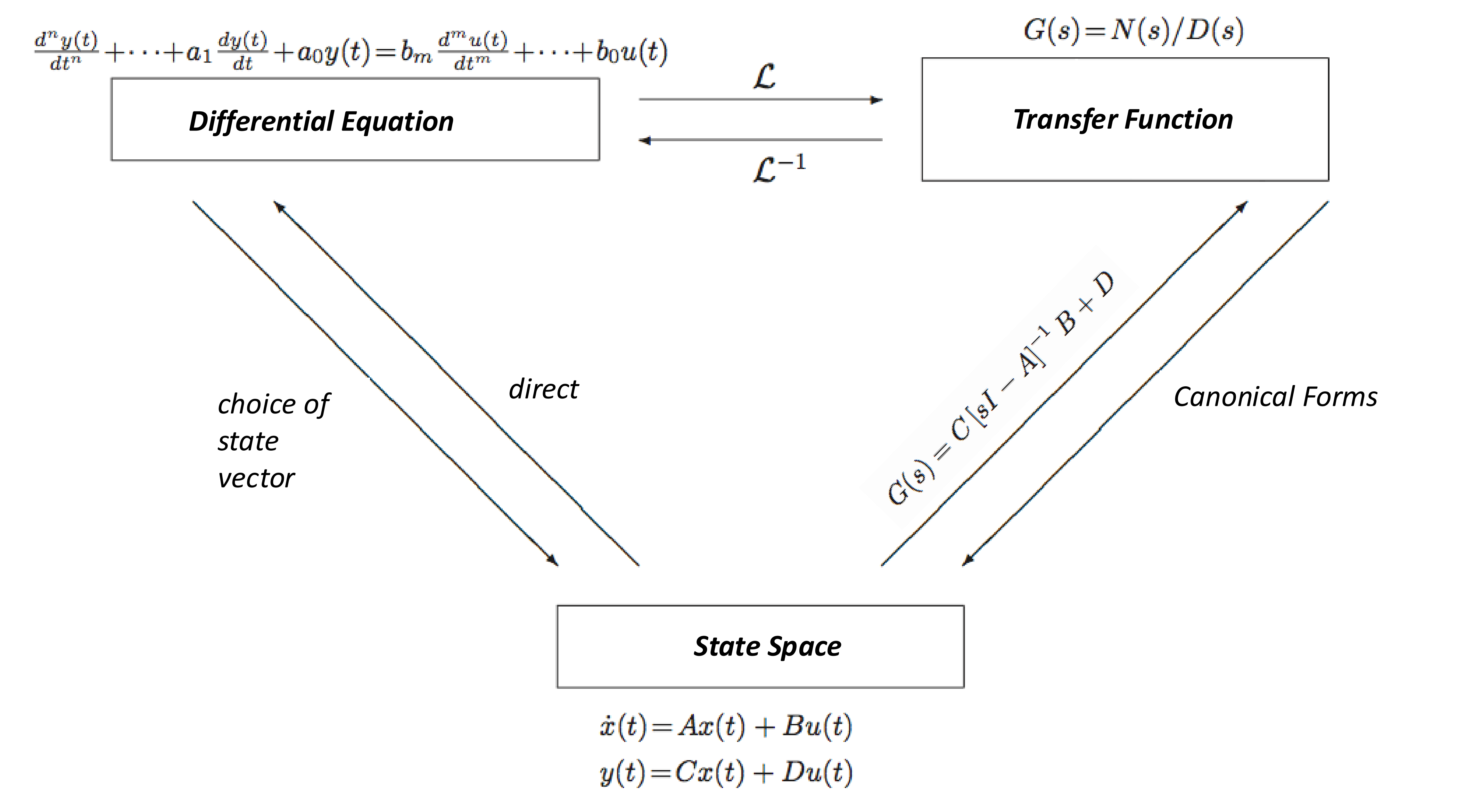
Transfer Matrix:

$$\begin{bmatrix} Y_1(s) \\ \vdots \\ Y_p(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & \cdots & G_{1m}(s) \\ \vdots & & \vdots \\ G_{p1}(s) & \cdots & G_{pm}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ \vdots \\ U_m(s) \end{bmatrix}$$

State Space representation:

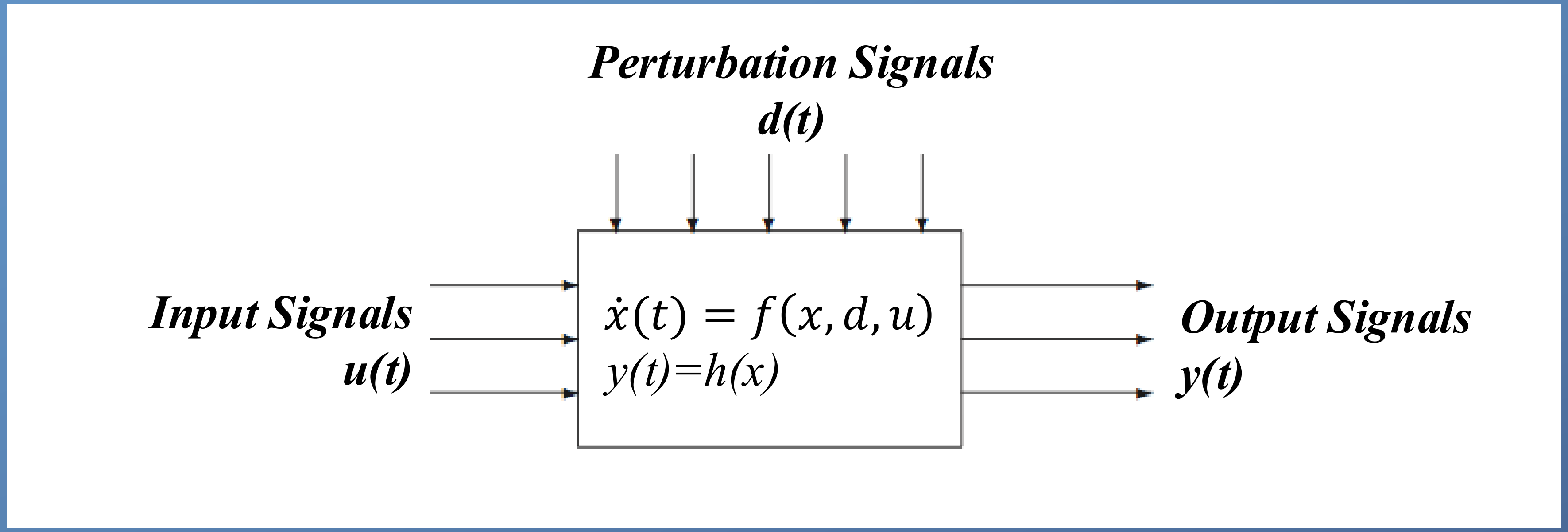
$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \quad , \quad \text{com: } A \in \mathbb{R}^{n\times n}, \text{ e } B \in \mathbb{R}^{n\times m} \\ y(t) &= Cx(t) + Du(t) \quad , \quad \text{com: } C \in \mathbb{R}^{p\times n}, \text{ e } D \in \mathbb{R}^{p\times m} \end{aligned}$$

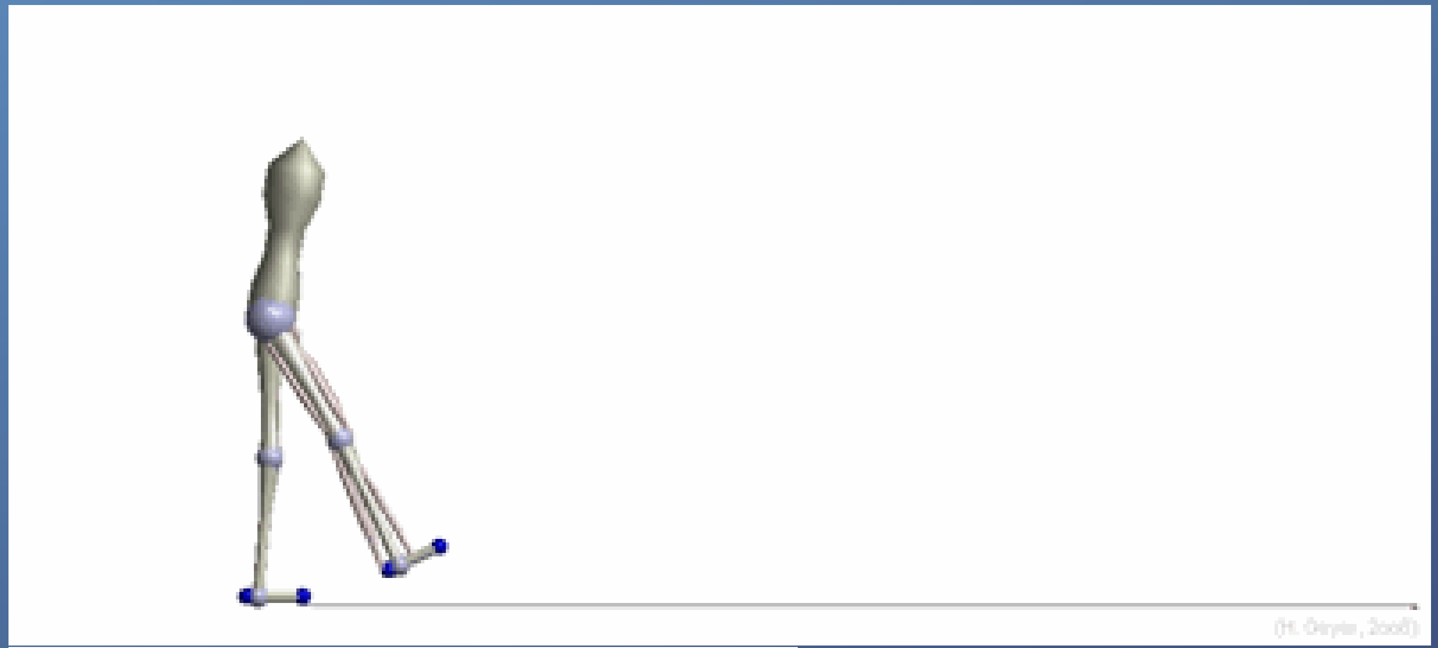
Foundations: Concept of **Signals** and **Systems**



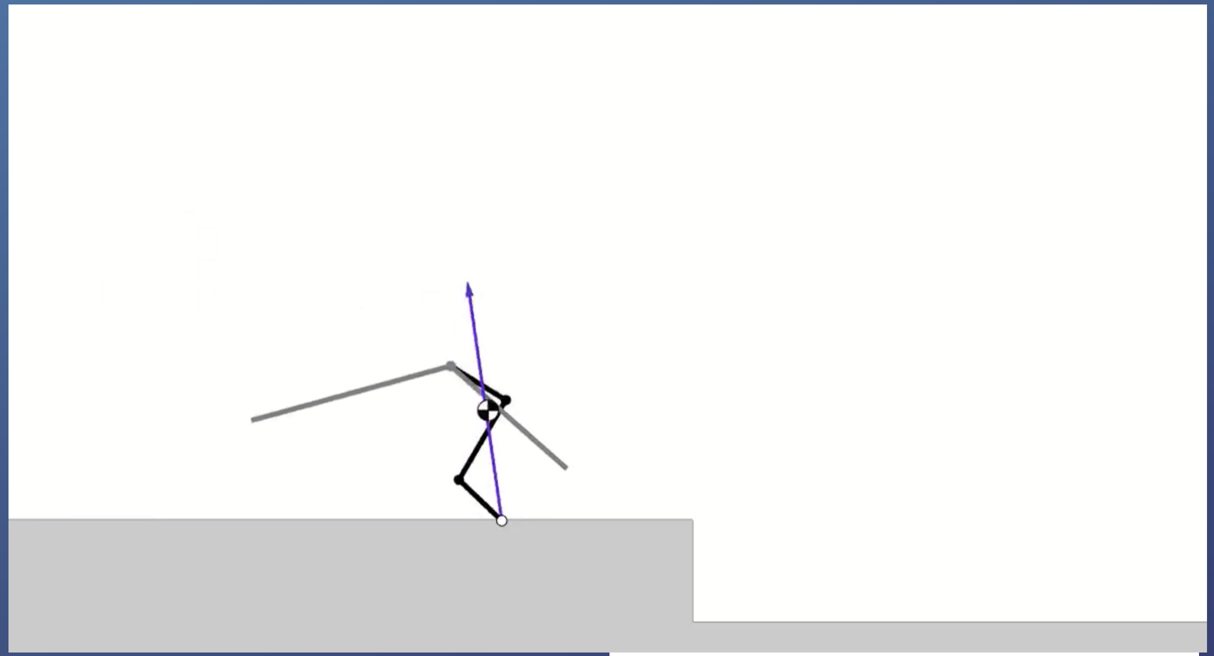
Foundations: Concept of **Signals** and **Systems**

Nonlinear Systems





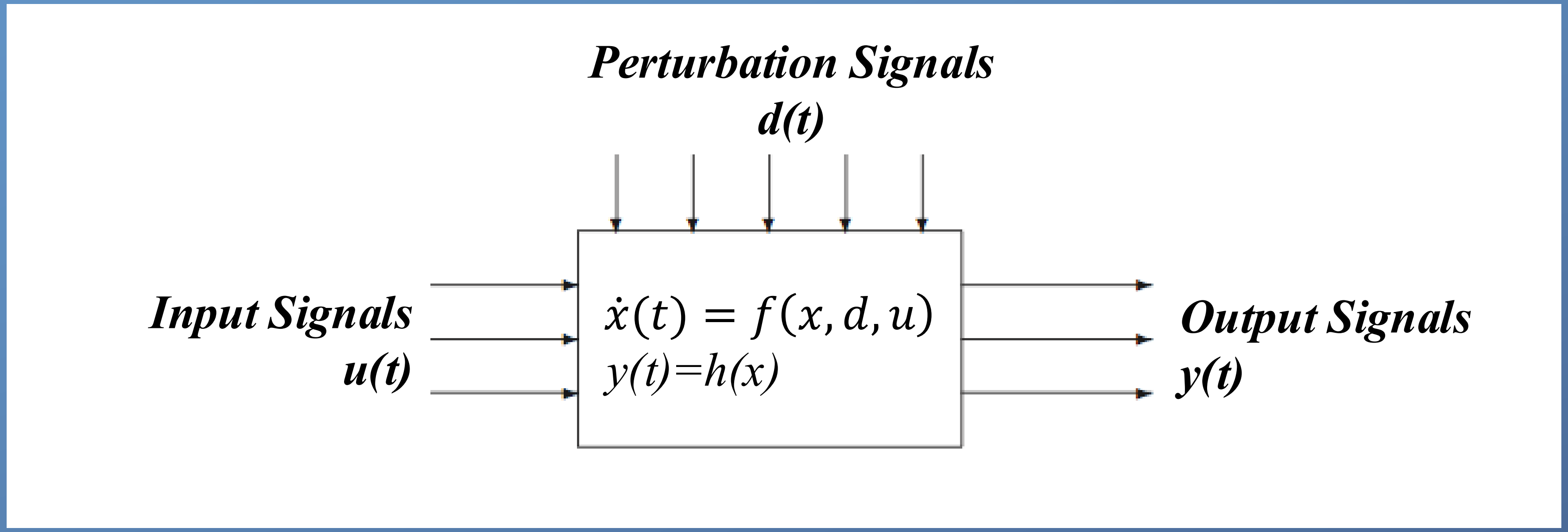
Geyer and Herr, IEEE Trans Neural Syst Rehabil Eng, 2010



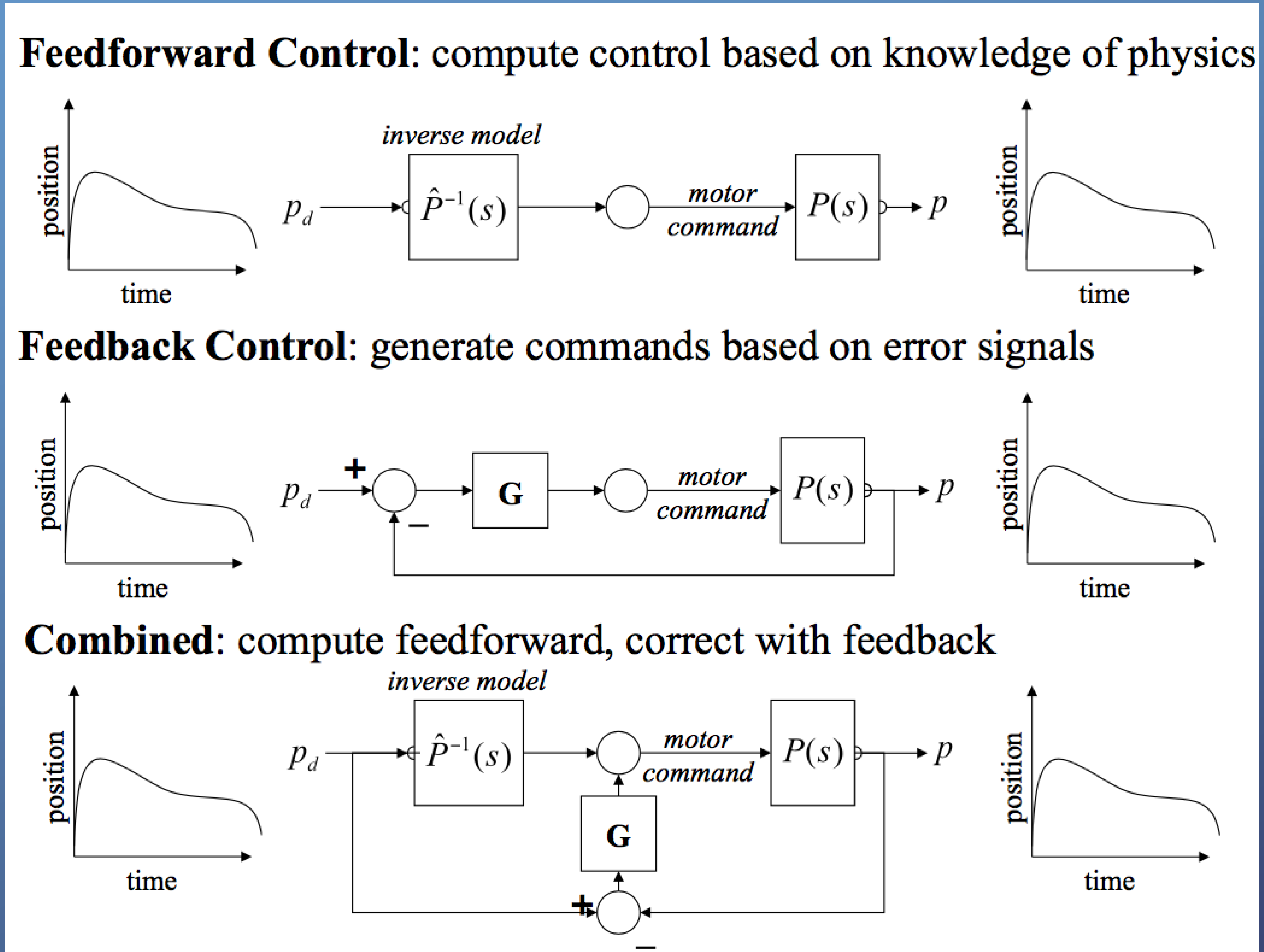
Carvalho and Martins, CONTROLO 2018

Foundations: Concept of **Signals** and **Systems**

Nonlinear Systems

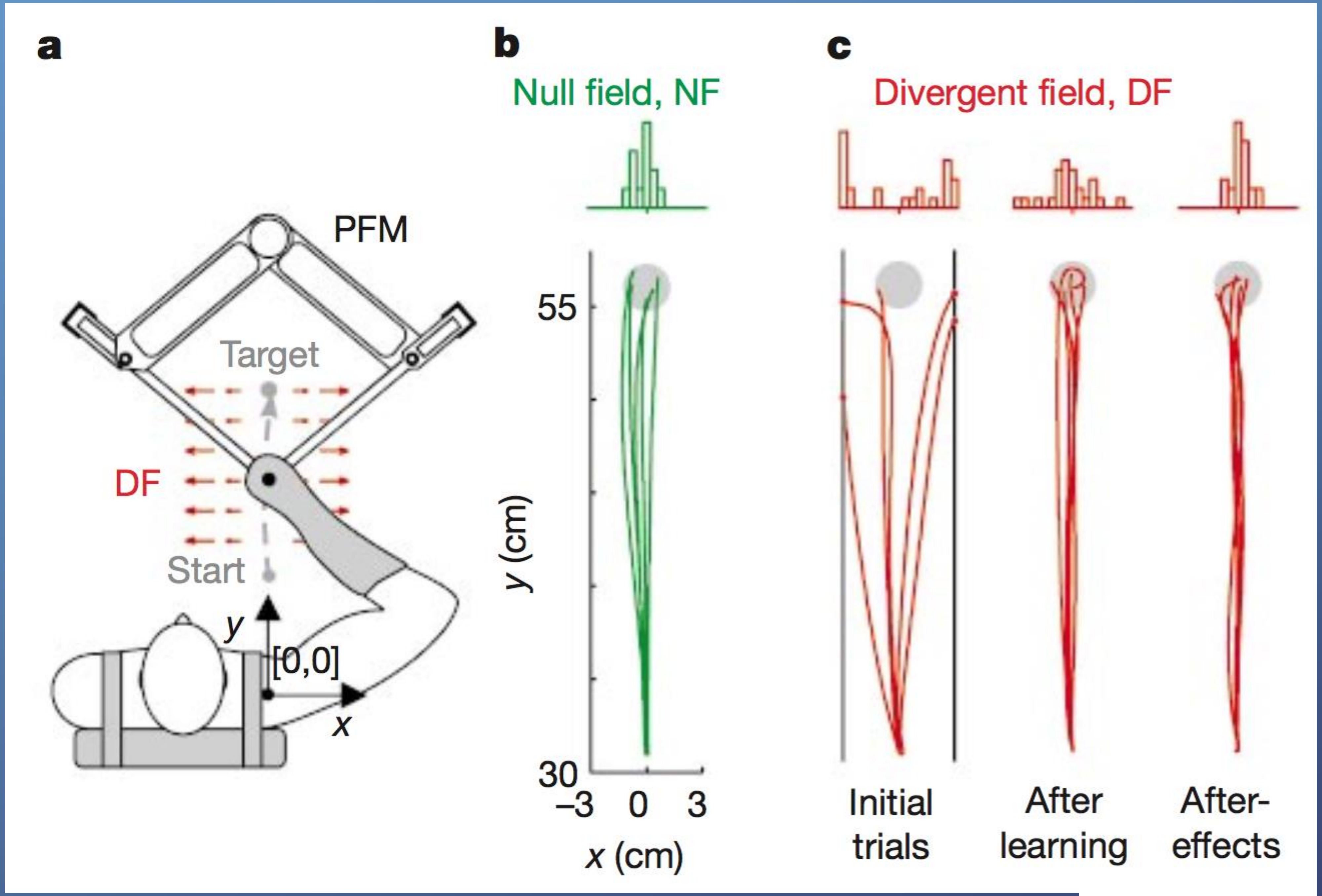


Concepts of **Feedback** and **Feedforward**



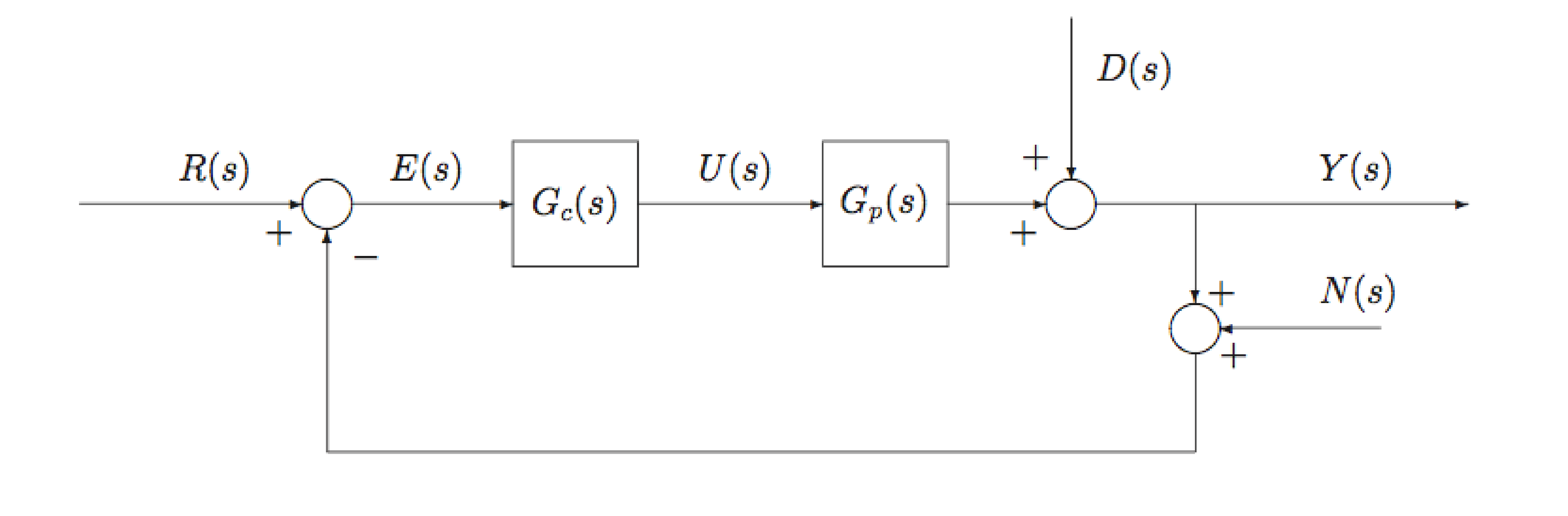
McIntyre, 2011

Concepts of **Feedback** and **Feedforward**



Burdet et al, Nature, 2001

Stability vs **Performance**: a trade-off in feedback control

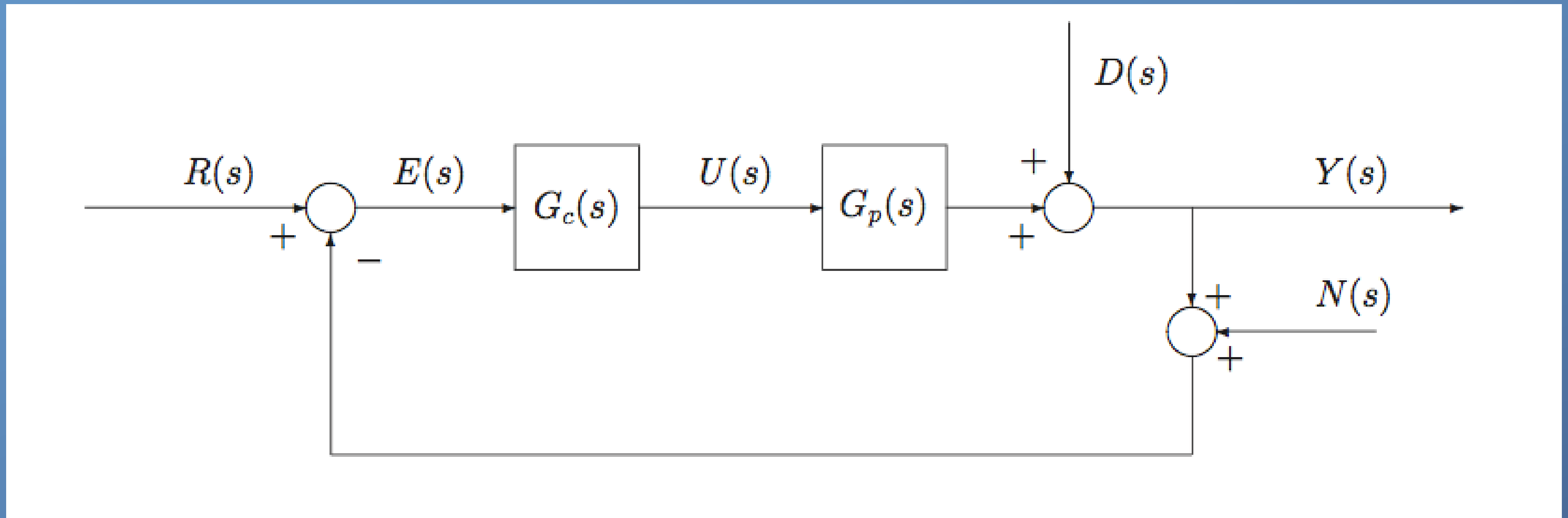


$$Y(s) = \underbrace{\frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}}_{=T(s)} R(s) + \underbrace{\frac{1}{1+G_c(s)G_p(s)}}_{=S(s)} D(s) - \underbrace{\frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}}_{=T(s)} N(s)$$

Complementary Sensitivity Function — Sensitivity Function — Complementary Sensitivity Function

$$T(s) + S(s) = 1$$

Stability vs **Performance**: a trade-off in feedback control

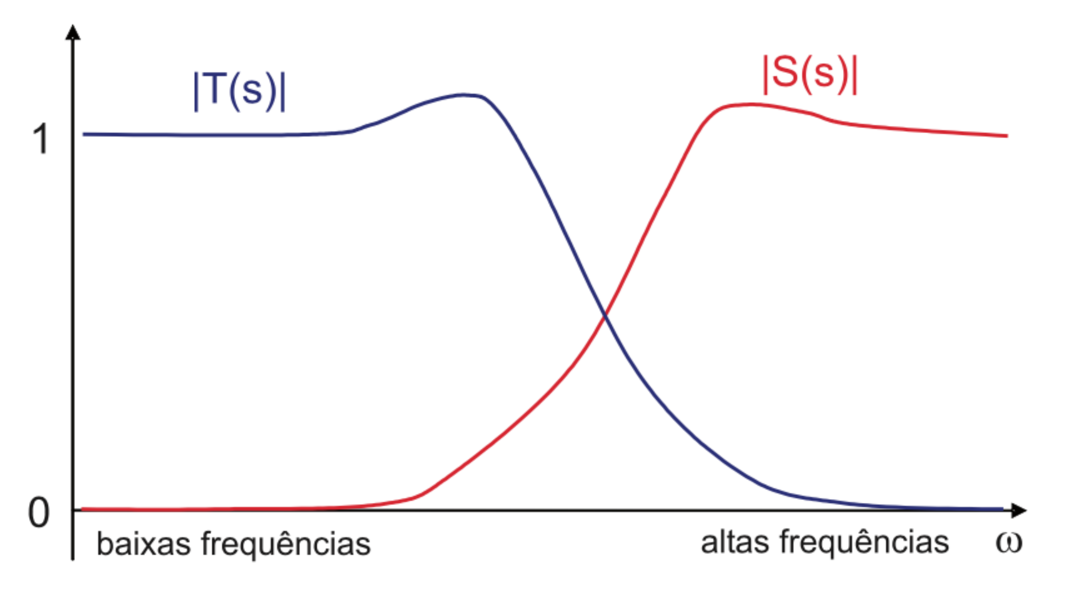


$$Y(s) = \underbrace{\frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}}_{=T(s)} R(s) + \underbrace{\frac{1}{1+G_c(s)G_p(s)}}_{=S(s)} D(s) - \underbrace{\frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}}_{=T(s)} N(s)$$

Good Tracking: $T(s) \approx 1$

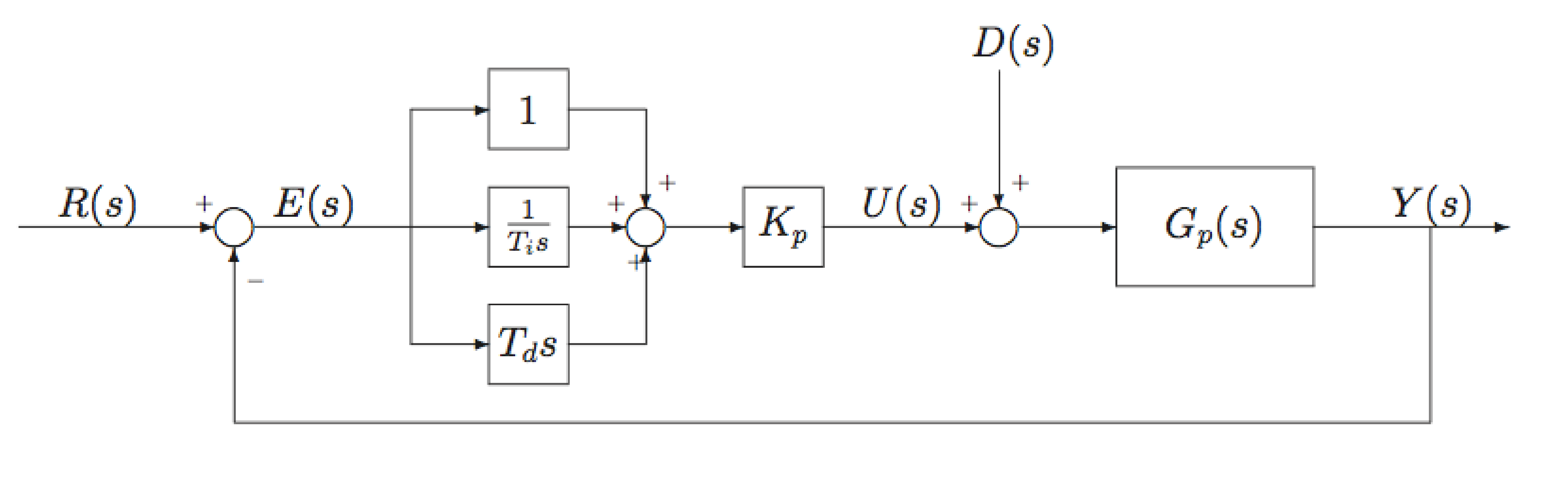
Good Disturbance Rejection: $S(s) \approx 0$

Good Noise Rejection: $T(s) \approx 0$



$$T(s) + S(s) = 1$$

Control Techniques: PID (Proportional, Integral, Derivative)
How to design the gains??



- Ziegler Nichols Methods (*Reaction Curve or Critical Gain*)
- Root-Locus Methods
- Approximations to PD – Lead Compensator or PI – Lag Compensator

Control Techniques: Optimal Control
Linear Quadratic Regulator LQR – discrete time *state feedback*

Minimizing the cost function:

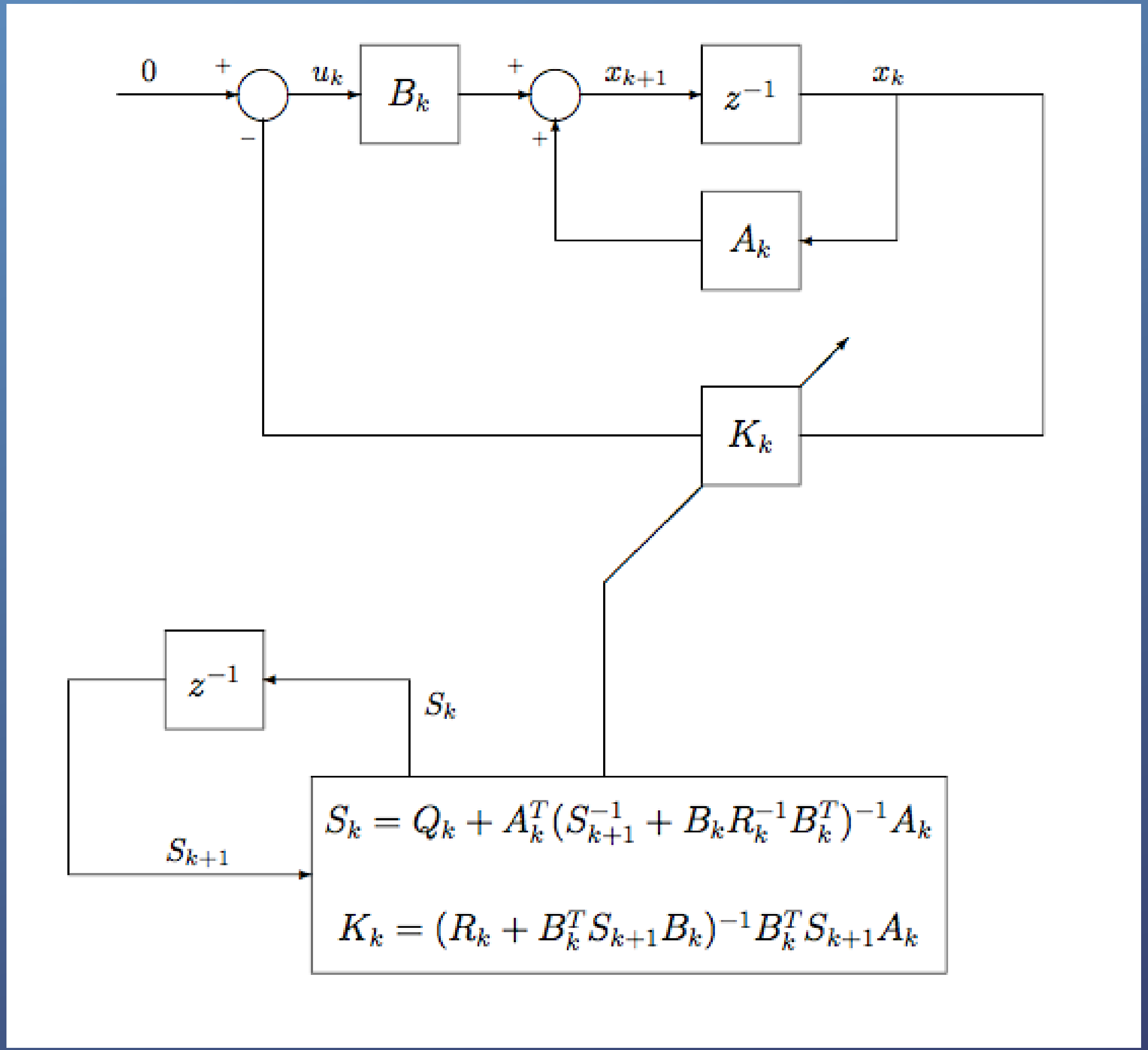
$$J_i = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2} \sum_{k=i}^{N-1} (x_k^T Q_k x_k + u_k^T R_k u_k)$$

Subject to the constraints:

$$x_{k+1} = A_k x_k + B_k u_k$$

Results the optimal control law with *Kalman* gains:

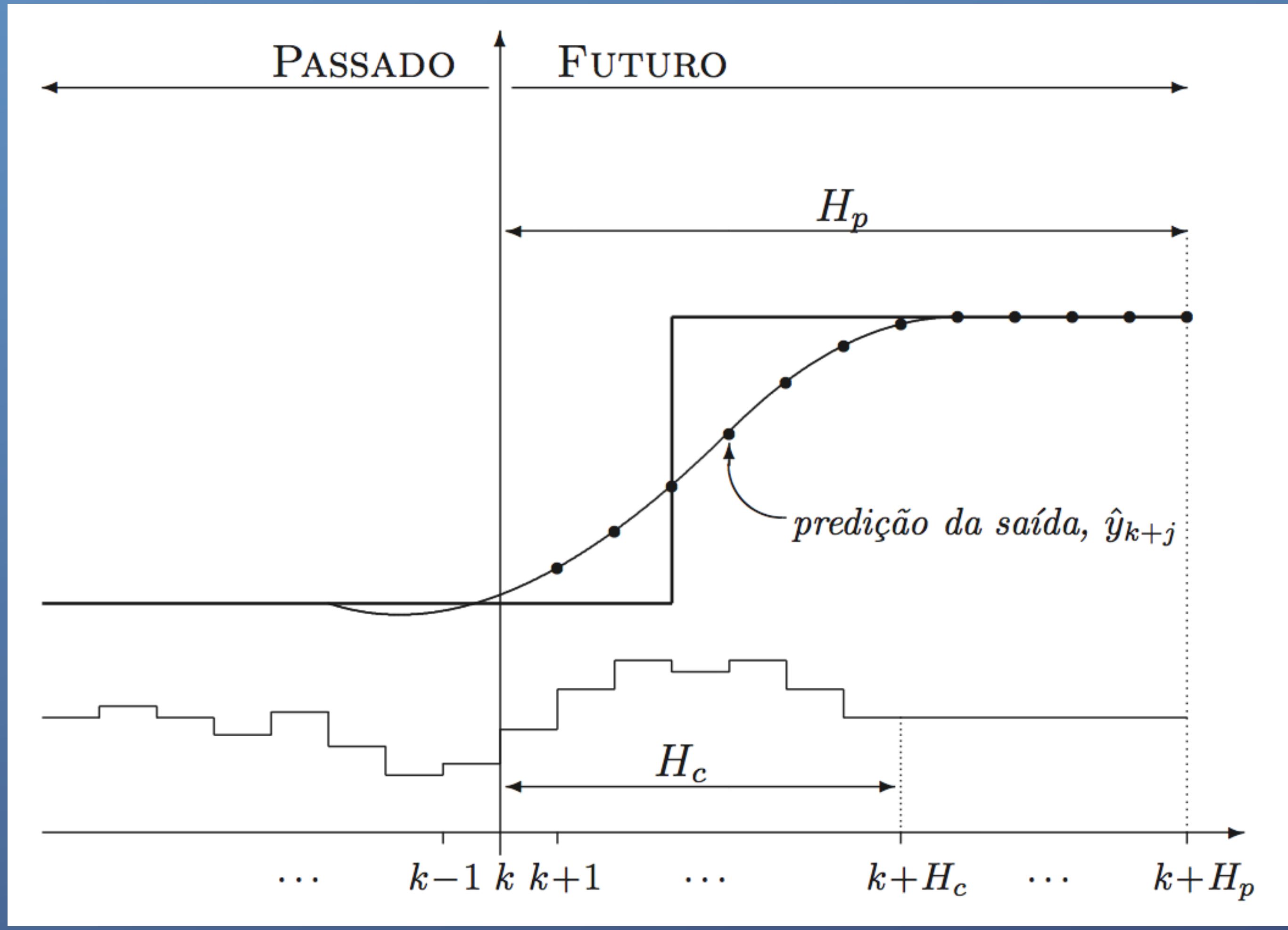
$$u_k^* = -K_k x_k, \quad k < N$$

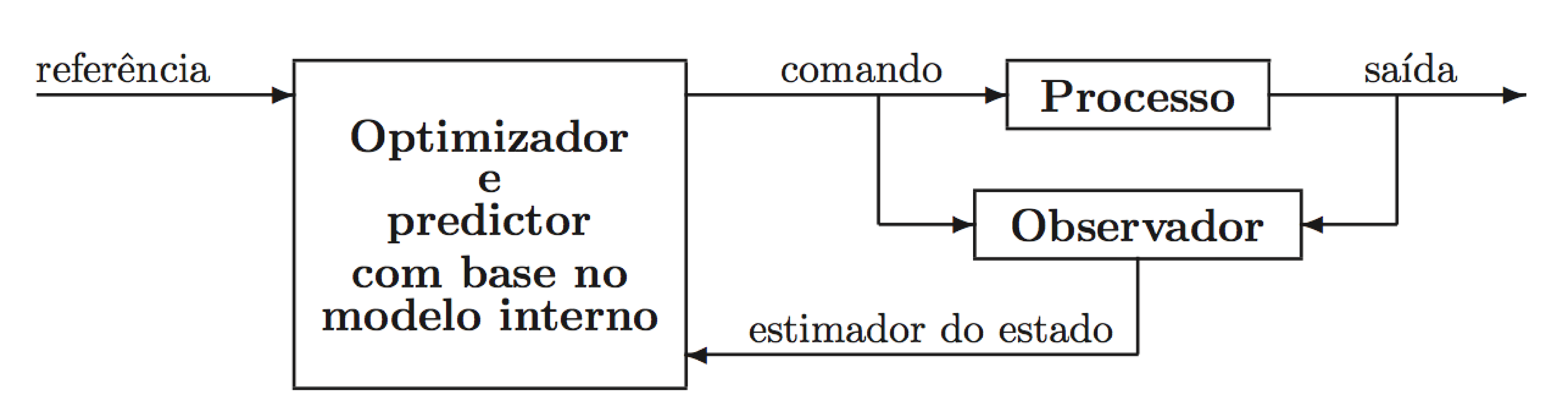


Control Techniques: Model Predictive Control

Minimize the cost function:

$$J(k) = \sum_{j=N_1}^{H_p} \| \hat{y}_{k+j} - r_{k+j} \|^2 + \lambda^2 \sum_{j=1}^{H_c} \| \Delta u_{k+j-1} \|^2$$

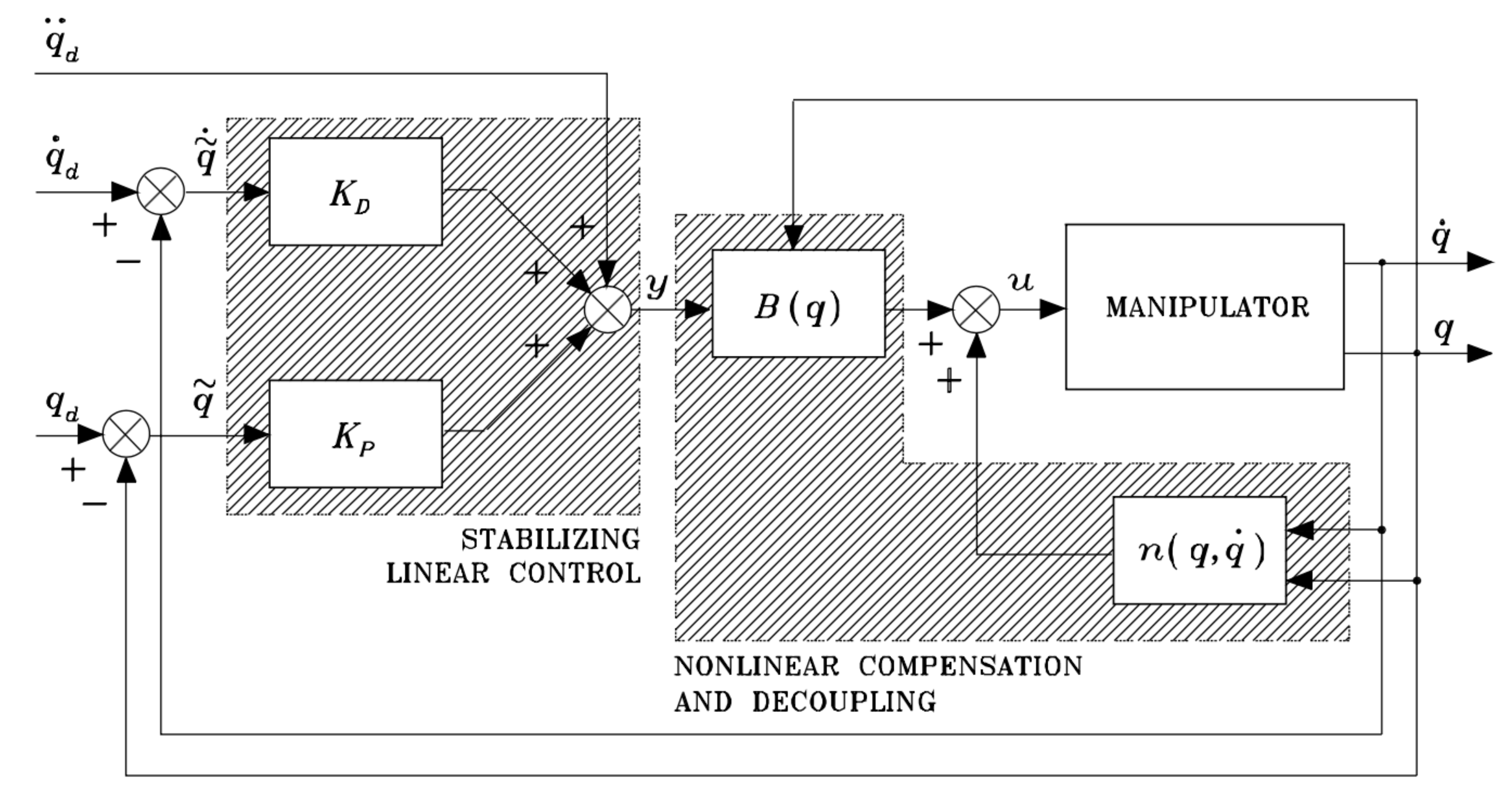




Control Techniques: Nonlinear Position Control – Inverse Dynamics

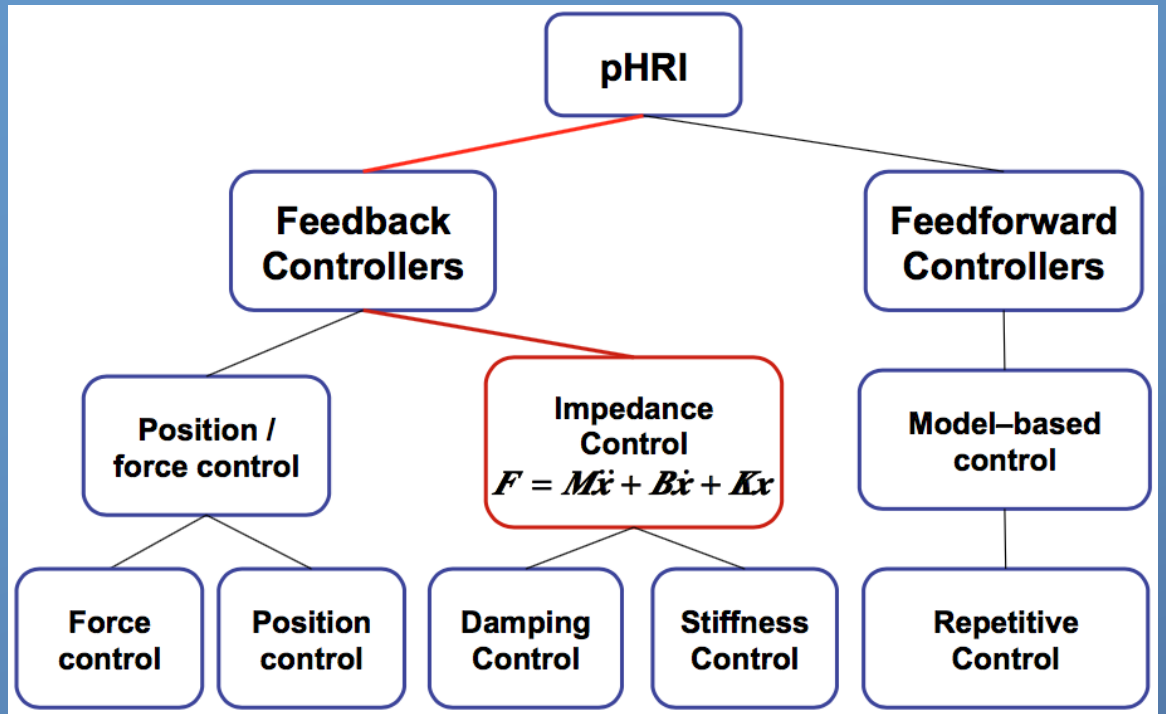
Lyapunov Stability Theory

$$B(q)\ddot{q} + n(q, \dot{q}) = u.$$



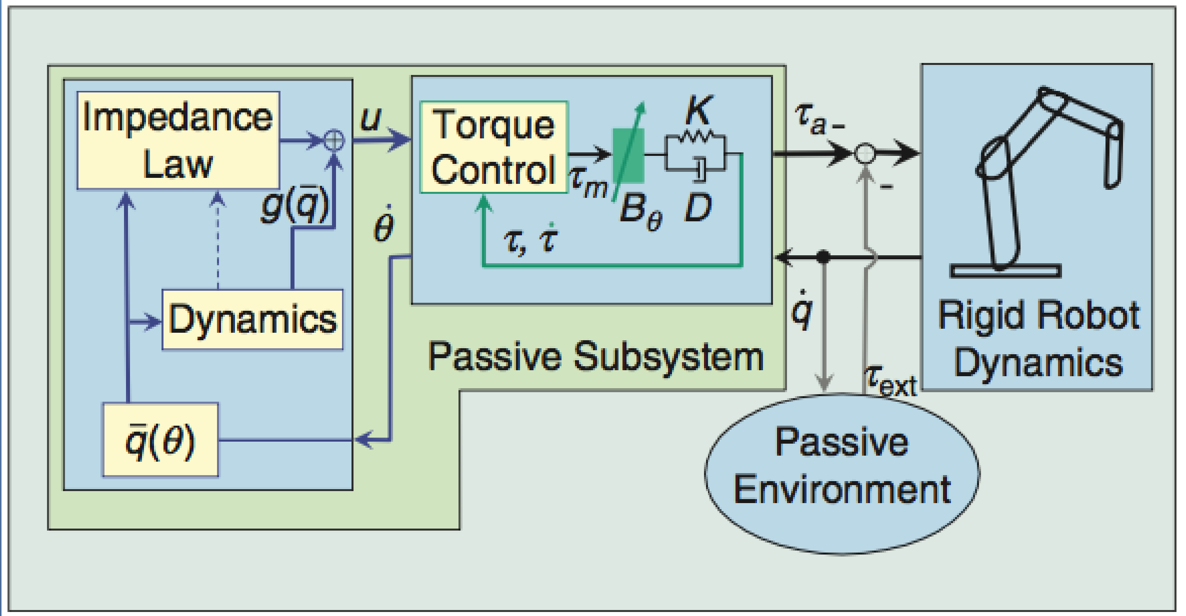
Siciliano et al, 2009

Control Techniques: Nonlinear Interaction Control – Impedance Control



Admittance Causality: $\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + Bs + K}$

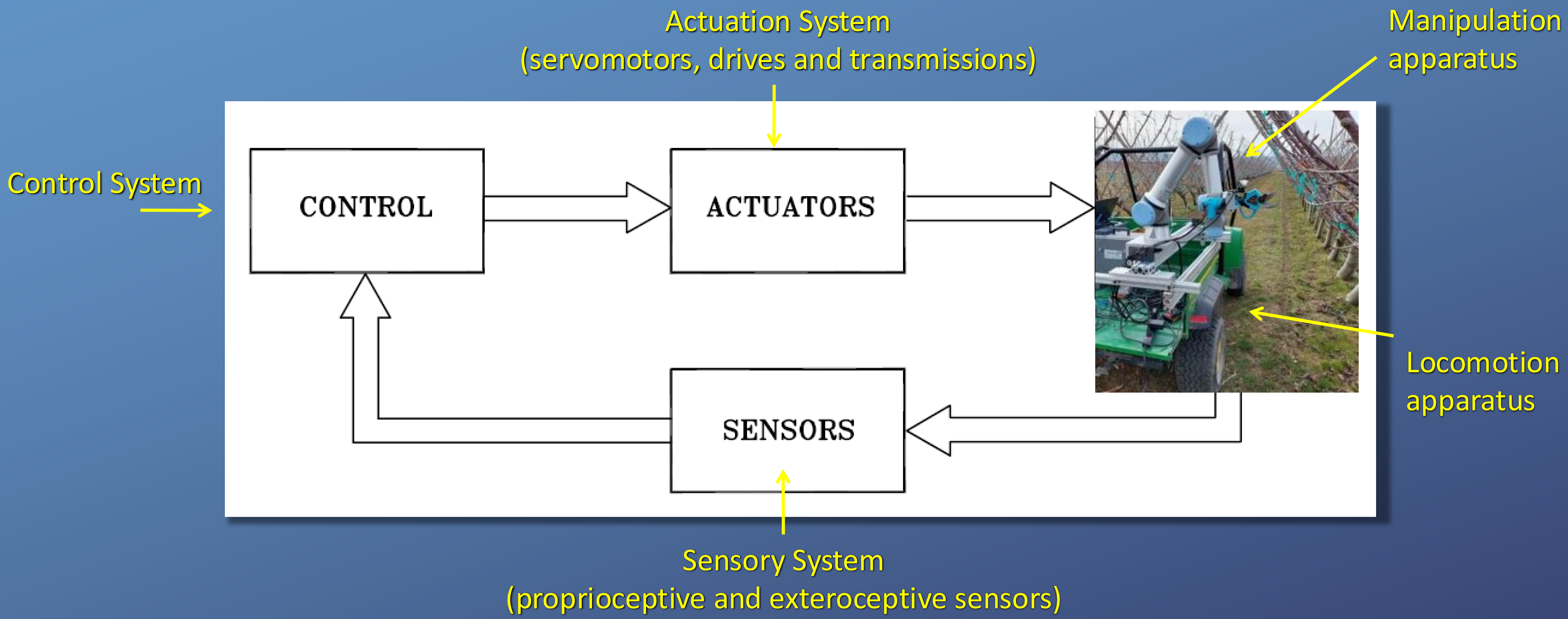
Impedance Causality: $\frac{F(s)}{X(s)} = Ms^2 + Bs + K$



A. Albu-Schaffer et al, IEEE RA Magazine, 2008

Robotics:

In science, robotics is commonly defined as the science studying the intelligent connection between perception and action.



Robotics: interdisciplinary field of mechanics, control, computers and electronics

Robotics:

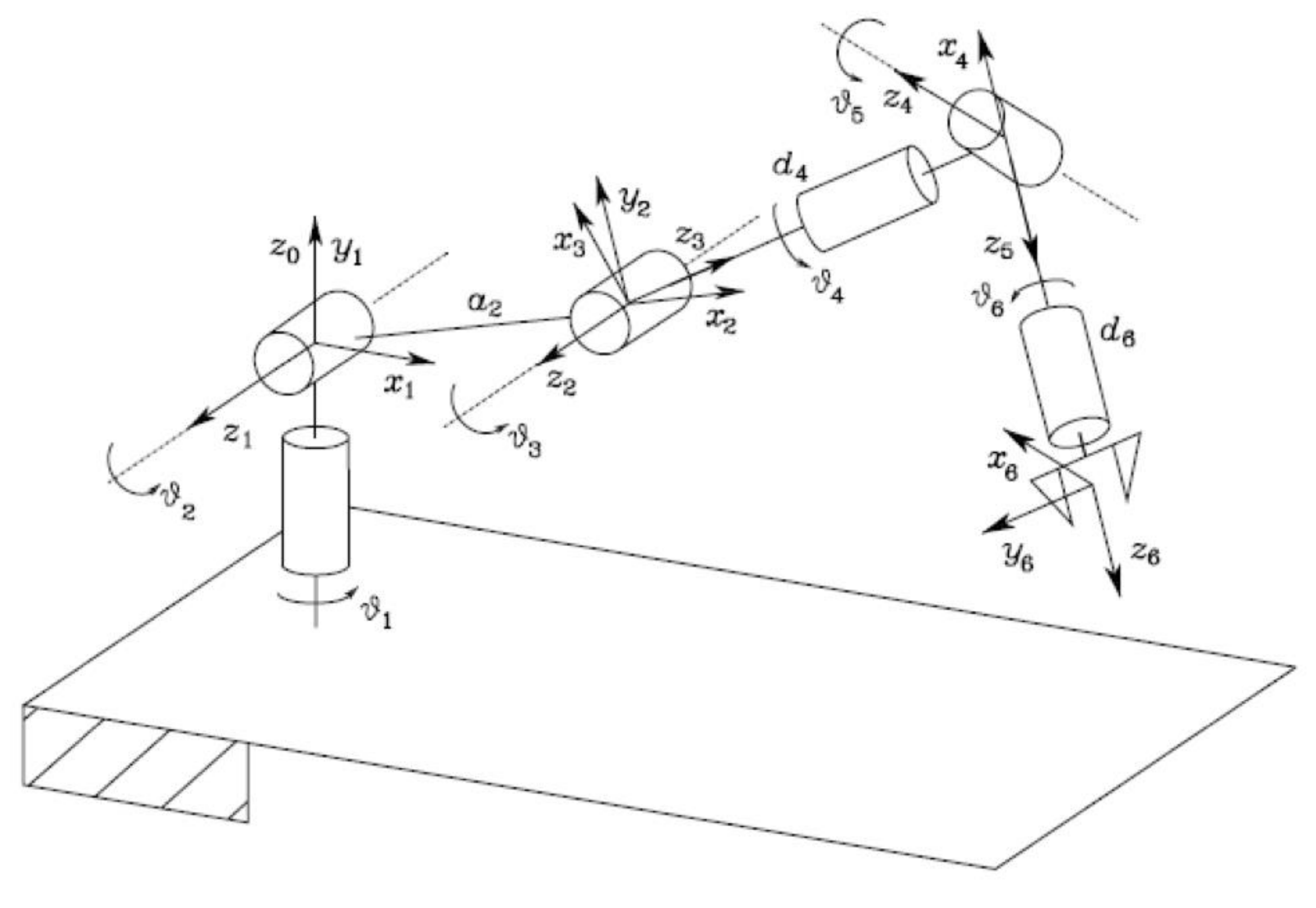
Robotics involves kinematics, dynamics and control.

Kinematics

$$T_e(q) = \begin{bmatrix} R_e(q) & p_e(q) \\ 0^T & 1 \end{bmatrix}$$

Differential Kinematics

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J(q)\dot{q}$$



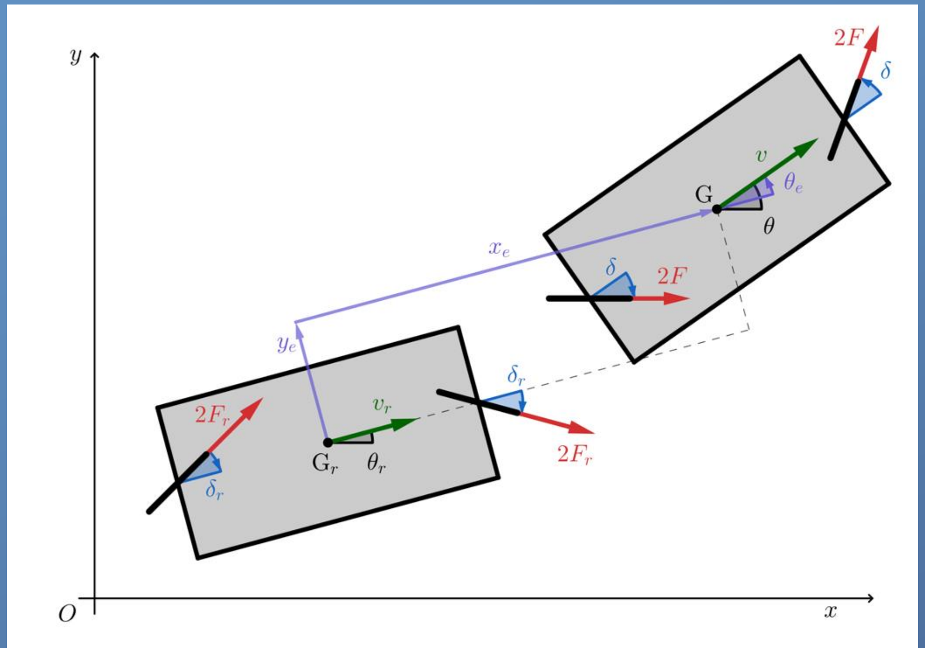
Dynamical Model

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + F_s\,\mathrm{sgn}\,(\dot{q}) + g(q) = \tau - J^T(q)h_e$$

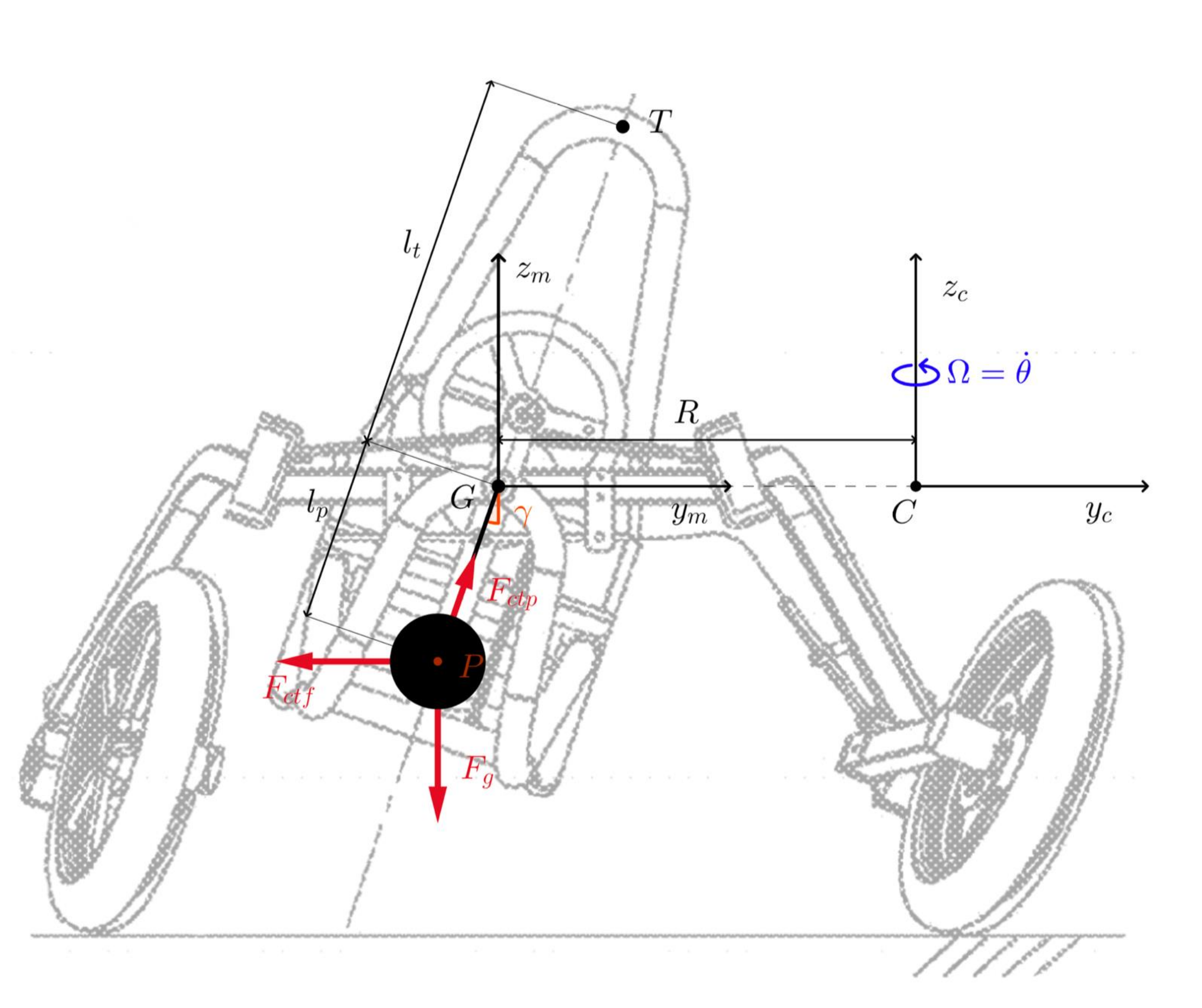
Robotics:

Robotics involves kinematics, dynamics and control.

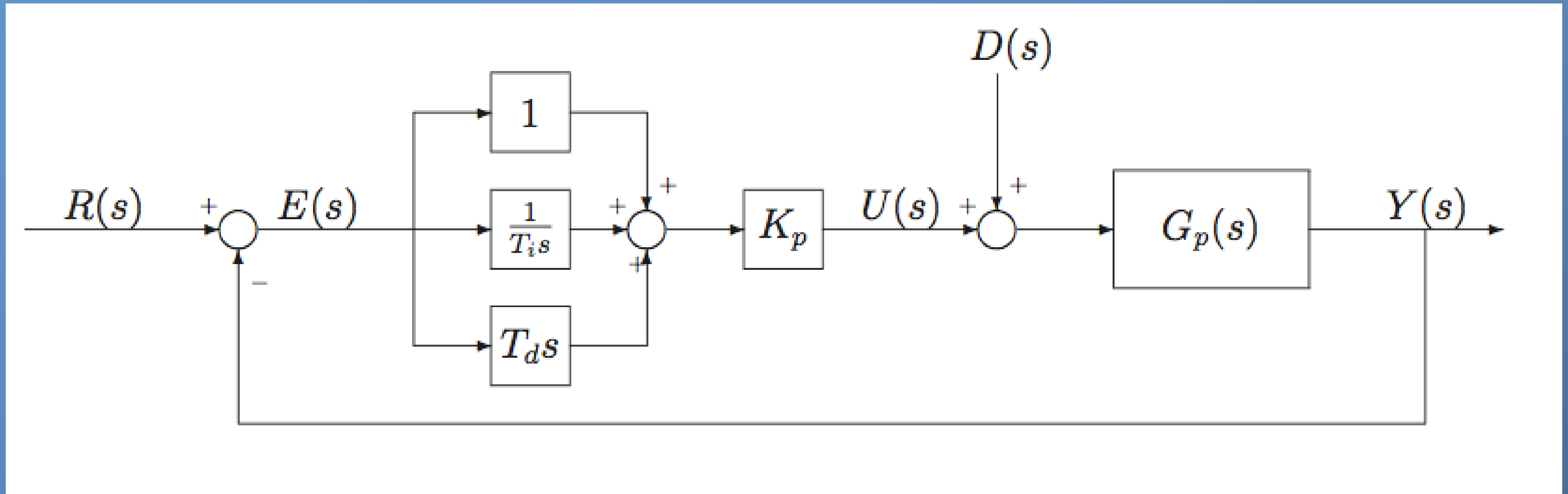
Tracking error



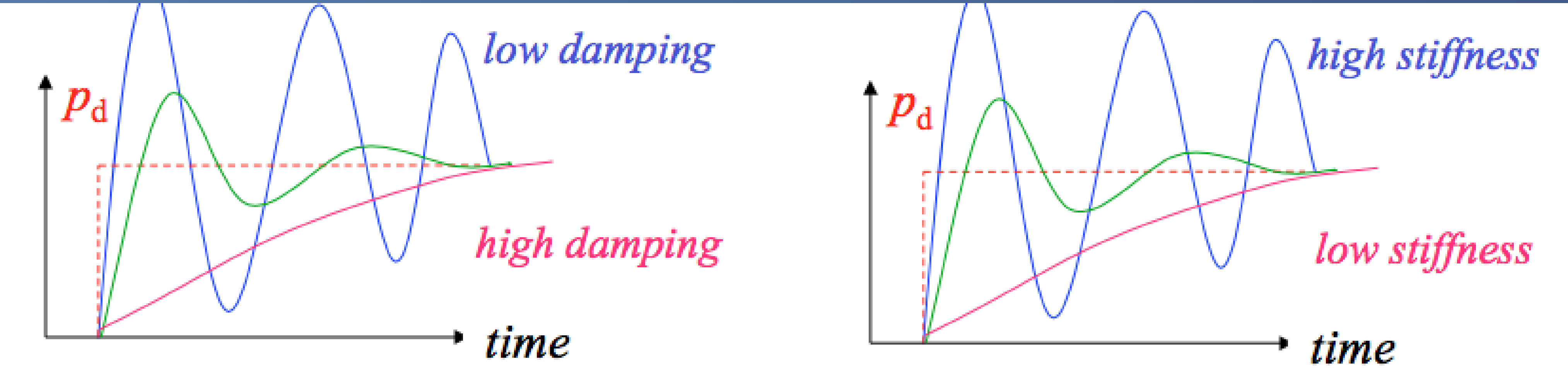
$$\begin{cases} x_e = (x - x_r)\cos\theta_r + (y - y_r)\sin\theta_r \\ y_e = -(x - x_r)\sin\theta_r + (y - y_r)\cos\theta_r \\ \theta_e = \theta - \theta_r, \end{cases}$$



Control Techniques: PID (Proportional, Integral, Derivative)



$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\eta) d\eta + T_d \frac{de(t)}{dt} \right]$$

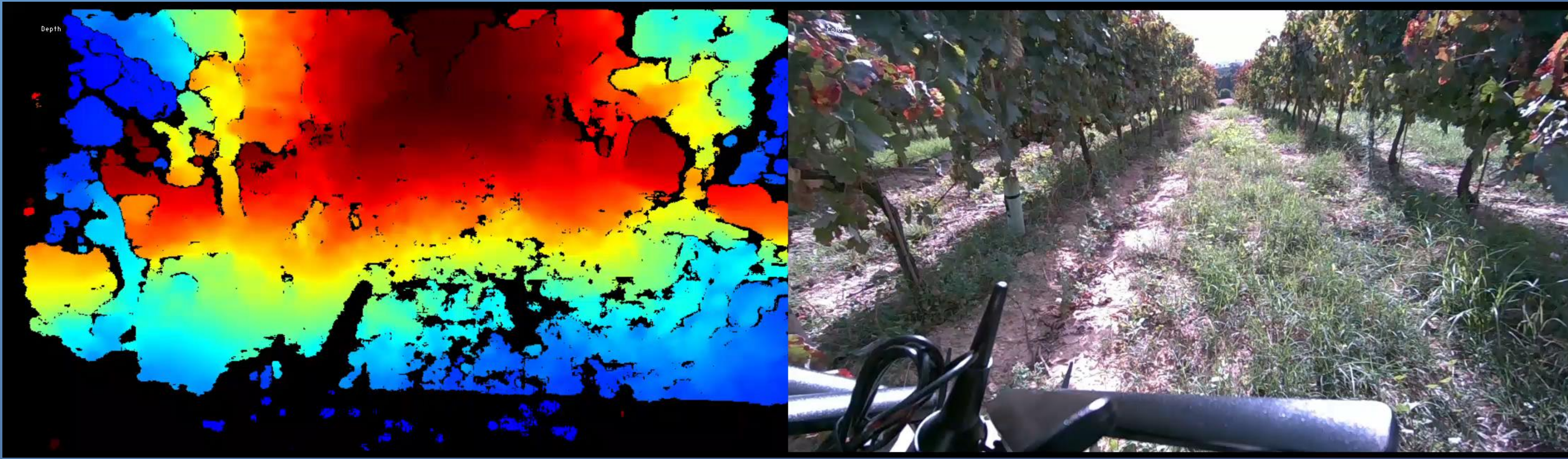


Actuation: Eletric Motors *(the sound of silence)*

Positioning: GPS RTK

Positioning: GPS RTK

Navigation: RGB-D imaging



Intelligent Systems:

"Machine learning is imperfect – That's why it's ideal for agriculture."
Elliot Grant, Mineral.ai

Intelligent Systems:

"Machine learning is imperfect – That's why it's ideal for agriculture."

Elliot Grant, Mineral.ai

Neural Network frameworks

TensorFlow

PyTorch

Keras

theano

Caffe

Chainer

Deeplearning4j

Microsoft Cognitive Toolkit

Robotics:

Machine Intelligence involves Robotics and AI

Robotics:

Machine Intelligence involves Robotics and AI

ROBOTICS 4 FARMERS Sistemas Robóticos Ecoeficientes e Colaborativos para uma Agricultura Inovadora, Inclusiva e Sustentável

AI enabled autonomous navigation in agricultural environments

Robotics:

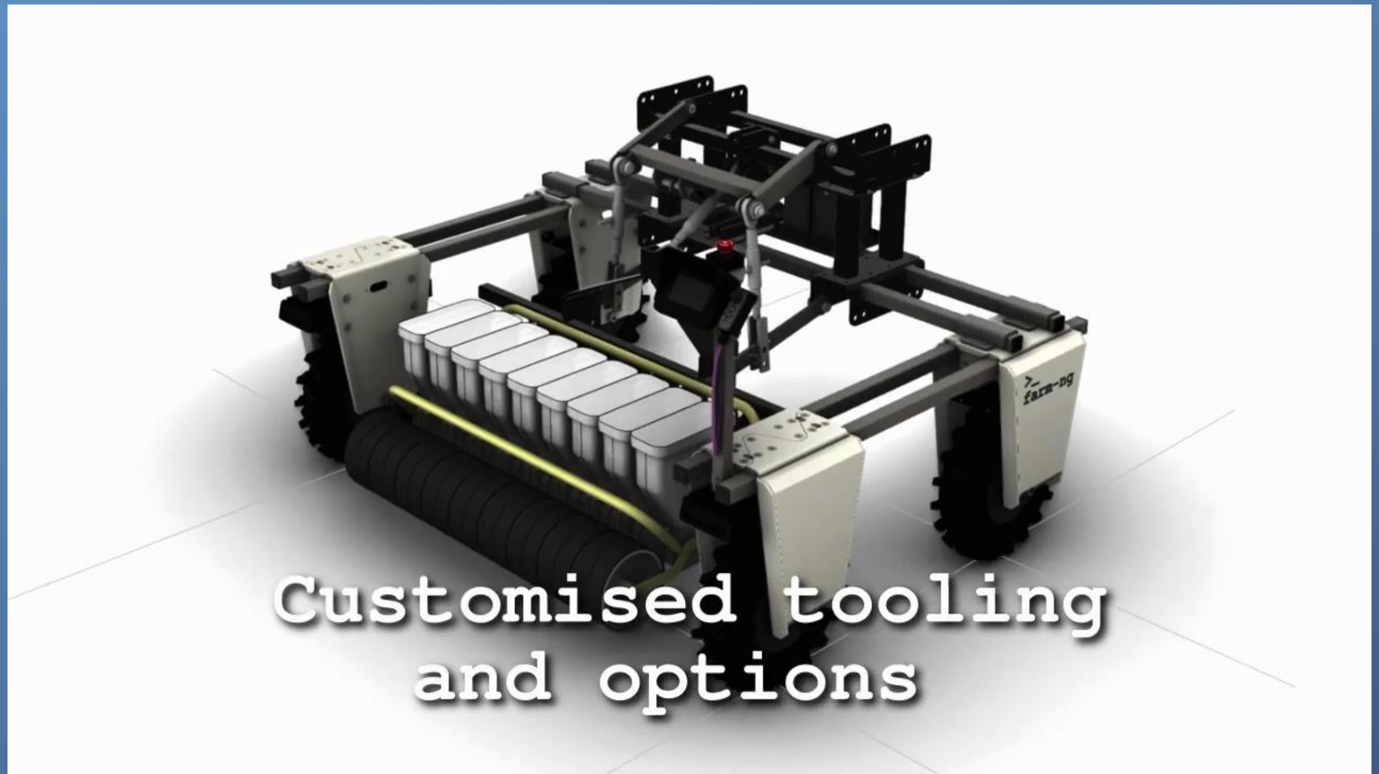
Machine Intelligence involves Robotics and AI

ROBOTICS 4 FARMERS

Sistemas Robóticos Ecoeficientes e Colaborativos para uma Agricultura Inovadora, Inclusiva e Sustentável

Agriculture Robotics Challenges:

>>>{Farm_\ Robotics_\ Challenge}



Control, Dynamics and Robotics

SIRCA-2526

Jorge Martins

Instituto Superior Técnico
Department of Mechanical Engineering
Center of Intelligent Systems – idMEC
Human Robotics Group
JorgeMartins@tecnico.ulisboa.pt

Trajectory Planning

Goal: To generate the reference inputs to the motion control system which ensures that the manipulator executes the planned trajectories.

Trajectory planning in the joint space

Trajectory planning in the operational space

Point-to-point motion

Motion through a sequence of points

Path ≠ **Trajectory**

The locus of points in the joint space, or in the operational space , which the manipulator has to follow in the execution of the assigned motion

A path on which a timing law is specified.

Trajectory planning in the joint space

Point-to-point motion – Polynomial trajectory profiles

Knowing the initial $(q_i, \dot{q}_i)$ and final $(q_f, \dot{q}_f, t_f)$ joint positions and velocities determine the coefficients of the *cubic polynomial*:

$$q(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

For the velocities and accelerations results

$$\dot{q}(t) = 3a_3 t^2 + 2a_2 t + a_1$$

$$\ddot{q}(t) = 6a_3 t + 2a_2.$$

and the trajectory coefficients are obtained by solving the boundary equations:

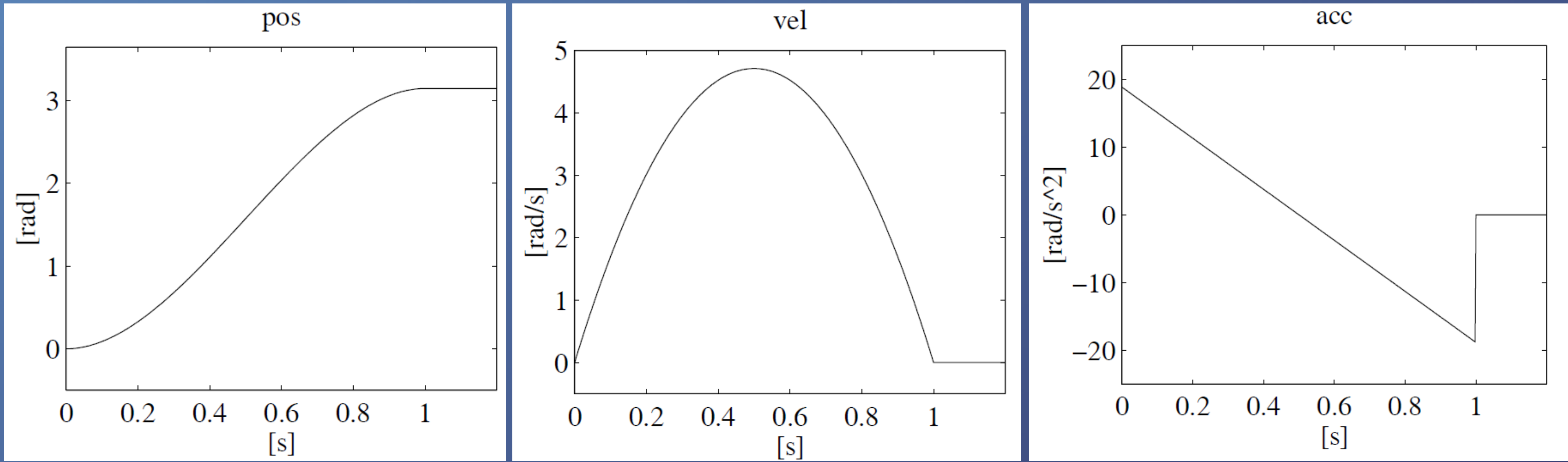
$$\begin{aligned} a_0 &= q_i \\ a_1 &= \dot{q}_i \\ a_3 t_f^3 + a_2 t_f^2 + a_1 t_f + a_0 &= q_f \\ 3a_3 t_f^2 + 2a_2 t_f + a_1 &= \dot{q}_f, \end{aligned}$$

Trajectory planning in the joint space

Point-to-point motion – Polynomial trajectory profiles

$$q(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$q_i = 0,\ q_f = \pi,\ t_f = 1, \text{ and } \dot{q}_i = \dot{q}_f = 0$$



Trajectory planning in the joint space

Point-to-point motion – Polynomial trajectory profile

Knowing the initial $(q_i, \dot{q}_i, \ddot{q}_i)$ and final $(q_f, \dot{q}_f, \ddot{q}_f, t_f)$ joint positions, velocities and accelerations determine the coefficients of the *fifth order polynomial*:

$$q(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0.$$

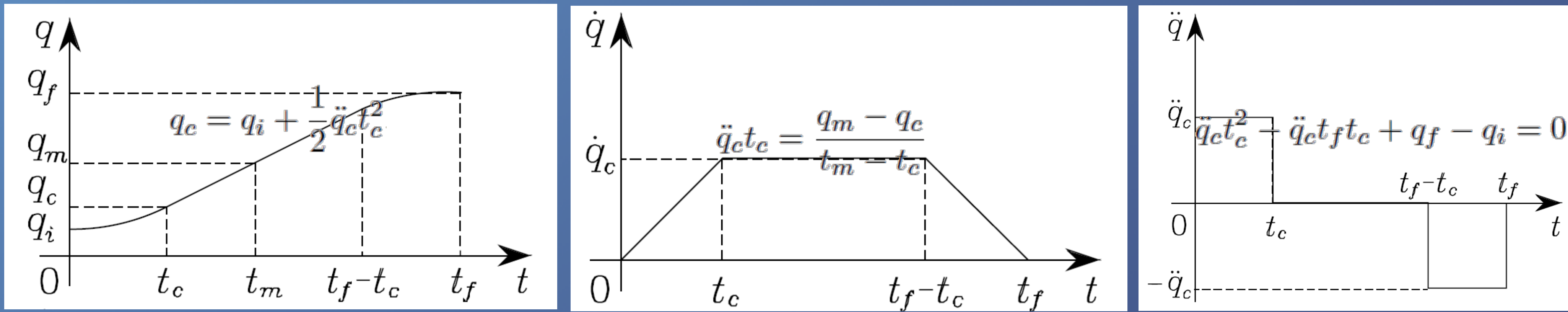
The same steps as for the cubic polynomial hold.

Higher order polynomials may be considered ….

Trajectory planning in the joint space

Point-to-point motion – Blended Polynomial trajectory profiles

When one wishes to easily verify the velocities and accelerations of the trajectory against the maximum allowed by the manipulator, blended polynomials are preferred:



Knowing the initial (q_i) and final (q_f, t_f) joint positions and the transition velocity $(\dot{q}_c)$ determine the three blended *polynomials*:

$$q(t) = \begin{cases} q_i + \frac{1}{2}\ddot{q}_c t^2 & 0 \le t \le t_c \\ q_i + \ddot{q}_c t_c (t - t_c/2) & t_c < t \le t_f - t_c \\ q_f - \frac{1}{2}\ddot{q}_c (t_f - t)^2 & t_f - t_c < t \le t_f . \end{cases}$$

with

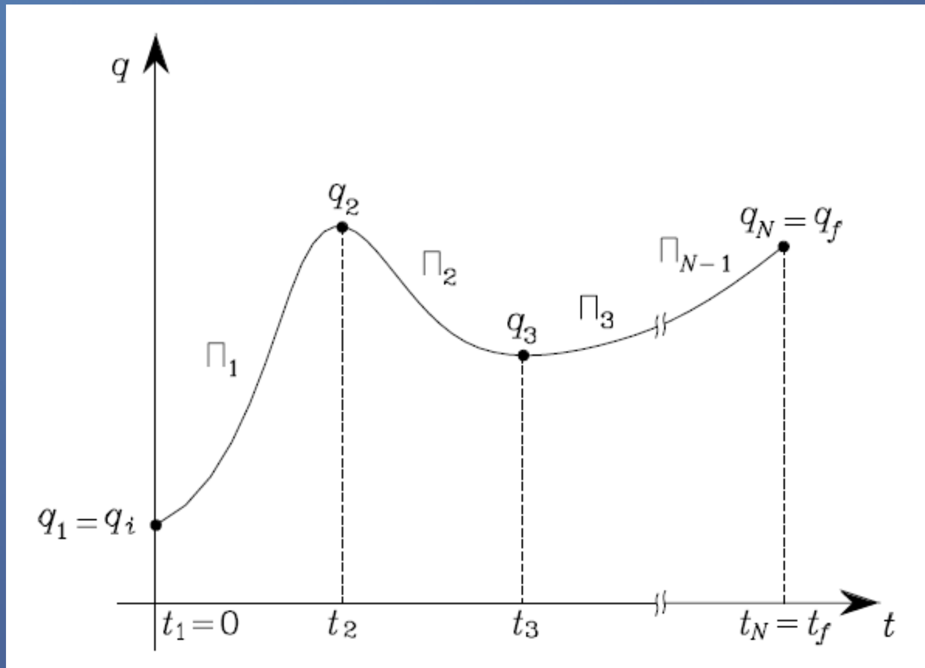
$$t_c = \frac{q_i - q_f + \dot{q}_c t_f}{\dot{q}_c}$$

$$\ddot{q}_c = \frac{\dot{q}_c^2}{q_i - q_f + \dot{q}_c t_f}$$

Trajectory planning in the joint space

Motion through a sequence of points

In many practical applications, it is necessary to describe the path with a number of points greater than two:

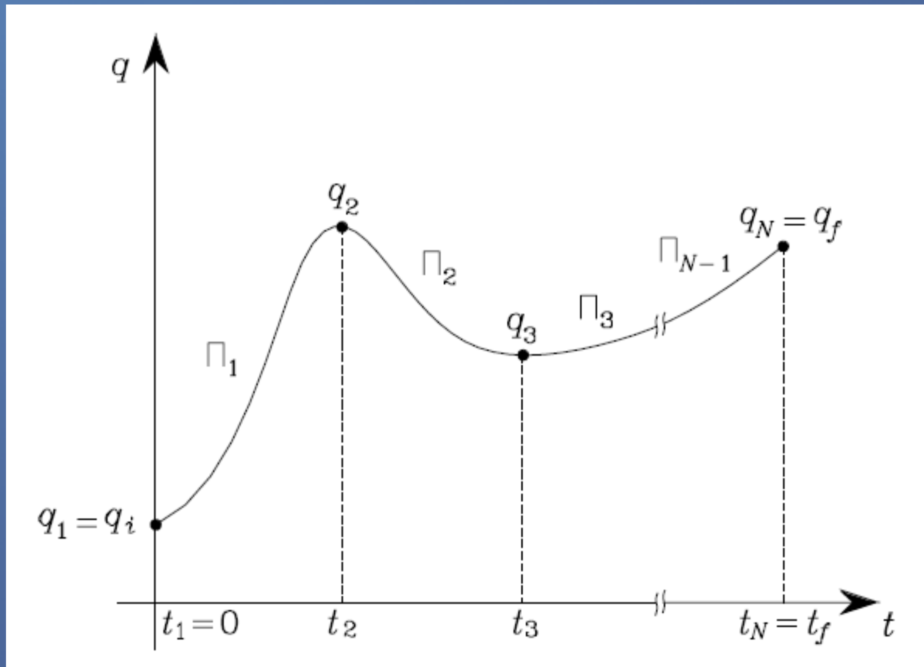


The problem is now to generate a trajectory where *N* points, termed *path points* , have to be reached by the manipulator at certain instants of time.

Trajectory planning in the joint space

Motion through a sequence of points

The solution is to choose *N-1* cubic polynomials continuous at the path points



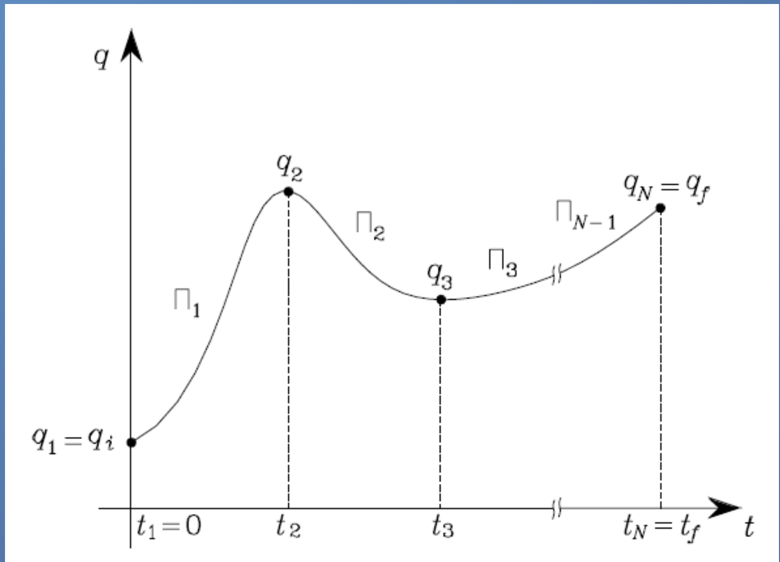
The following situations can be considered:

1. Interpolating polynomials with imposed velocities at path points;
2. Interpolating polynomials with continuous accelerations at path points (splines)

Trajectory planning in the joint space

Motion through a sequence of points

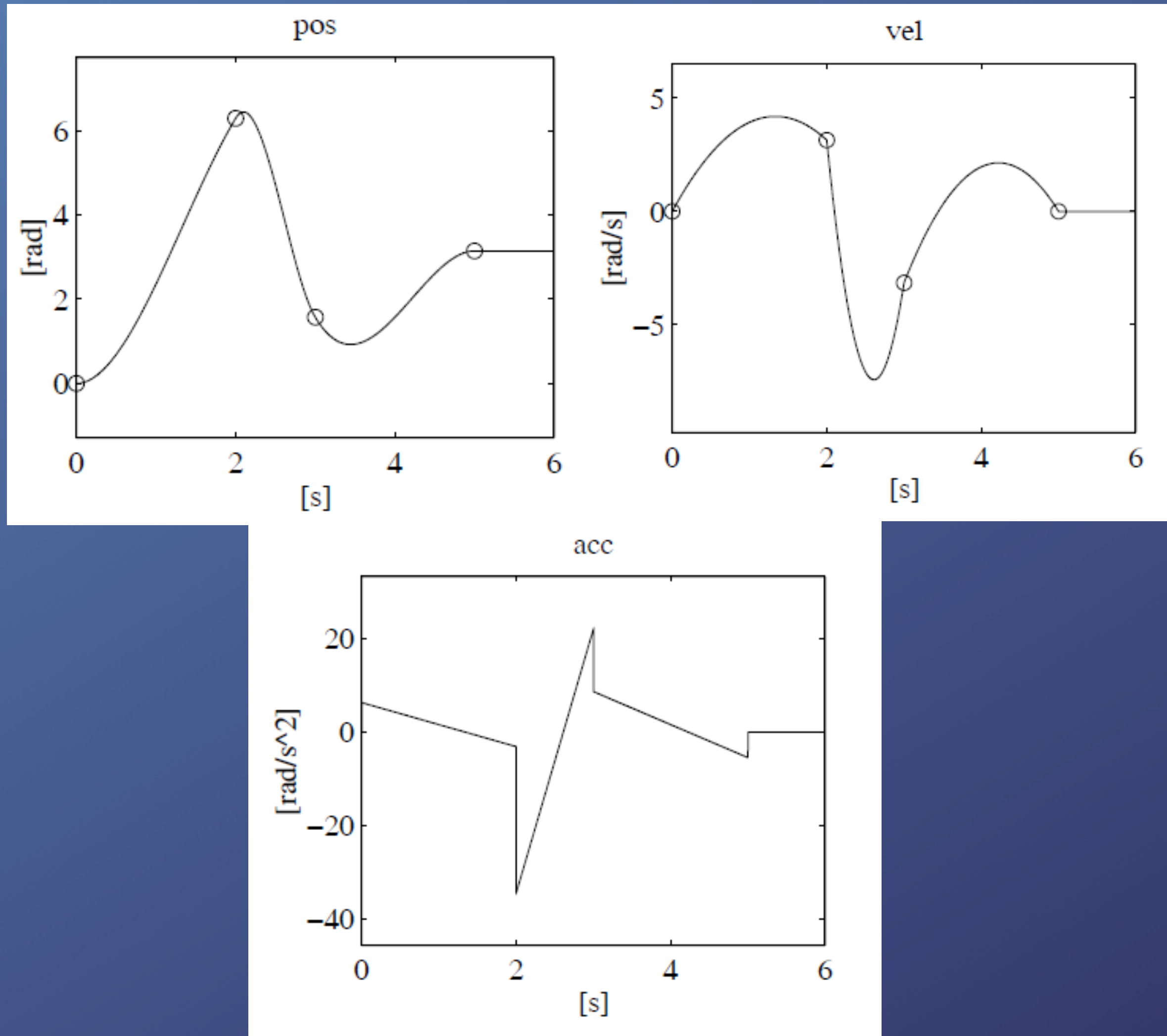
1. Interpolating polynomials with imposed velocities at path points;



for $k = 1, \ldots, N-1$:

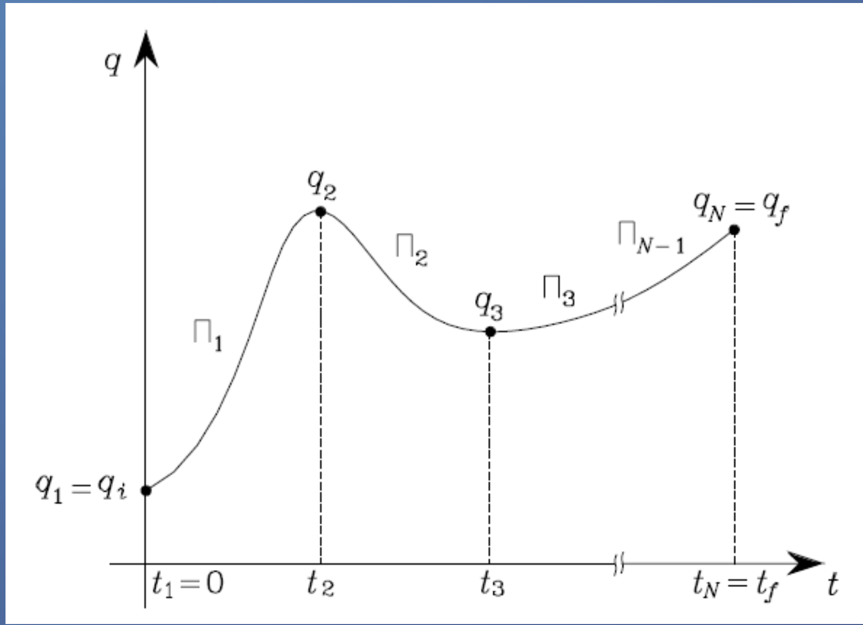
$$\Pi_k(t_k) = q_k$$
$$\Pi_k(t_{k+1}) = q_{k+1}$$
$$\dot{\Pi}_k(t_k) = \dot{q}_k$$
$$\dot{\Pi}_k(t_{k+1}) = \dot{q}_{k+1}.$$

$$\ddot{\Pi}_k(t_{k+1}) = \ddot{\Pi}_{k+1}(t_{k+1})$$



Trajectory planning in the joint space
Motion through a sequence of points

2. Interpolating polynomials with continuous accelerations at path points (splines)

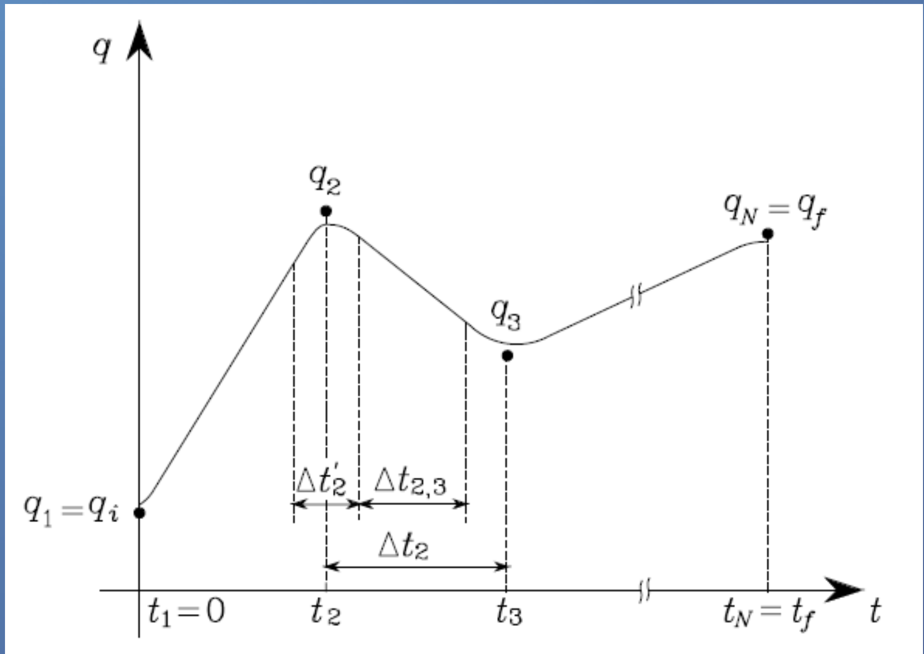


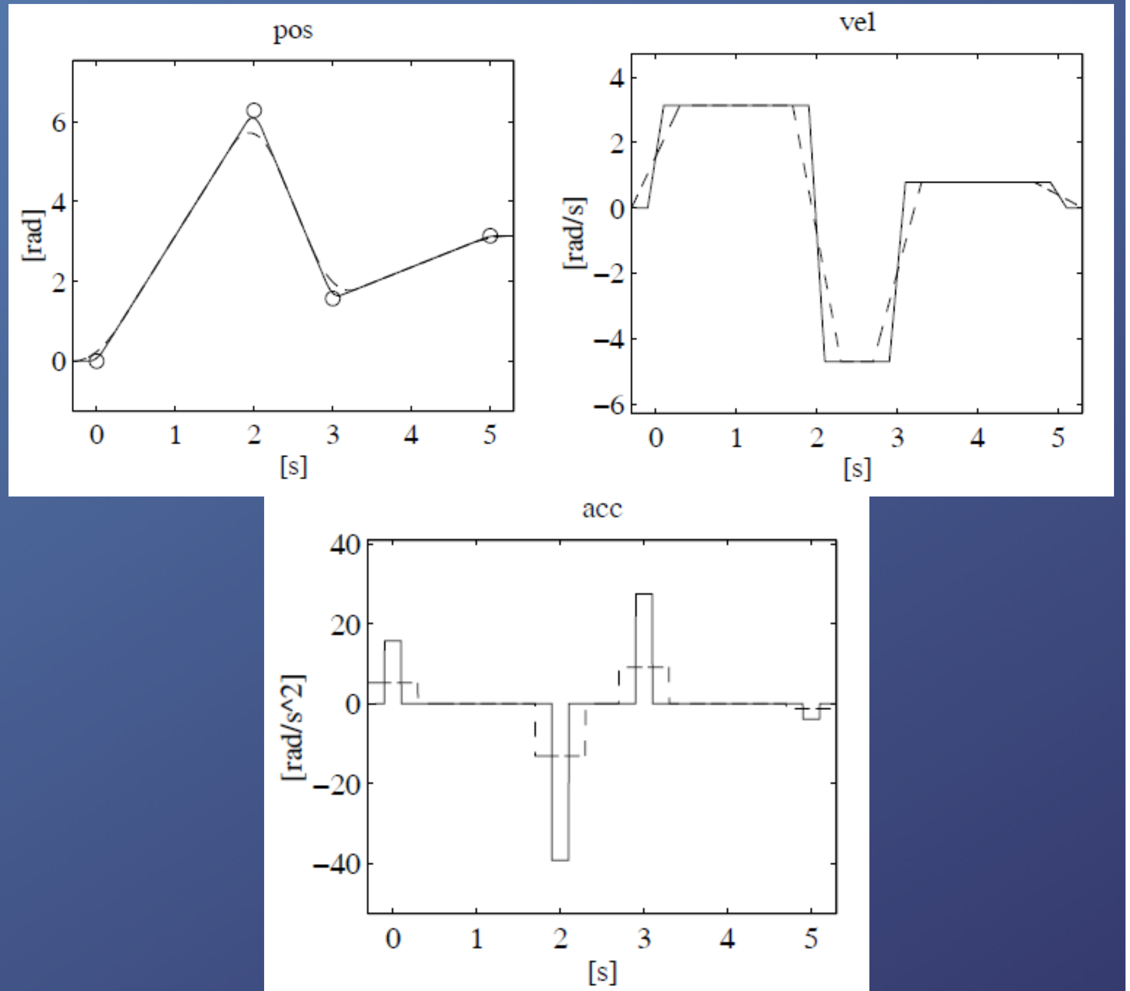
$$\begin{aligned}
\Pi_{k-1}(t_k) &= q_k \\
\Pi_{k-1}(t_k) &= \Pi_k(t_k) \\
\dot{\Pi}_{k-1}(t_k) &= \dot{\Pi}_k(t_k) \\
\ddot{\Pi}_{k-1}(t_k) &= \ddot{\Pi}_k(t_k).
\end{aligned}$$

Trajectory planning in the joint space

Motion through a sequence of points

Approximating linear polynomials with parabolic blends





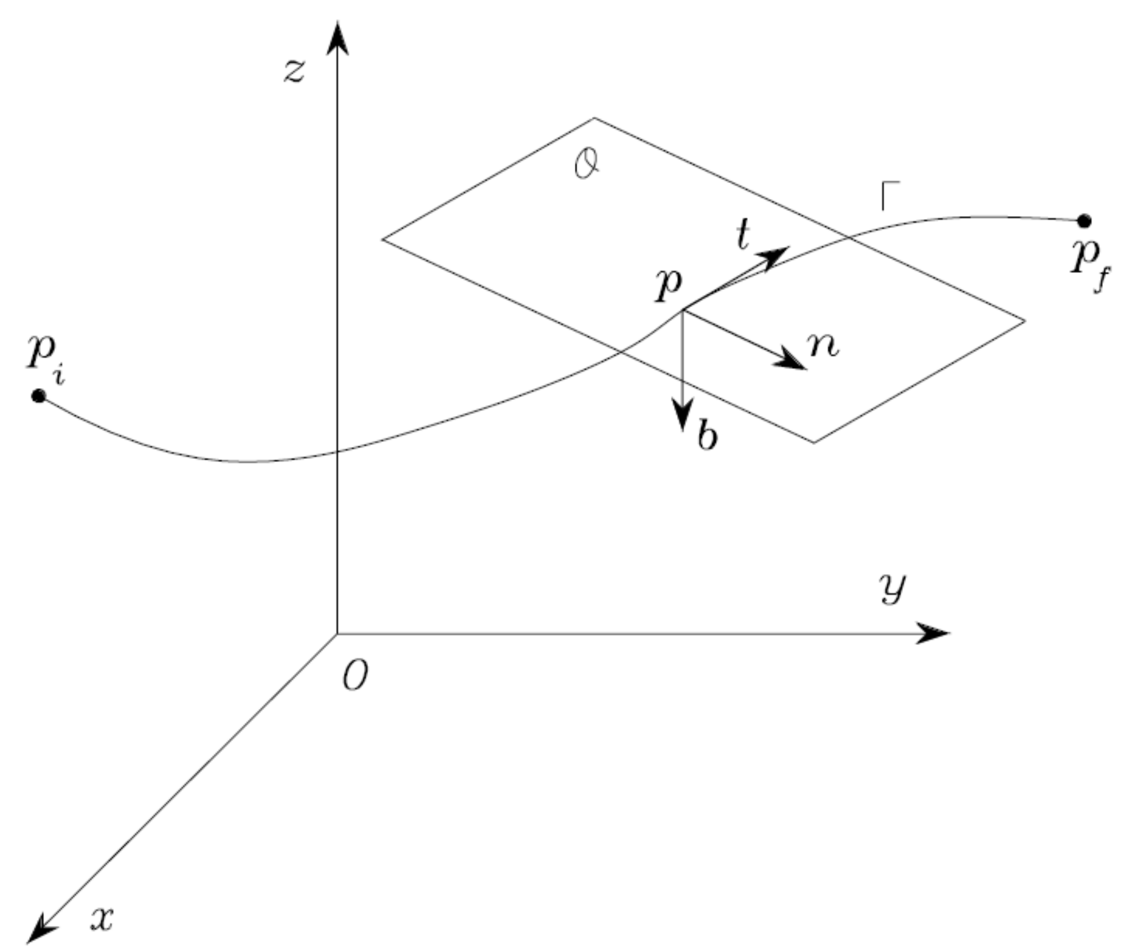
$$\dot{q}_{k-1,k} = \frac{q_k - q_{k-1}}{\Delta t_{k-1}}$$

$$\ddot{q}_k = \frac{\dot{q}_{k,k+1} - \dot{q}_{k-1,k}}{\Delta t'_k};$$

Trajectory planning in the operational space
Analytical trajectories – parametric curves

Parametric description of trajectories in space

- Motion Primitives – geometric features of the path
- Time Primitives – timing law on the path



$$\boldsymbol{p} = \boldsymbol{f}(s)$$

s – Arc Length

Path reference frame

$$\boldsymbol{t} = \frac{d\boldsymbol{p}}{ds}$$

$$\boldsymbol{n} = \frac{1}{\left\| \frac{d^2\boldsymbol{p}}{ds^2} \right\|} \frac{d^2\boldsymbol{p}}{ds^2}$$

$$\boldsymbol{b} = \boldsymbol{t} \times \boldsymbol{n}.$$

Trajectory planning in the operational space

Analytical trajectories – Rectilinear path

$$\boldsymbol{p}(s) = \boldsymbol{p}_i + \frac{s}{\|\boldsymbol{p}_f - \boldsymbol{p}_i\|}(\boldsymbol{p}_f - \boldsymbol{p}_i)$$

$$\boldsymbol{p}(0) = \boldsymbol{p}_i \text{ and } \boldsymbol{p}(\|\boldsymbol{p}_f - \boldsymbol{p}_i\|) = \boldsymbol{p}_f$$

$$\frac{d\boldsymbol{p}}{ds} = \frac{1}{\|\boldsymbol{p}_f - \boldsymbol{p}_i\|}(\boldsymbol{p}_f - \boldsymbol{p}_i)$$

$$\frac{d^2\boldsymbol{p}}{ds^2} = \boldsymbol{0}.$$

For a rectilinear path it is not possible to define the path reference frame uniquely.

Timing law for manipulator end effector position

$$\dot{\boldsymbol{p}}_e = \dot{s}\frac{d\boldsymbol{p}_e}{ds} = \dot{s}\boldsymbol{t}$$

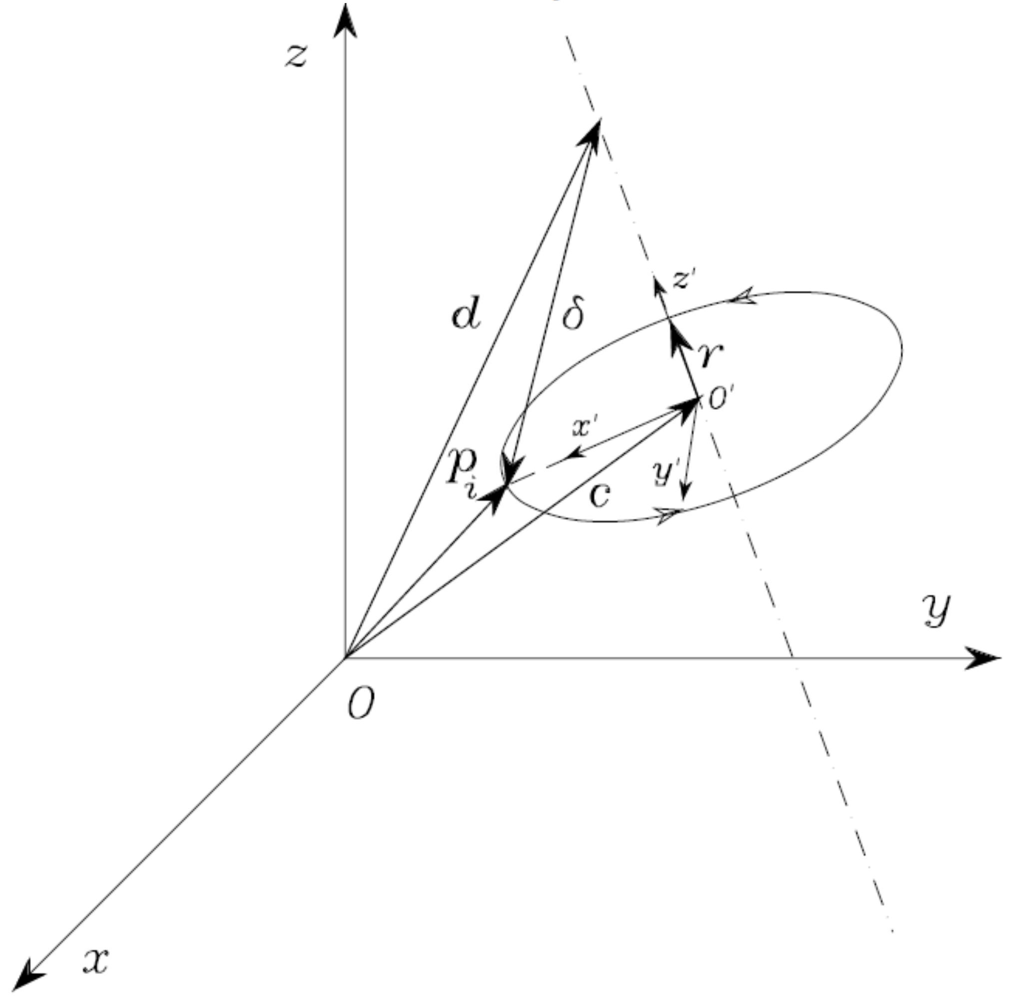
$$\dot{\boldsymbol{p}}_e = \frac{\dot{s}}{\|\boldsymbol{p}_f - \boldsymbol{p}_i\|}(\boldsymbol{p}_f - \boldsymbol{p}_i) = \dot{s}\boldsymbol{t}$$

$$\ddot{\boldsymbol{p}}_e = \frac{\ddot{s}}{\|\boldsymbol{p}_f - \boldsymbol{p}_i\|}(\boldsymbol{p}_f - \boldsymbol{p}_i) = \ddot{s}\boldsymbol{t}.$$

Trajectory planning in the operational space

Analytical trajectories – Circular path

the unit vector of the circle axis $\boldsymbol{r}$,
the position vector $\boldsymbol{d}$ of a point along the circle axis,
the position vector $\boldsymbol{p}_i$ of a point on the circle.



c – center of the circle

$$\boldsymbol{c} = \boldsymbol{d} + (\boldsymbol{\delta}^T \boldsymbol{r})\boldsymbol{r}$$

ρ – radius of the circle

$$\rho = \|\boldsymbol{p}_i - \boldsymbol{c}\|$$

Circle reference frame

$$\boldsymbol{R} = [\,\boldsymbol{x}' \quad \boldsymbol{y}' \quad \boldsymbol{z}'\,]$$

Parametric representation

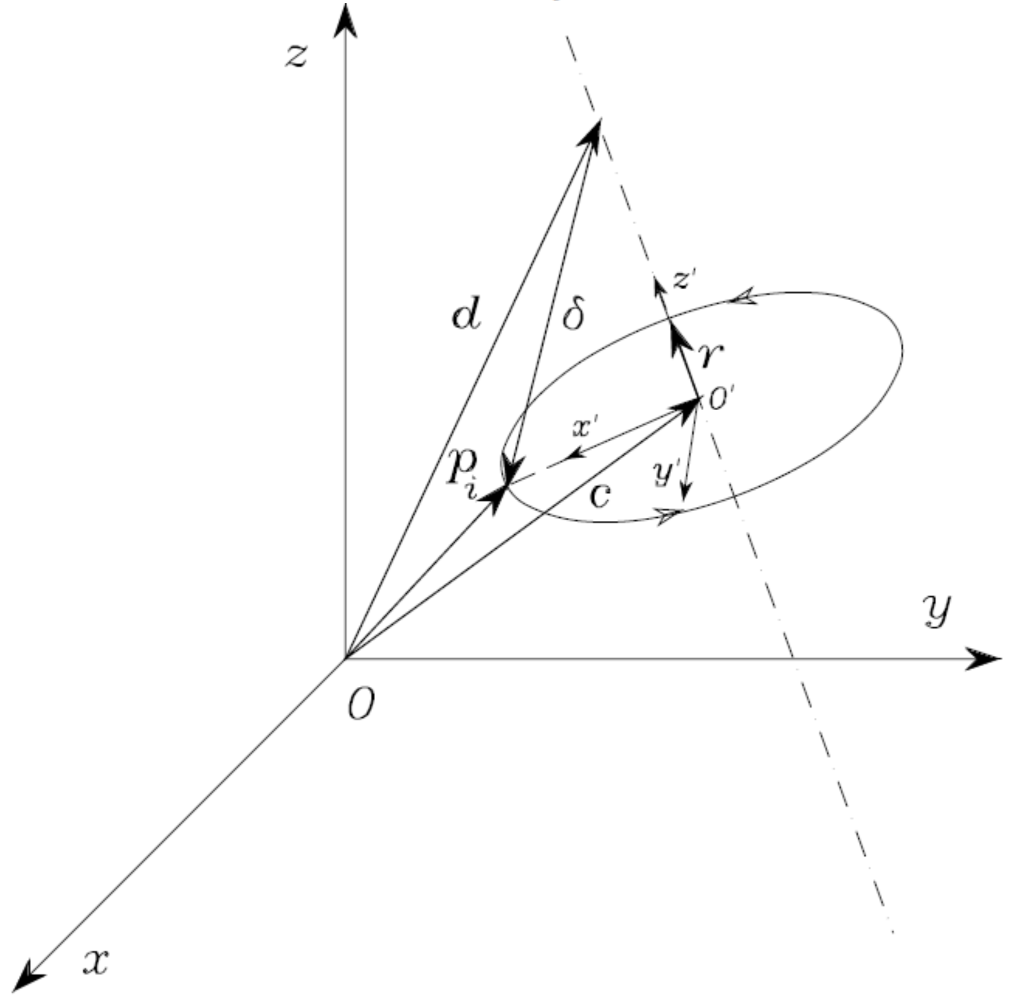
$$\boldsymbol{p}'(s) = \begin{bmatrix} \rho\cos(s/\rho) \\ \rho\sin(s/\rho) \\ 0 \end{bmatrix}$$

$$\boldsymbol{p}(s) = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{p}'(s)$$

Trajectory planning in the operational space

Analytical trajectories – Circular path

the unit vector of the circle axis $\boldsymbol{r}$,
the position vector $\boldsymbol{d}$ of a point along the circle axis,
the position vector $\boldsymbol{p}_i$ of a point on the circle.



Parametric representation

$$\boldsymbol{p}(s) = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{p}'(s)$$

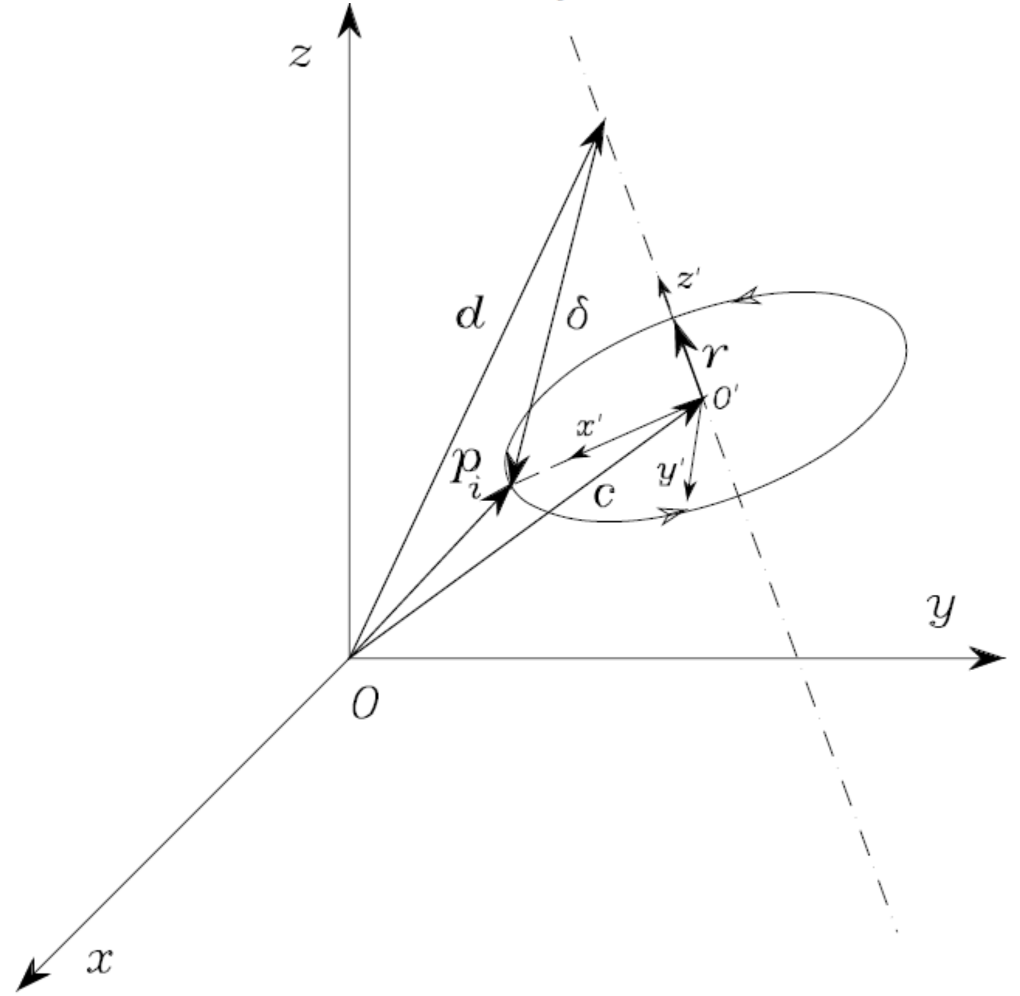
$$\frac{d\boldsymbol{p}}{ds} = \boldsymbol{R}\begin{bmatrix} -\sin(s/\rho) \\ \cos(s/\rho) \\ 0 \end{bmatrix}$$

$$\frac{d^2\boldsymbol{p}}{ds^2} = \boldsymbol{R}\begin{bmatrix} -\cos(s/\rho)/\rho \\ -\sin(s/\rho)/\rho \\ 0 \end{bmatrix}$$

Trajectory planning in the operational space

Analytical trajectories – Circular path

the unit vector of the circle axis $\boldsymbol{r}$,
the position vector $\boldsymbol{d}$ of a point along the circle axis,
the position vector $\boldsymbol{p}_i$ of a point on the circle.



Timing Law

$$\dot{\boldsymbol{p}}_e = \dot{s}\frac{d\boldsymbol{p}_e}{ds} = \dot{s}\boldsymbol{t}$$

$$\dot{\boldsymbol{p}}_e = \boldsymbol{R}\begin{bmatrix} -\dot{s}\sin(s/\rho) \\ \dot{s}\cos(s/\rho) \\ 0 \end{bmatrix}$$

$$\ddot{\boldsymbol{p}}_e = \boldsymbol{R}\begin{bmatrix} -\dot{s}^2\cos(s/\rho)/\rho - \ddot{s}\sin(s/\rho) \\ -\dot{s}^2\sin(s/\rho)/\rho + \ddot{s}\cos(s/\rho) \\ 0 \end{bmatrix}$$

Trajectory planning in the operational space
Analytical trajectories – Orientation

Euler Angles

$$\phi_e = (\varphi, \vartheta, \psi)$$

Linear segment

$$\phi_e = \phi_i + \frac{s}{\|\phi_f - \phi_i\|}(\phi_f - \phi_i)$$
$$\dot{\phi}_e = \frac{\dot{s}}{\|\phi_f - \phi_i\|}(\phi_f - \phi_i)$$
$$\ddot{\phi}_e = \frac{\ddot{s}}{\|\phi_f - \phi_i\|}(\phi_f - \phi_i);$$

Angle and Axis

Given R_i *and* R_f *at* t_f :

$$R_f^i = R_i^T R_f = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\vartheta_f = \cos^{-1}\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right)$$

$$\boldsymbol{r} = \frac{1}{2\sin\vartheta_f}\begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix}$$

The matrix $R^i(t)$ with $R^i(0) = I$ and $R^i(t_f) = R_f^i$ can be interpreted as the matrix $R^i(\vartheta(t), r^i)$, a *single joint problem*, where it suffices to assign a timing law to $\vartheta(t)$: $\vartheta(0) = 0$ and $\vartheta(t_f) = \vartheta_f$ with constant r^i

$$\boldsymbol{\omega}^i = \dot{\vartheta}\,\boldsymbol{r}^i$$
$$\dot{\boldsymbol{\omega}}^i = \ddot{\vartheta}\,\boldsymbol{r}^i.$$

$$\boldsymbol{R}_e(t) = \boldsymbol{R}_i \boldsymbol{R}^i(t)$$
$$\boldsymbol{\omega}_e(t) = \boldsymbol{R}_i \boldsymbol{\omega}^i(t)$$
$$\dot{\boldsymbol{\omega}}_e(t) = \boldsymbol{R}_i \dot{\boldsymbol{\omega}}^i(t).$$