AN OVERVIEW OF ROBOT CONTROL FOR MANIPULATION. IN 3 HOURS

Guillaume Morel

Sorbonne Université

Robotics Principia, Clermont Ferrand – April 2026

CONTENTS

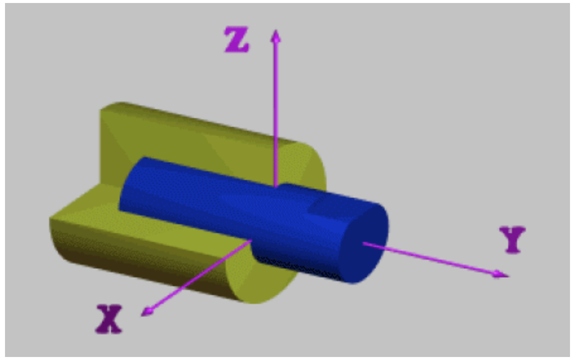
1. Models that are useful for robot control
   - Modelling a robot joint (a little bit of technology)
   - Multi-joint models (a little bit of multibody kinematics and dynamics)
2. Control of robot movements in the free space:
   - With velocity controlled actuators
   - With torque control actuators
3. Controlling multi-joint robots when contacts occur with the environment:
   - Force control, hybrid position/force control
   - Impedance control

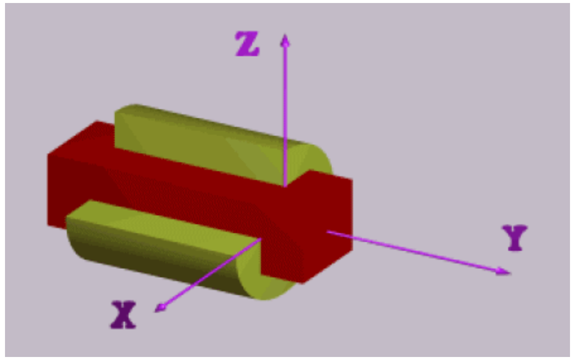
PART 1: MODELLING

1.1 Modelling a robot joint

Types of joints and main component

- 2 main types:
  - Revolute: the mobile body rotates around a fixed axis with respect to the fixed body, called joint axis.
  - Linear: the mobile body translates along a fixed axis with respect to the fixed body, called joint axis as well.
- Composed of:
  - An actuator: converts (electric) energy into mechanical energy
  - A guidance system: constrains the type of movement (revolute or linear)
  - A mechanical transmission system: transmits mechanical power from the actuator to the mobile body
  - A set of sensors: either embedded in the actuator or added in the joint
  - An electronic interface with a computer



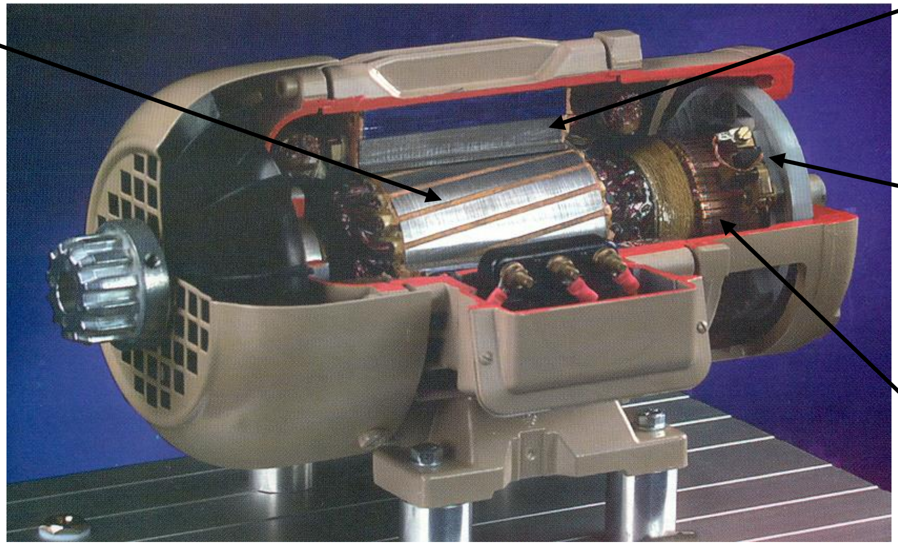


Two main types of actuators

In a first approximation an actuator can be either

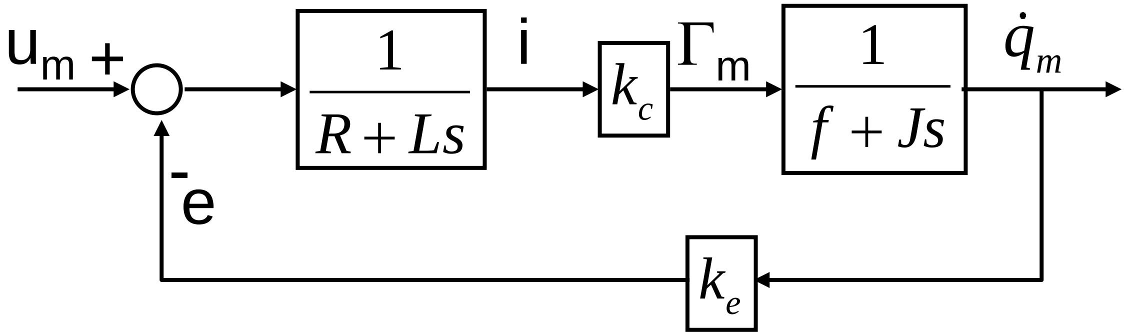
- A velocity generator, receiving a command $\dot{q}_c$ and generating a velocity $\dot{q}$, namely:
  - A rotational velocity (in rad/s) for rotative actuators
  - A linear velocity (in m/s) for linear actuators
- Or a generalized force generator, receiving a command γ_c and generating a generalized force γ, namely:
  - A torque around the actuator axis (in Nm) for rotative actuators
  - A force along the actuator axes (in m) for linear actuators

Brushed DC motors as an example



Linear model

- Electric dynamics within the armature : $u_m - e = Ri + L\frac{di}{dt}$, where R is the electric resistance, L is the self induction coefficient, and e is a voltage resulting from movement (back electromotive force, *bemf*): $e = k_e\, \dot{q}_m$
- The motor torque γ_m is then : $\gamma_m = k_c i$ where k_c is the torque constant.
- The torque induces an angular acceleration given by : $J\ddot{q}_m = \gamma_m - f\dot{q}_m$ where J is the motor inertia and f the coefficient of viscous friction.

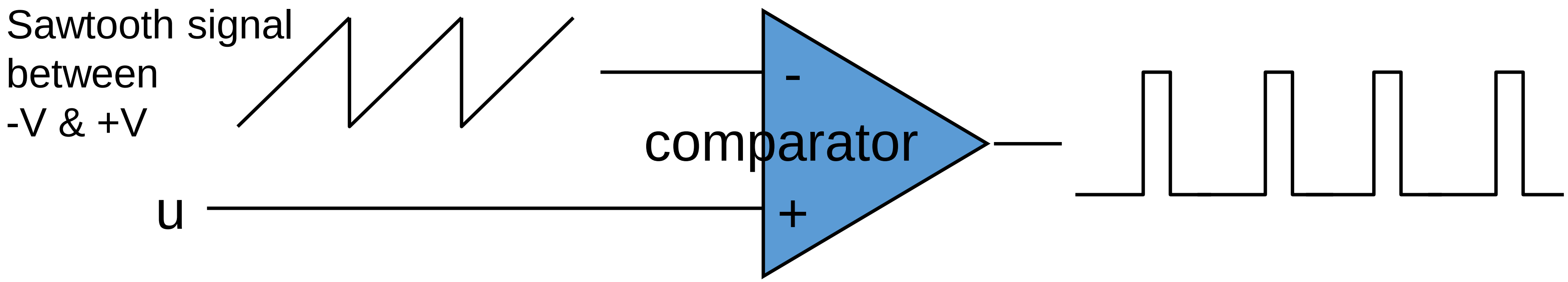


Power amplification

- Role :
  - Amplify the electric power from a digital device (e.g. a computer) to a high power device.
    - Example: An analog output +-10V, 10mA max (0.1 W max) cannot drive electric motor 10V-3A (30 W max).
  - Low level current control loop aimed at:
    - Improving safety (against high currents).
    - Linearizing (against non linear disturbances of any kinds).
    - Controlling Torques (as $\gamma_m = k_c i$).

EXAMPLE : ELECTRONIC CHOPPER

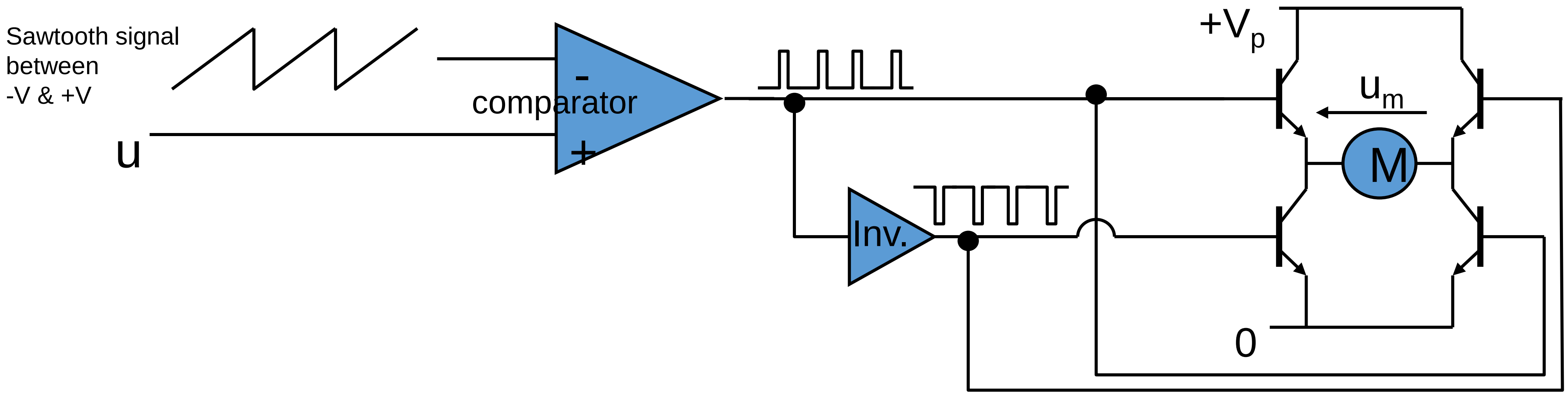
- Principle of an electronic chopper:
  - The output of the controller is an analog voltage u between+V & -V.
  - This voltage is converted into a pulse width modulated (PWM) signal by the following simple circuit :



The the duty cycle of the output signal is : $\alpha = \frac{1}{2}\left(\frac{u}{V}+1\right)$

Example : electronic chopper

- The pulse width modulated signal (PWM) then controls a H bridge of switching transistors (controlled switches).



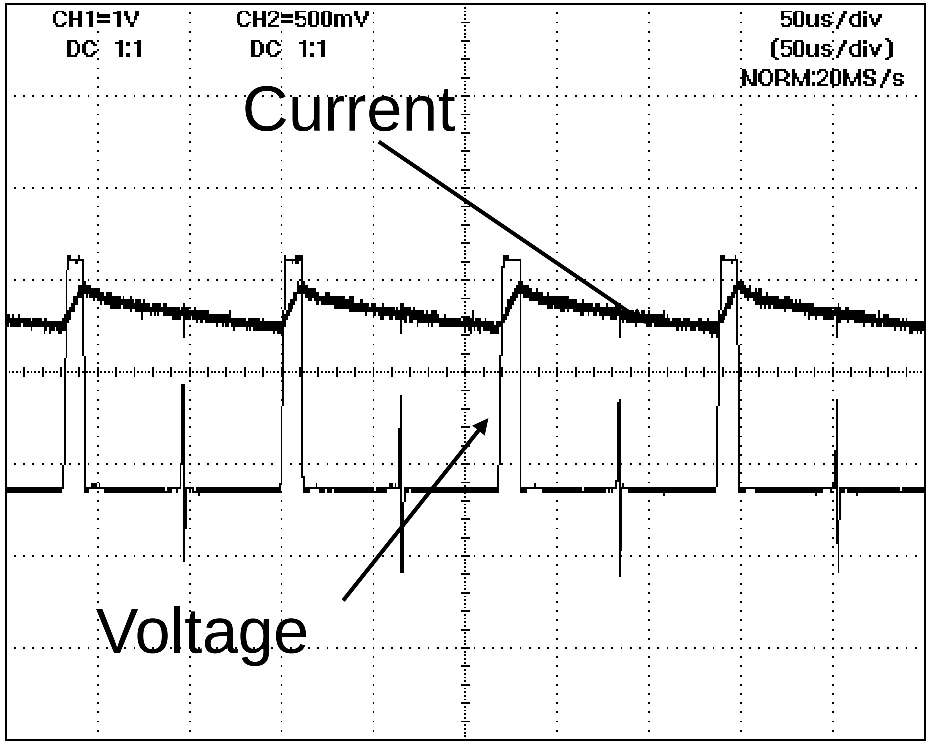
- When u is 1 (respectively 0), the voltage u_m at the motor terminals is Vp (resp. -Vp).
- Average value of u_m:

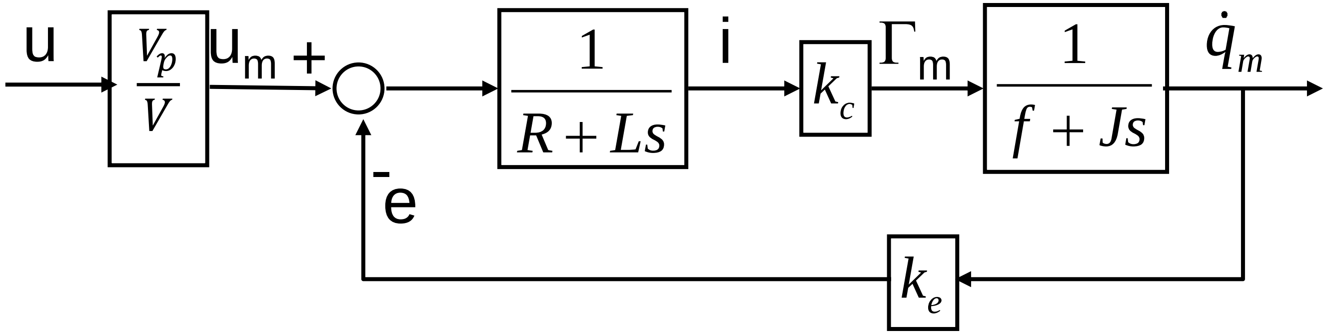
$$\bar{u}_m = \alpha V_p + (1-\alpha)(-V_p) = (2\alpha+1)V_p = \frac{V_p}{V}u$$

Example : electronic chopper

- Given the high frequency, u_m can be identified to with its average value

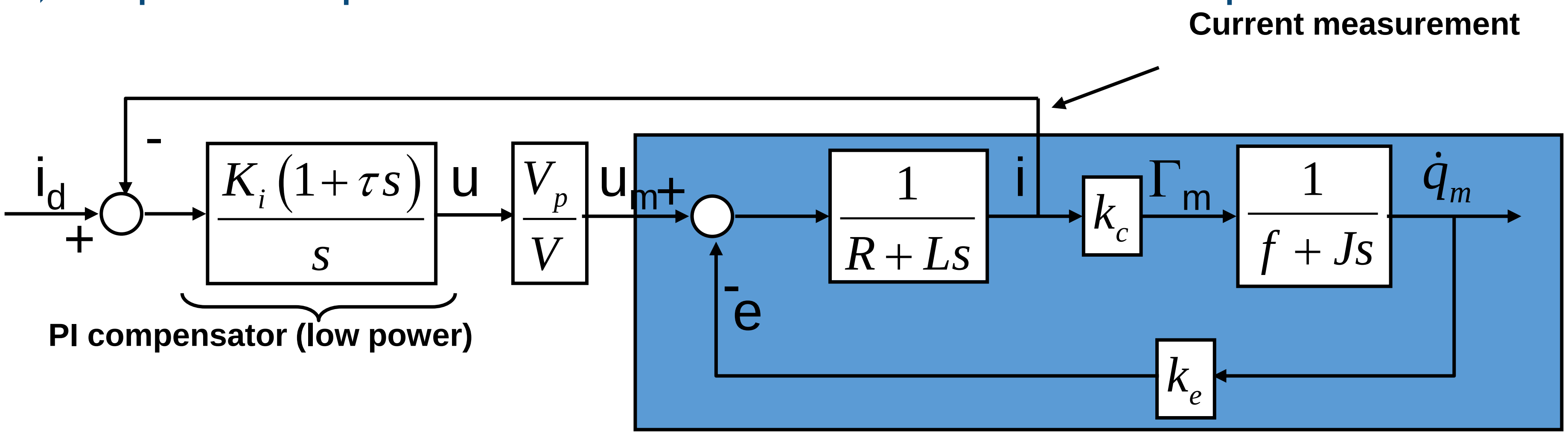
$$u_m \simeq \bar{u}_m = \frac{V_p}{V} u$$





Adding a current control loop

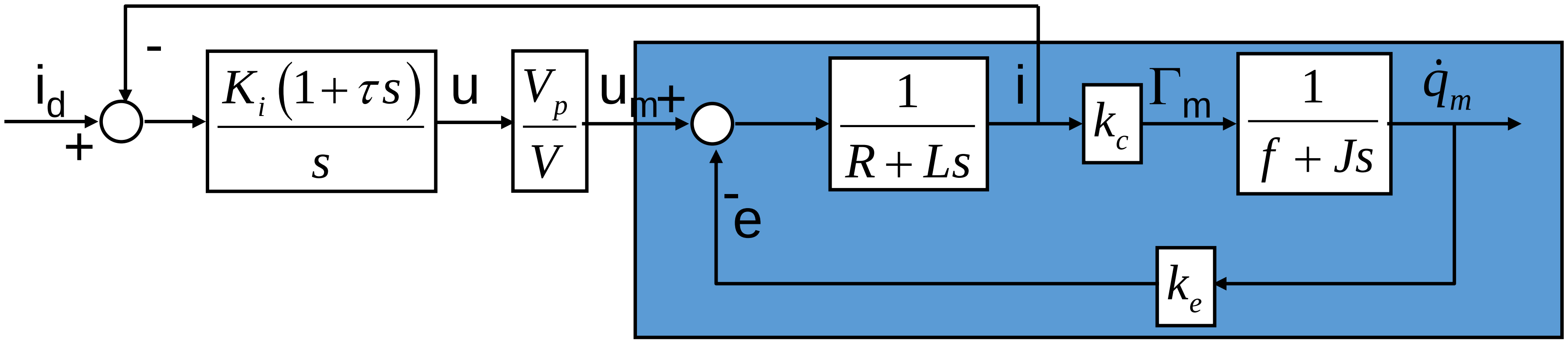
- In general, the power amplifier also carries out a current control loop:



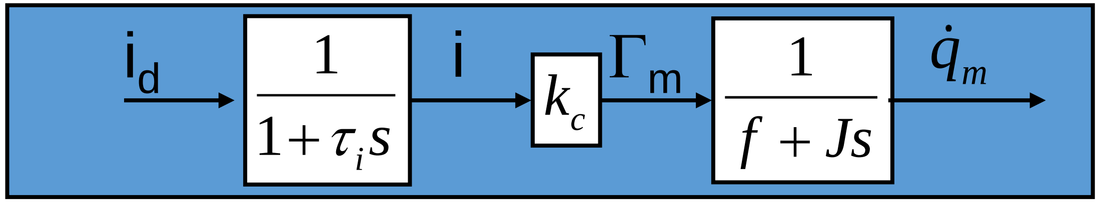
- Role:
  - Master the current in the motor (stay below I_{max})
  - Accelerate the application of a current (response time less than 1ms in general), while rejecting the disturbance of the bemf.
  - Control the state of the motor even if the cables of connection are long (thus with a resistance not insignificant compared to the engine).

Final model for a motor + power amp + current servo loop

Due to its high bandwidth the system:



is roughly equivalent to:

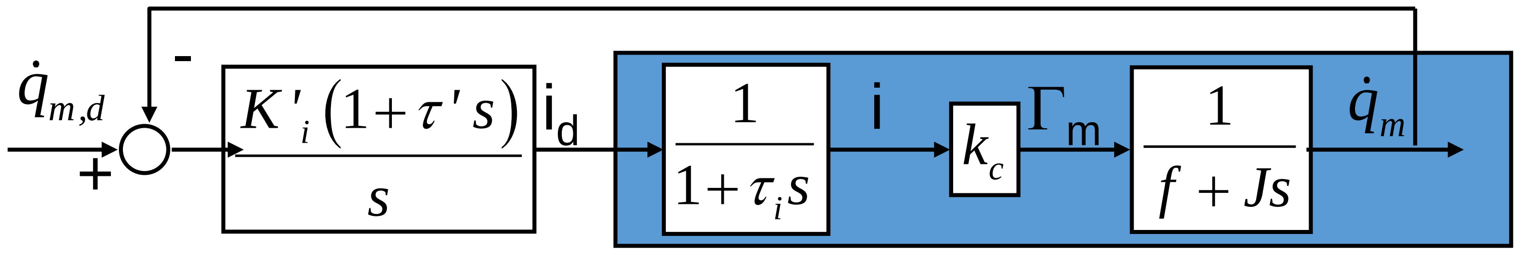


where, typically, τ_i < 1ms.

The current response is generally considered instantaneous, which is to say that one directly controls the motor torque.

Adding a low level velocity loop

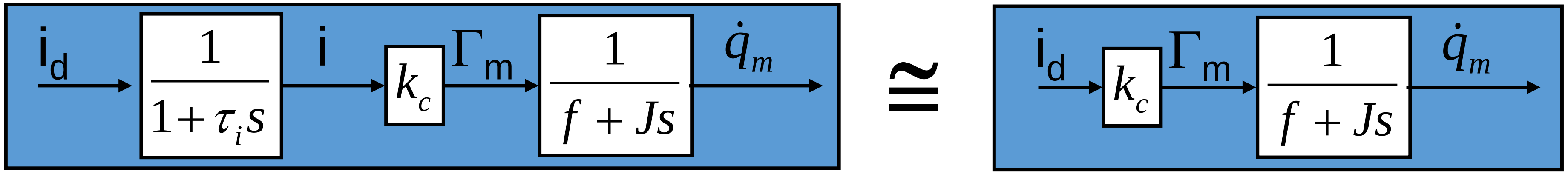
- Power amplifiers also often allow speed regulation, from a signal delivered by a velocity sensor :



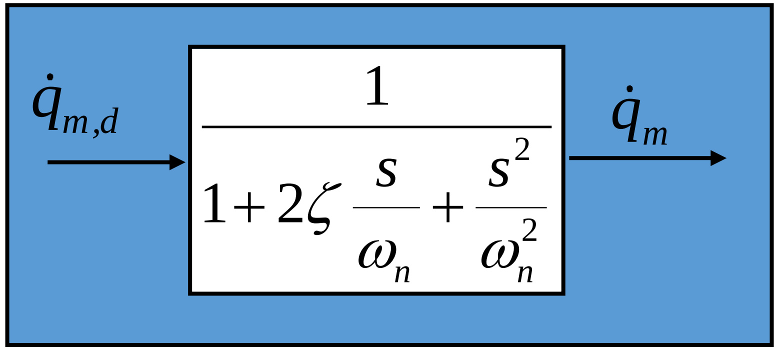
- Roles:
  - Allowing the controller direct access to robot speed (simpler control than a couple)
  - Linearizing the operation of the axis by rejecting disturbances due to dry friction or dynamic couplings
- Typical bandwidth in closed loop: from 15Hz to 60Hz.

FINALLY, FROM THE ROBOT CONTROLLER VIEWPOINT, TWO CASES

CASE 1 : the chopper has a current loop but no velocity loop, and then the system is called "torque controlled", with an open loop model of the following form:



CASE 2 : the chopper has, in addition to the current loop, a velocity loop, and the model is then:



Note that each of these two models assumes a 1DOF linear mechanics, of constant inertia, without backlash, without friction, without couplings.

GUIDANCE AND TRANSMISSION

Guidance System

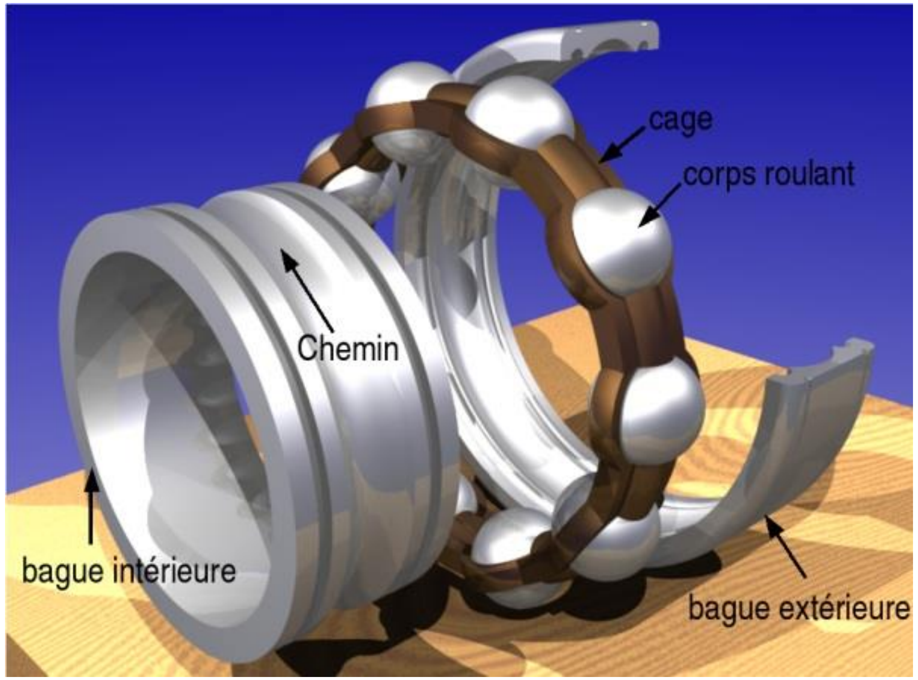
- Mechanical parts that constrain the movement of the mobile body with respect to the fixed body. Therefore, defines the joint type.
- A robot with guidance but no actuator, nor transmission is like a puppet.

Issues with guidance

- Precision (backlash, deformations).
- Friction that has to be minimized
  - By reducing the coefficient of friction between parts that slide against one another (choice of materials, surface finishes, lubrication)
  - By preventing sliding and prioritising rolling motion without slippage
- Load-bearing capacity in directions other than those of the joint the movement.

Ball bearings

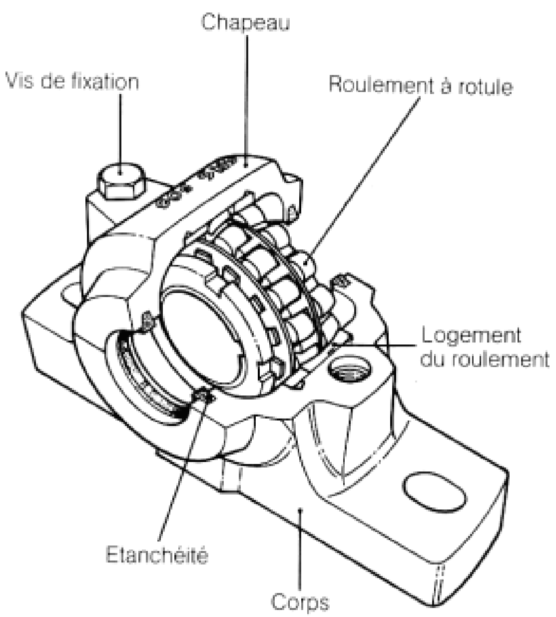
- A ball bearing guides a rotation



- Never use just one.

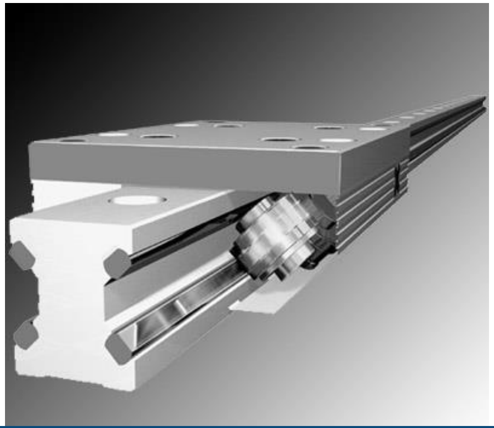
Many different types

- Ball bearings, roller bearings, needle bearings, spherical roller bearings, thrust ball bearings, etc., etc.



Linear guiding systems

- For linear motion.
- The same principles of friction reduction apply.
- Low friction sling parts or ball-bearing guide rails
- Cantilever issues.



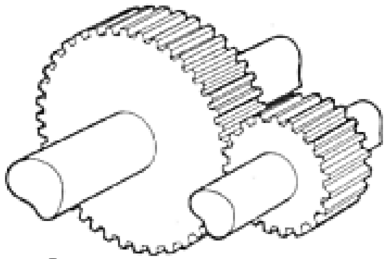
TRANSMISSION

- Most actuators (including electric motor) move *rapidly* but produce small generalized force (motor torque), compared to what a robot requires to move.
- One thus connects the motor to the robot moving body through a transmission system, most often a gear box.
- Principle :
  - The output velocity is devided by N : $\dot{q}_s = \frac{1}{N} \dot{q}_m$
  - Le output torque is multiplied by N : $\tau_s = N \tau_m$
  - N is the gear ratio
- This supposes that the gear box does not dissipates energy.

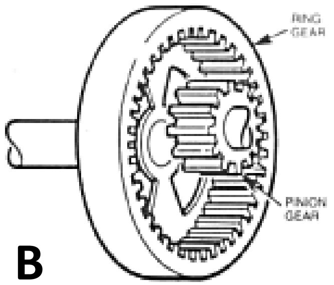
Pulleys and belts

Gears

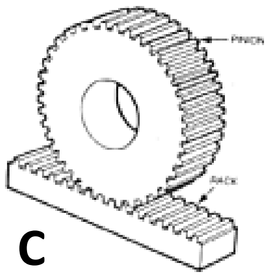
Straight cylindrical gears:
A: external,
B: internal,
C: pinion/rack



A



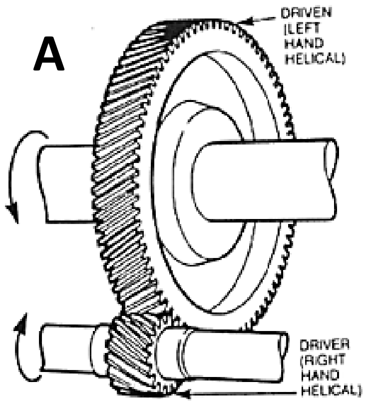
B



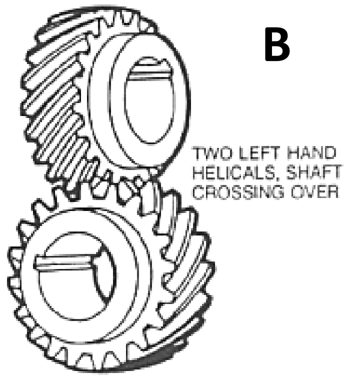
C

Helical gears:
A: parallel shafts,
B: perpendicular shafts,

A

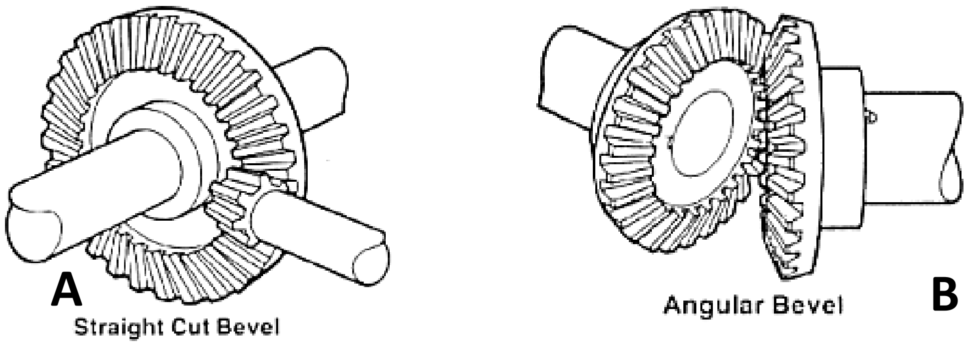


B

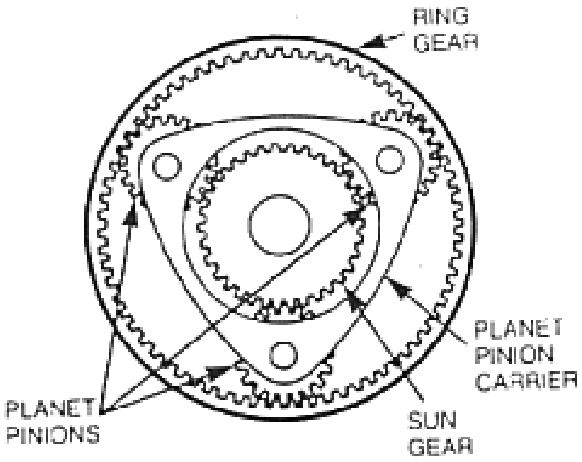


GEARS

Bevel gears:
A: right-angled (90 deg.),
B: any angle.



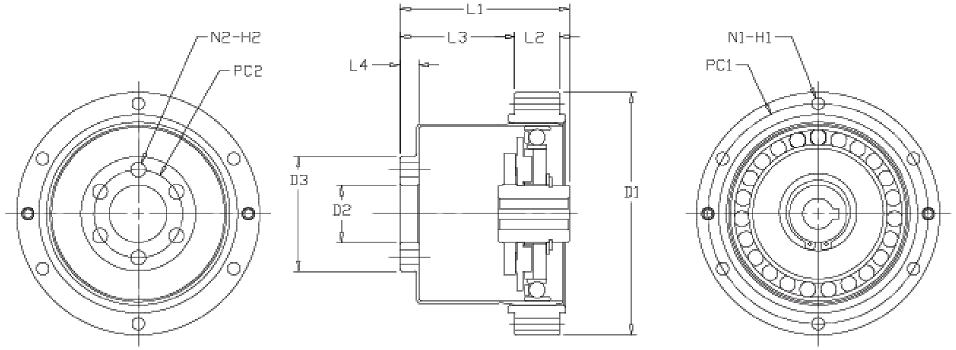
Planetary or epicycloidal gears

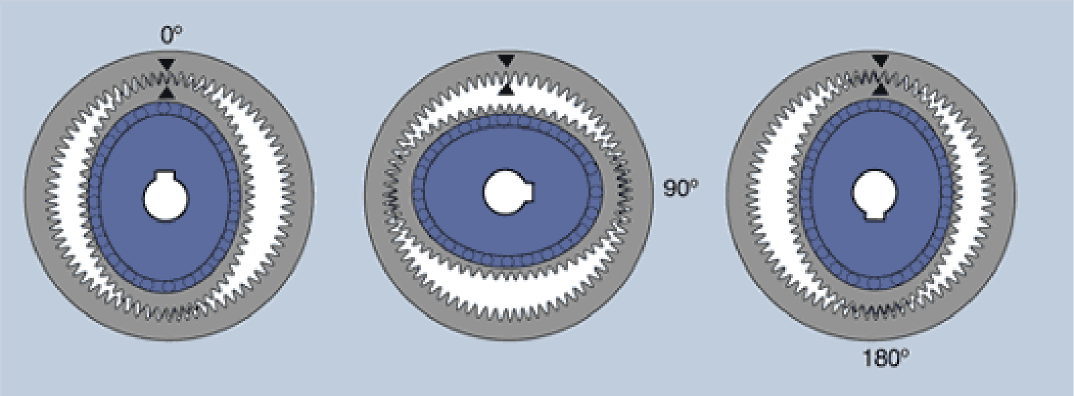


Harmonic Drive

"The teeth on the non-rigid Flexspline and the rigid Circular Spline are in continuous engagement. Since the Flexspline has two teeth fewer than the Circular Spline, one revolution of the input causes relative motion between the Flexspline and the Circular Spline equal to two teeth. With the Circular Spline rotationally fixed, the Flexspline rotates in the opposite direction to the input at a reduction ratio equal to one-half the number of teeth on the Flexspline."

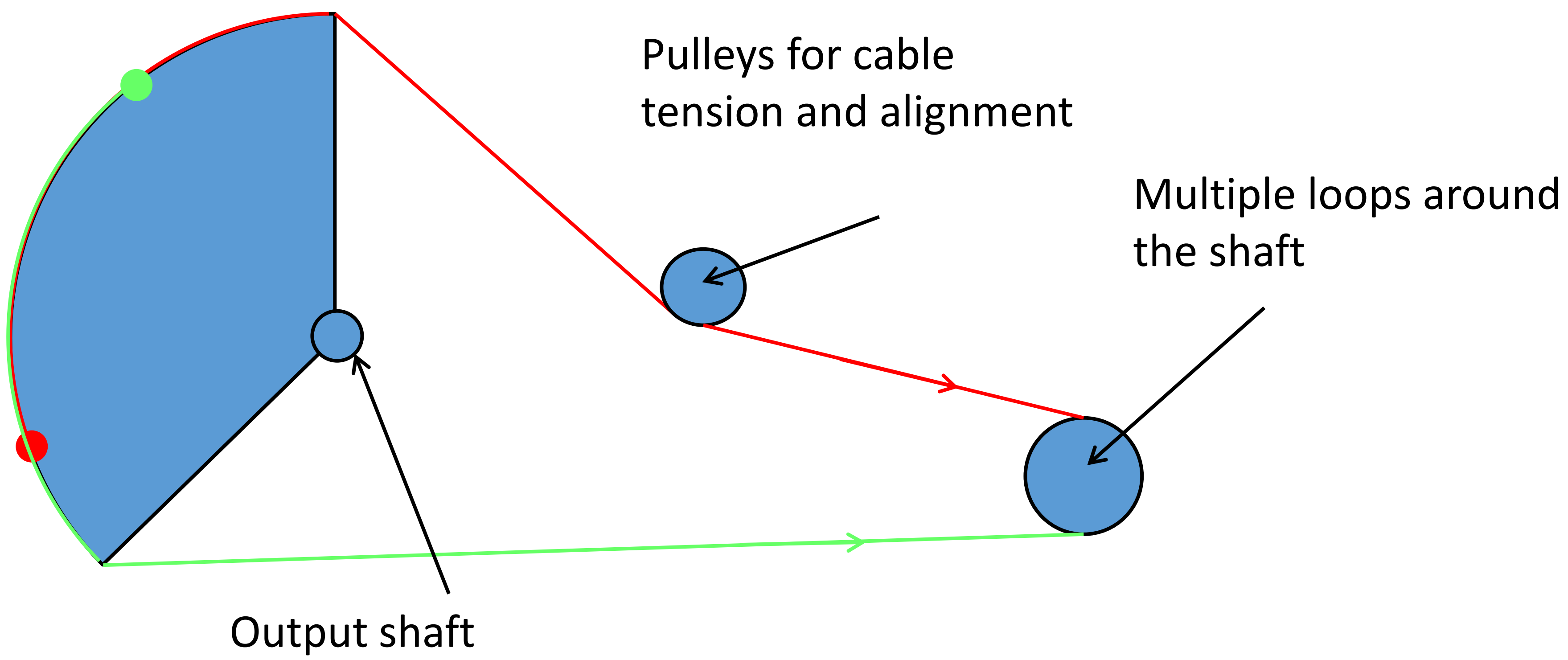
In: http://www.harmonic-drive.com/





Ball screws

Cable transmission



Example: WAM robot

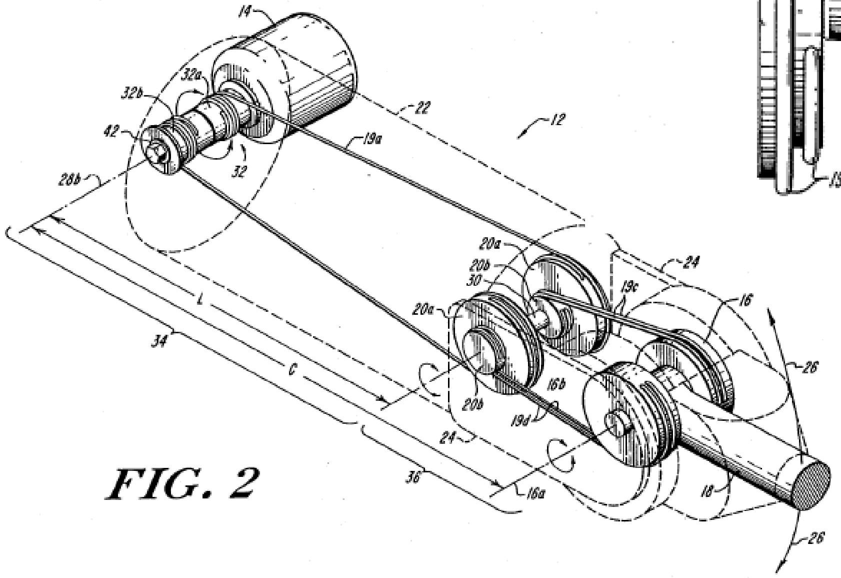
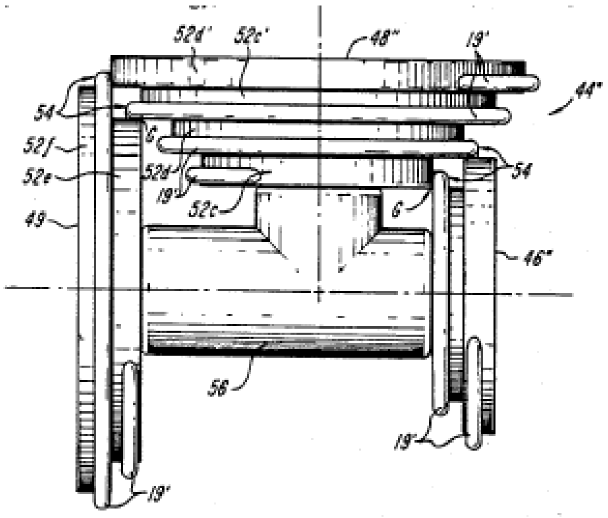


FIG. 2



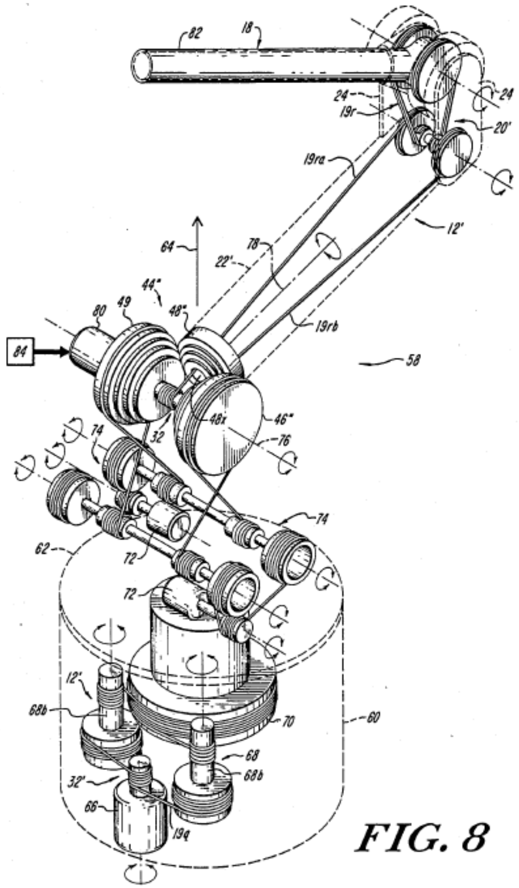
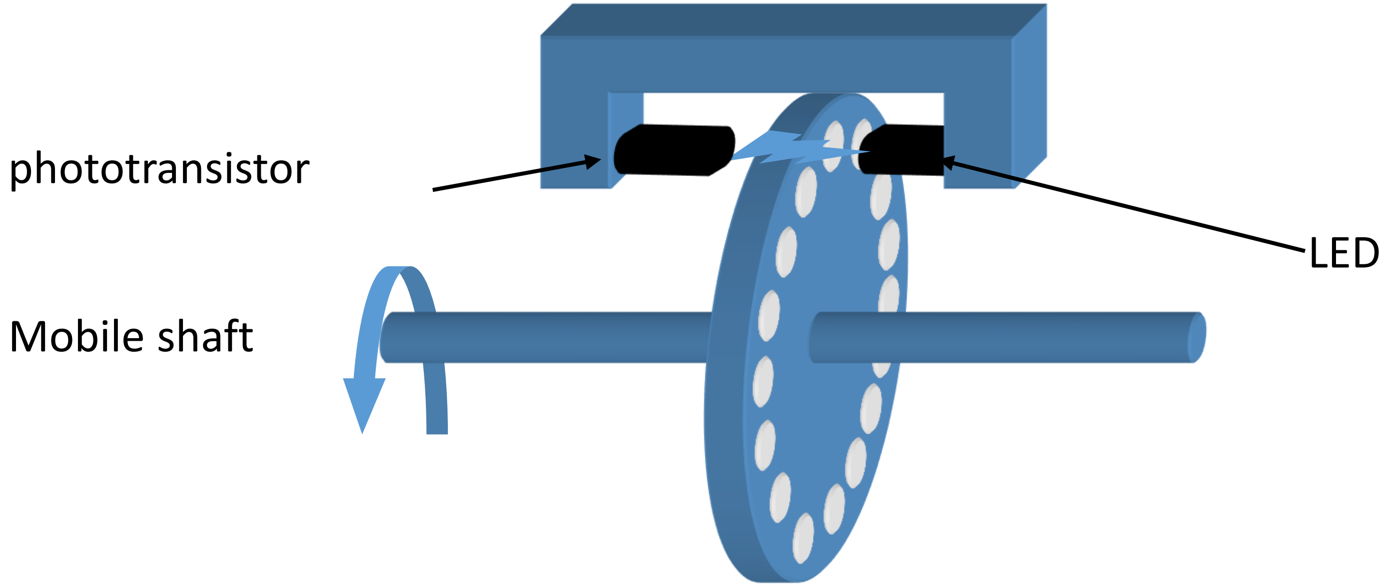


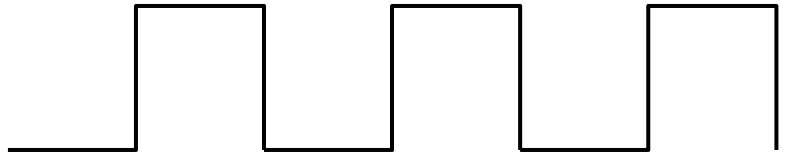
FIG. 8

Joint sensors

Optical encoders: principle

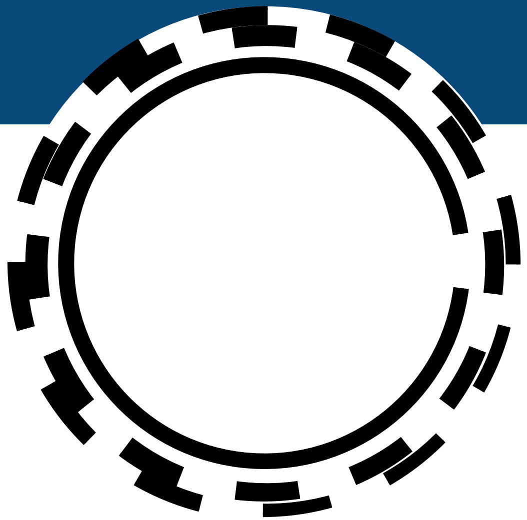


When the shaft rotates, a square signal is produced

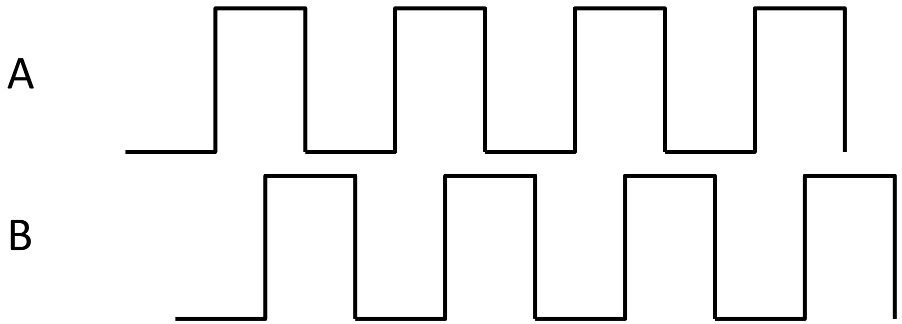


Measuring rotational displacement = counting impulses

Incremental joint encoders

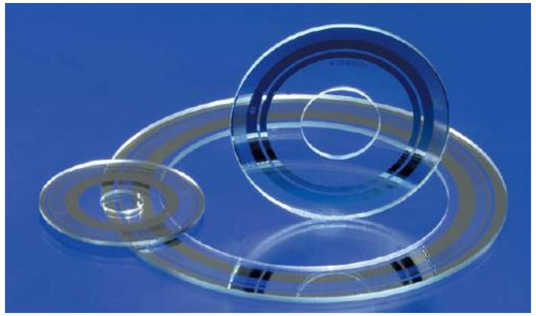


- Two channels with a fourth period shift
- For each one, one phototransistor.
- Signals

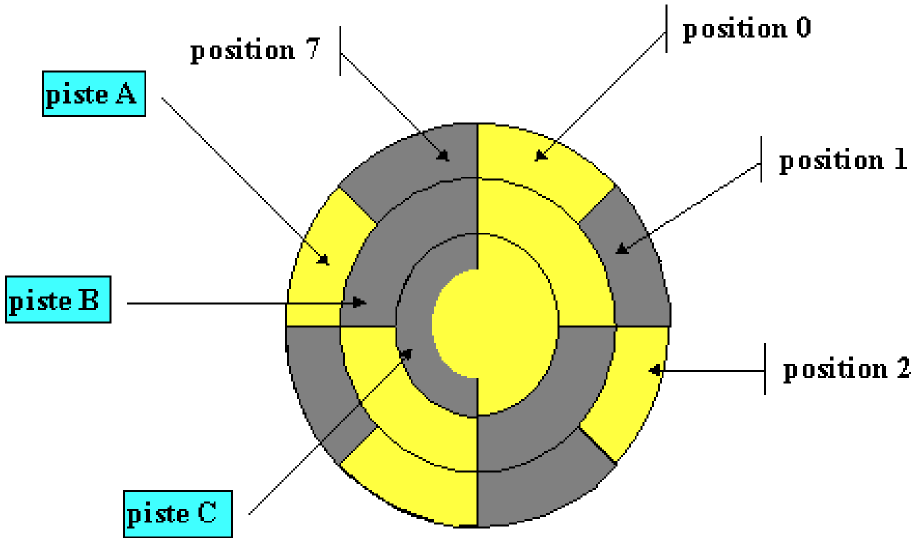


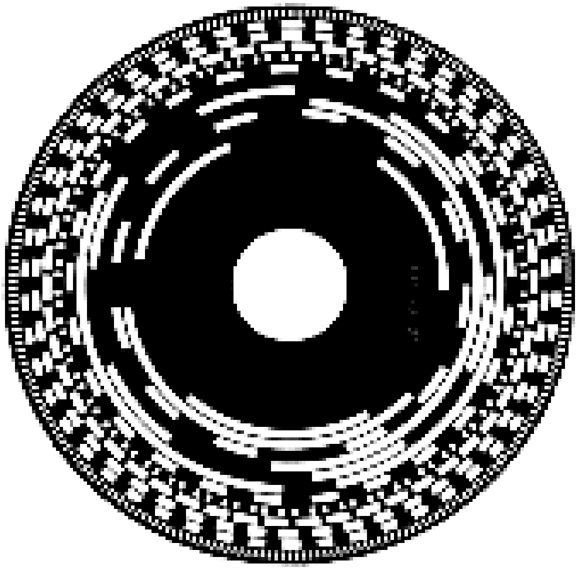
Phase shift encodes the direction of rotation

- Adding top signal: one per revolution, used to initialize the joint measure



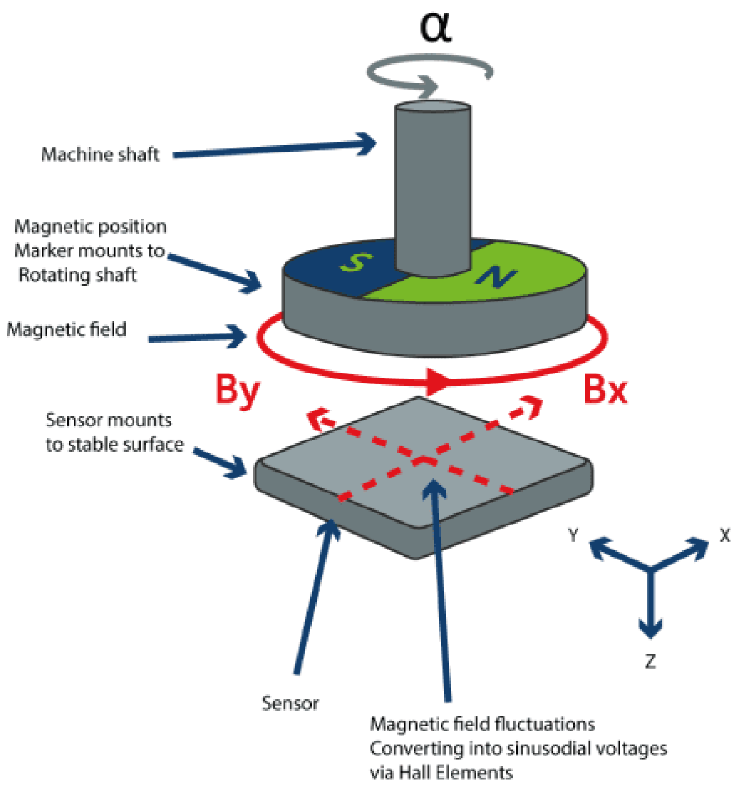
Absolute optical encoders





- More cumbersome
- More wires (not a negligible issue in robotics)

Hall effect sensors

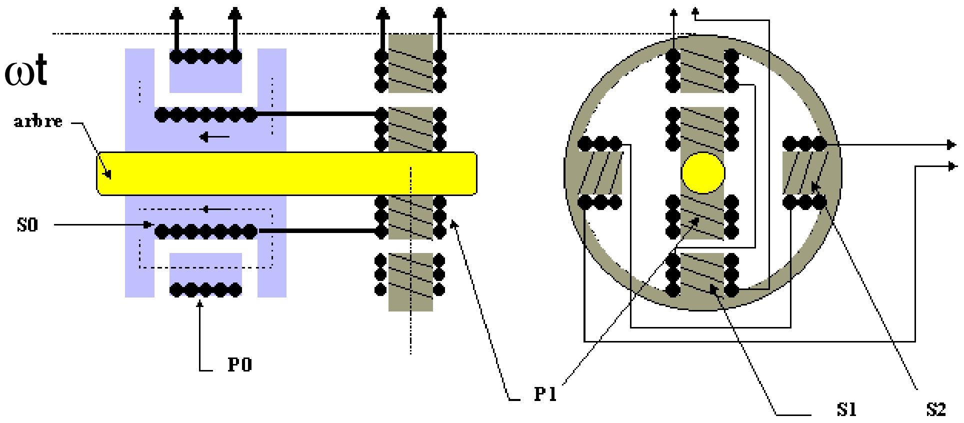


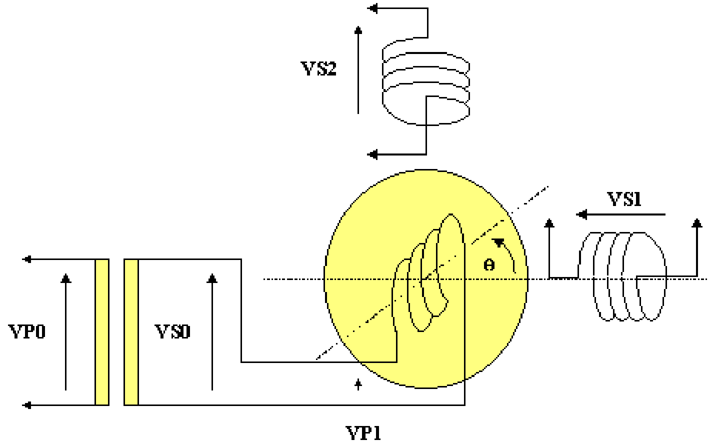
Resolvers

1. The stator primary coil P_0 receives $U_0 \sin\omega t$..
2. The rotor secondary coil S_0 then gets $K_0\, U_0 \sin\omega t$ (independently from the joint position).
3. S_0 feeds P_1 on the rotor with $K_0\, U_0 \sin\omega t$. P_1 is perpendicular to the joint.
4. Two coils S_1 and S_2 , at the stator, are perpendicular and radial, such that they get:

$$V_{S_1} = k \cos\theta \sin\omega t$$
$$V_{S_2} = k \sin\theta \sin\omega t$$

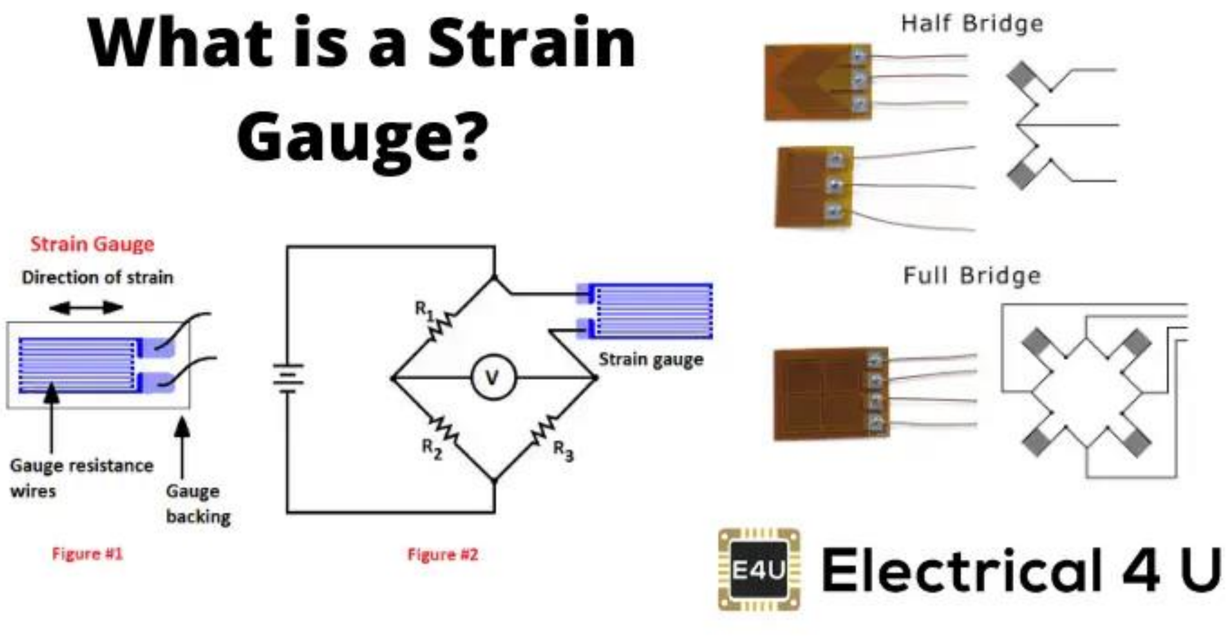
5. After demodulation, one can recover for $\cos\theta$ and $\sin\theta$, then θ

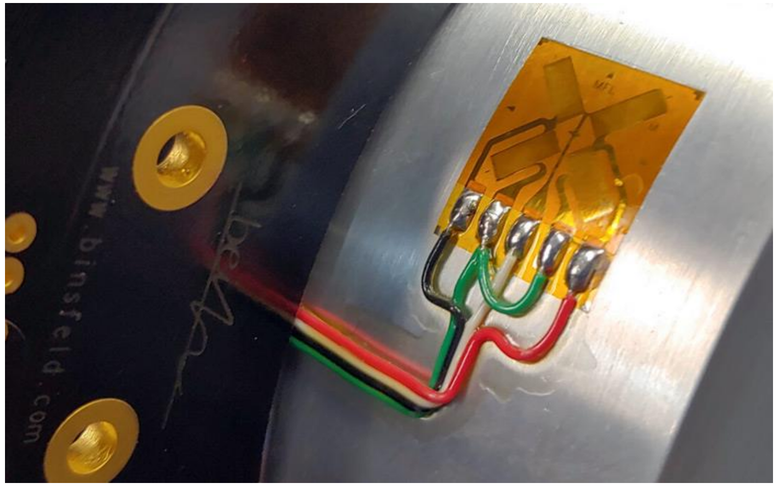




Joint torque sensors

- Based on strain gauges





Sensor placement

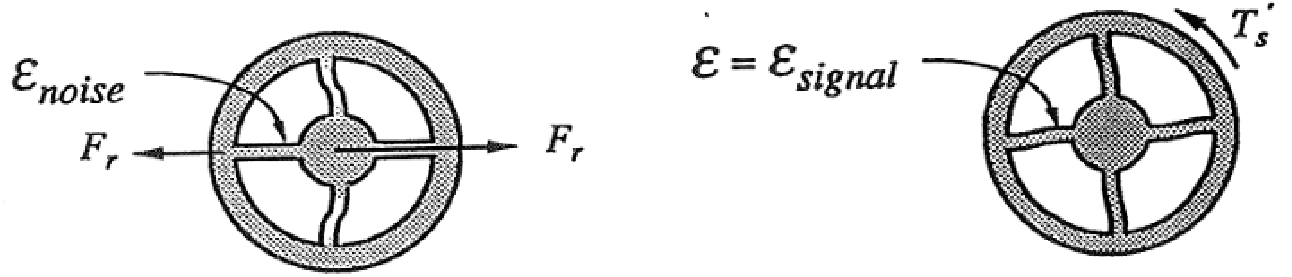


Figure 13: Signal-to-Noise Ratio

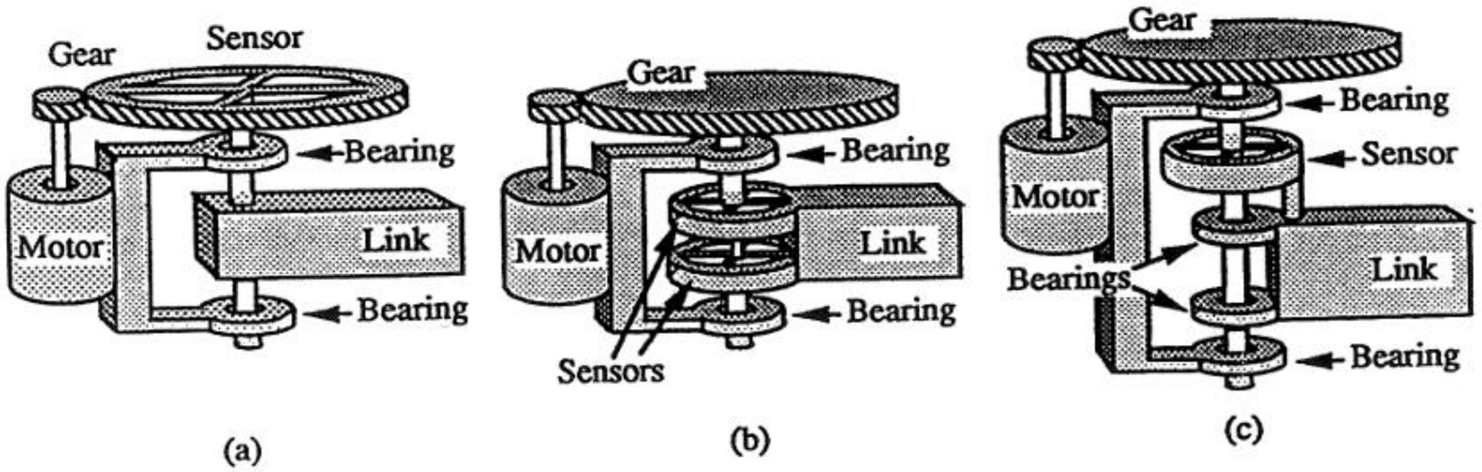
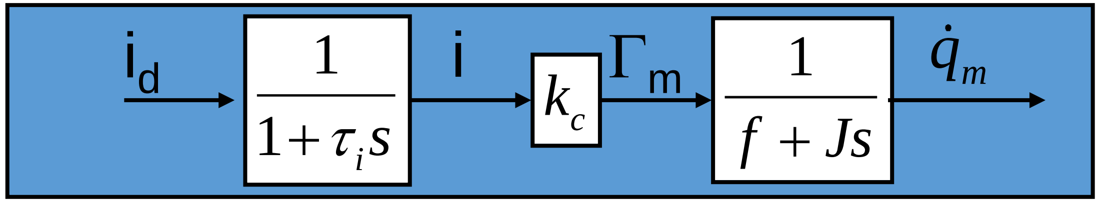


Figure 12: Sensor Placement

Modelling a joint with actuator + transmission

- Example:
  - Brushless DC motor.
  - Power amplifier with current loop
  - Gearbox with gear ratio N.
- The actuator model



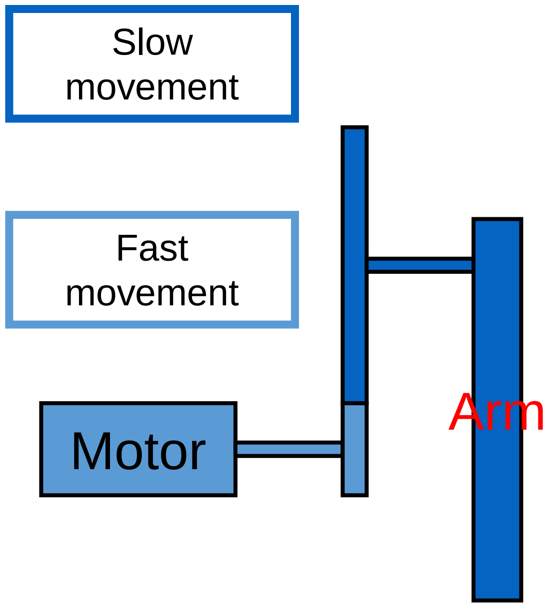
shall be completed with the transmission and arm model

Ideal gearbox

- Rigid
- No backlash
- No friction

$$\dot{q}_s = \frac{1}{N}\dot{q}_m$$

$$\tau_s = N\,\tau_m$$



Fast moving dynamics

$$\Gamma_m - \Gamma_t - f_m\dot{q}_m = (J_m + J_{r1})\ddot{q}_m$$

Slow moving dynamics

$$N\Gamma_t - f_s\dot{q}_s = (J_s + J_{r2})\ddot{q}_s + g(q_s)$$

$$\Gamma_t = \frac{1}{N}\left(f_s\frac{\dot{q}_m}{N} + (J_s + J_{r2})\frac{\ddot{q}_m}{N} + g(q_s)\right)$$

$$\Gamma_m - \left(\frac{f_s}{N^2} + f_m\right)\dot{q}_m - \frac{g(q_s)}{N} = \left(J_m + J_{r_1} + \frac{J_s + J_{r2}}{N^2}\right)\ddot{q}_m$$

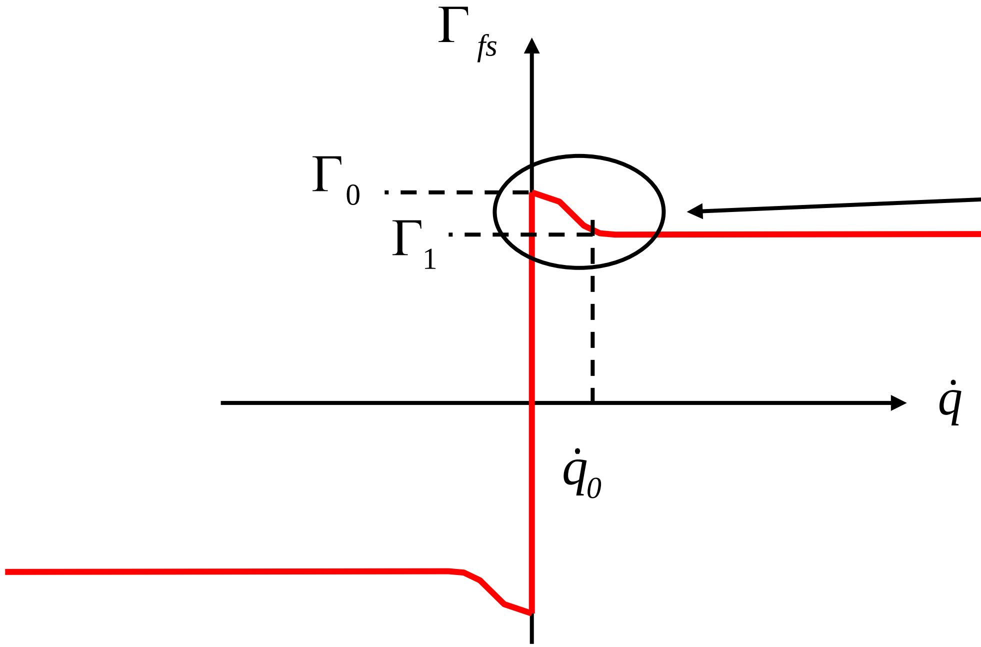
$$N\Gamma_m - (f_s + N^2 f_m)\dot{q}_s - g(q_s) = \left(N^2\left(J_m + J_{r_1}\right) + J_s + J_{r2}\right)\ddot{q}_s$$

NONLINEAR FRICTION

- So far, friction was modeled as purely viscous
- When the transmitted torque Γ_t is low, the joint does not move (dry friction Γ_{fs})
- A simple (non linear) model writes:
  - If $\Gamma_t < \Gamma_0$ then $\dot{\mathrm{q}} = 0$ and $\Gamma_{fs} = \Gamma_t$
  - Otherwise : $\Gamma_{fs} = \Gamma_0$
- Another form writes: $\Gamma_{fs} = \Gamma_0 \,\mathrm{sgn}(\dot{\mathrm{q}})$. However, it does not provide the friction torque when $\dot{q} = 0$
- Combined with viscous friction this leads to:

$$\Gamma_f = \Gamma_0\, sgn(\dot{q}) + f\dot{q}$$

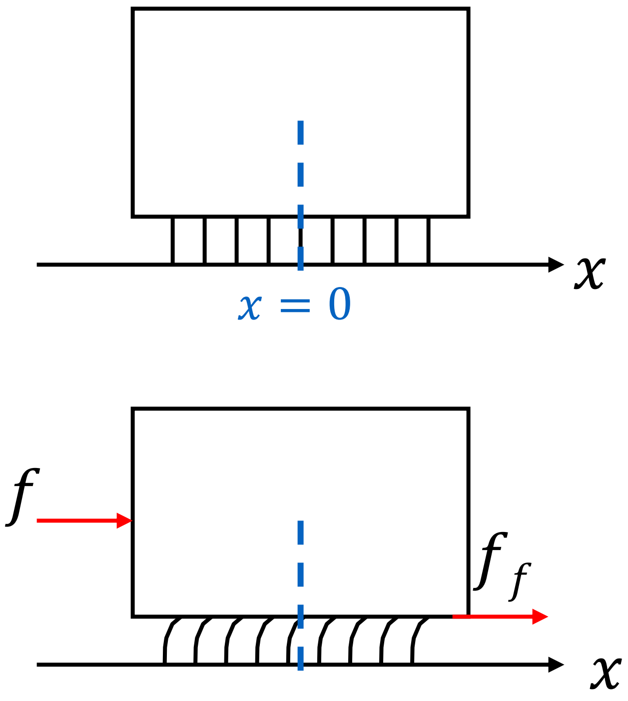
STRIBECK EFFECT



At low velocities, a negative slope is observed. This is this origin of so-called stick-slip phenomenon, leading to problems in controlling the movement at low velocity

Friction modelling at low velocities

Principle: distinguish between 2 deformations: an elastic one (z, reversible) and a plastic one (w, non reversible)



$$x = z + w$$

z = presliding, cancelled if $f = 0$
S'annule si on annule f.

w = sliding, persists when f drops to zero.

The LUGRE model (1995)

$$x = z + w$$

$$\begin{cases} f_f = \sigma_0 z + \sigma_1 \dot{z} \; (+\sigma_2 \dot{w}) \\ \dot{z} = \dot{x}\left(1 - \dfrac{\sigma_0}{\left|f_{ss}(\dot{x})\right|} sgn(\dot{x})\, z\right) \end{cases}$$

- σ_0 is a stiffness
- Presliding is damped (σ_1)
- Stribeck effect modelled by f_{ss}.

Joint flexibility

Fast parts dynamics

$$\Gamma_m - \Gamma_t - f_m \dot{q}_m = \left(J_m + J_{r1}\right)\ddot{q}_m$$

Slow parts dynamics after the spring

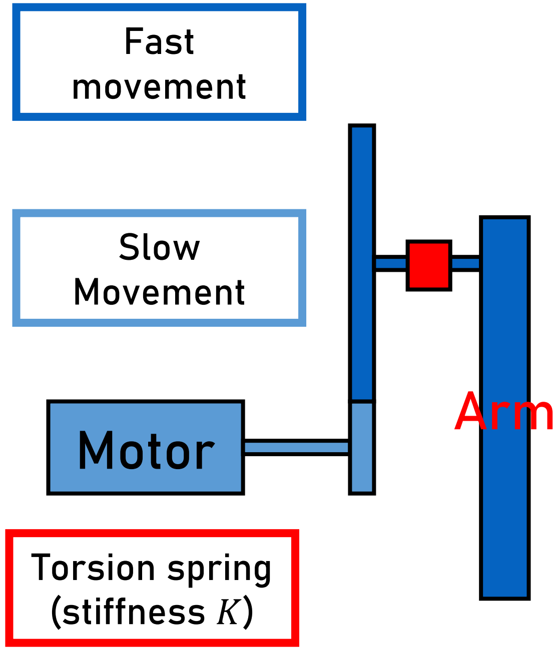
$$\Gamma_r - f_s \dot{q}_s = J_s \ddot{q}_s + g(q)$$

Slow parts dynamics before the spring

$$N\Gamma_t - \Gamma_r = J_{r2}\ddot{q}_i$$

Spring

$$\Gamma_r = K\left(q_i - q_s\right)$$



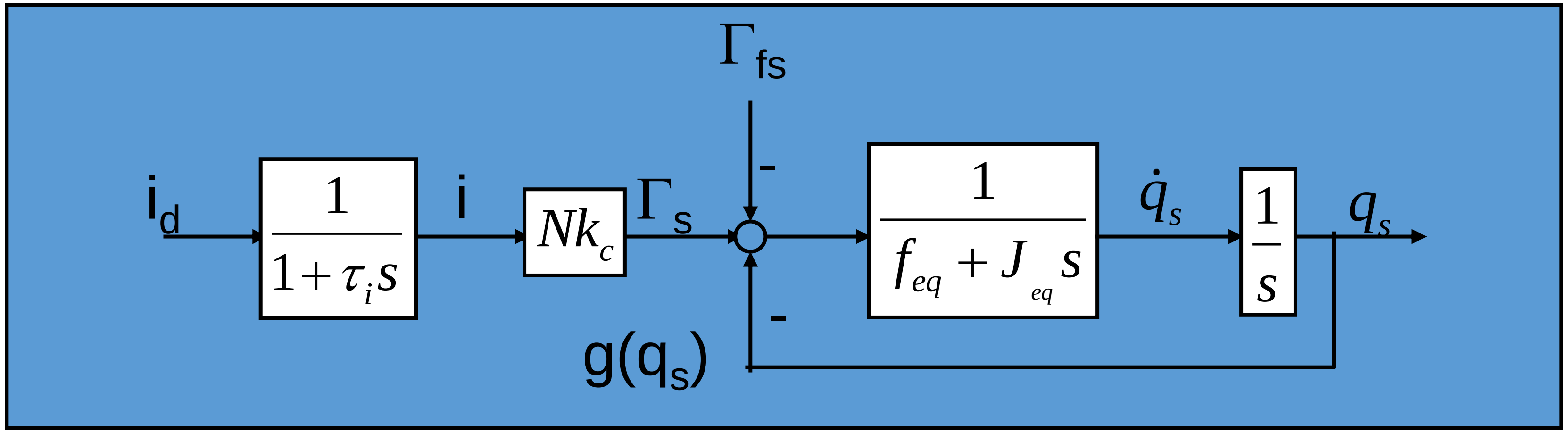
One state variable must be added

<u>**In practice:**</u> first flexible modes are at a few Hertz, a rigid model can be assumed if the joint position control bandwidth remain lower than this.

A POPULAR MODEL IN THE ROBOT CONTROL LITERATURE

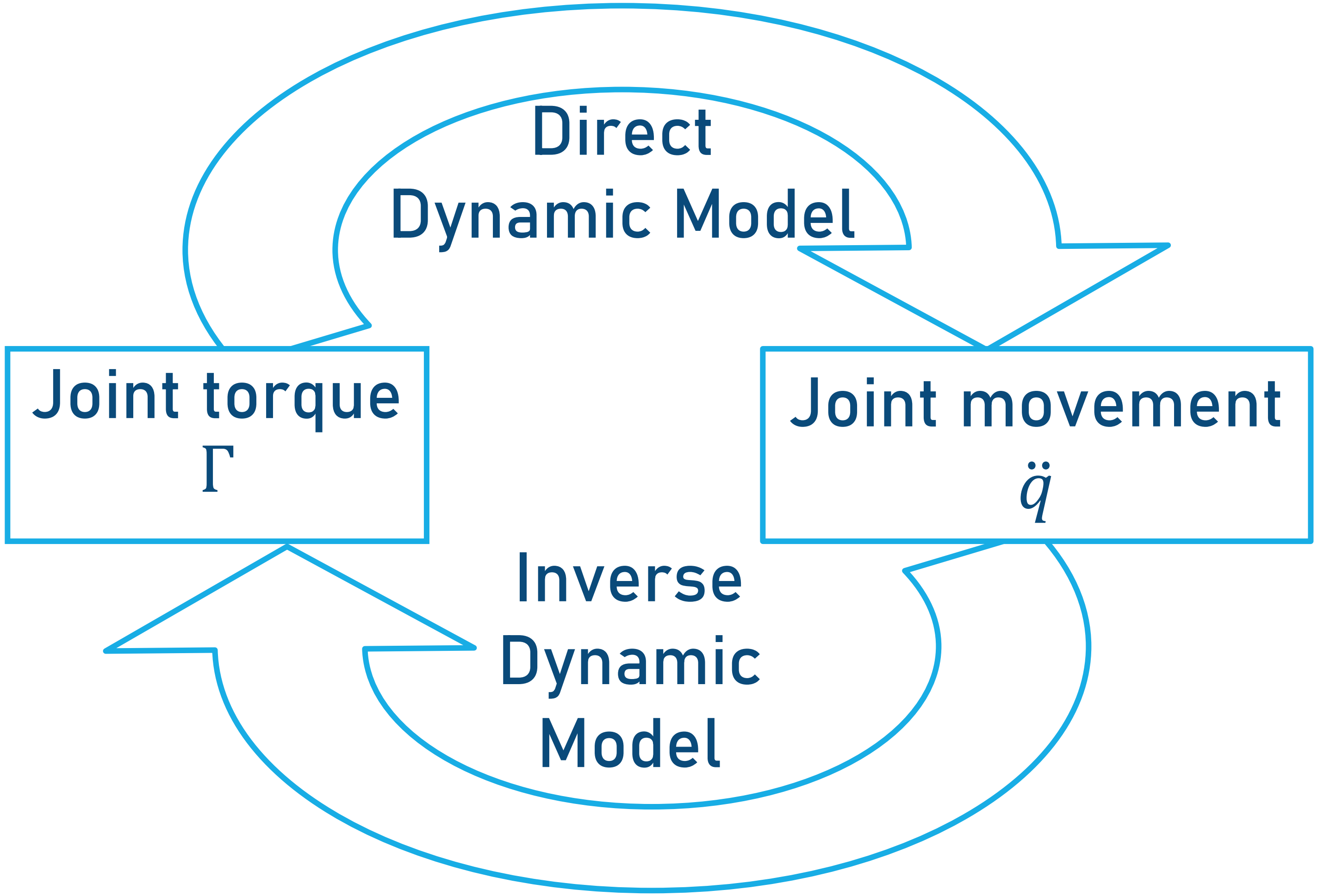
$$N\Gamma_m = \underbrace{\left(N^2\left(J_m + J_{r_1}\right) + J_s + J_{r2}\right)}_{J_{eq}}\ddot{q}_s + \underbrace{\left(f_s + N^2 f_m\right)}_{f_{eq}}\dot{q}_s + \Gamma_{fs} + g(q_s)$$

$$\Gamma_m = K_c i_m$$



1.2 Multi-joint models used in robot control

DYNAMICS MODELLING



From Sebastien Briot's talk

Differentiating the Lagrangian, we obtain

$$\tau = \mathbf{M}_t(\mathbf{q}_t)\ddot{\mathbf{q}}_t + \mathbf{C}(\mathbf{q}_t, \dot{\mathbf{q}}_t)\dot{\mathbf{q}}_t + \mathbf{f}_g(\mathbf{q}_t) \quad (19)$$

where:

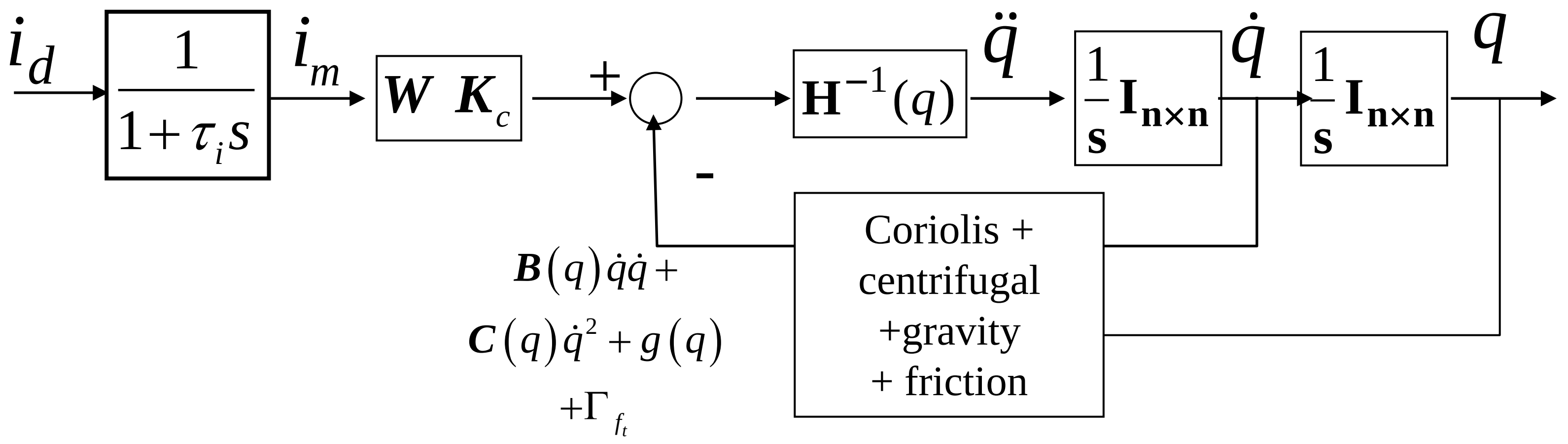
- τ is a $(n \times 1)$ vector of the robot input efforts
- $\mathbf{C}(\mathbf{q}_t, \dot{\mathbf{q}}_t)\dot{\mathbf{q}}_t$ is the $(n \times 1)$ vector of Coriolis and centrifugal torques, such that:

$$\mathbf{C}(\mathbf{q}_t, \dot{\mathbf{q}}_t)\dot{\mathbf{q}}_t = \dot{\mathbf{M}}_t\dot{\mathbf{q}}_t - \frac{\partial E}{\partial \mathbf{q}_t}$$

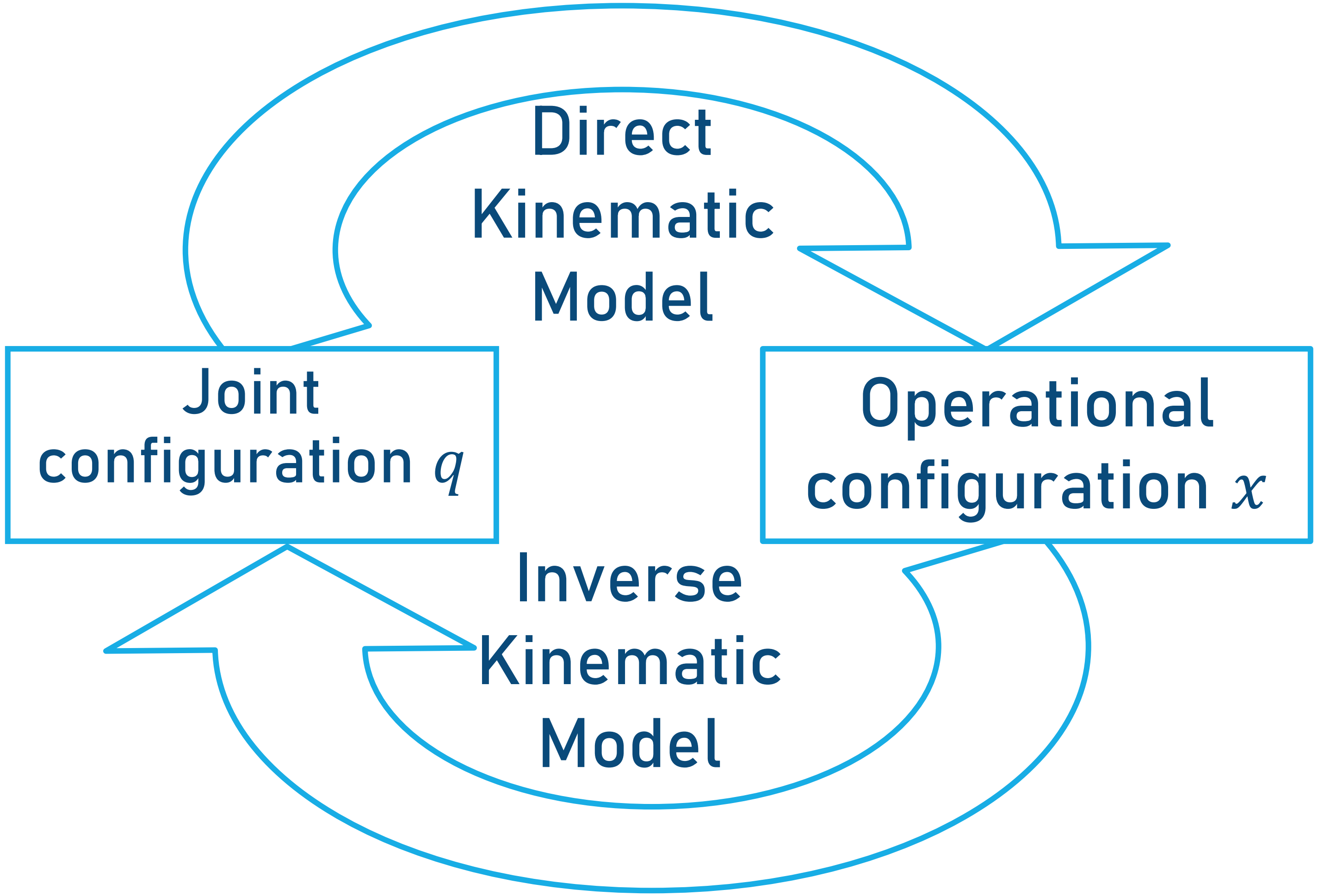
The dynamics are coupled

$$\tau = \mathbf{M}_t(\mathbf{q}_t)\ddot{\mathbf{q}}_t + \mathbf{Ia}\,\ddot{\mathbf{q}}_t + \mathbf{C}(\mathbf{q}_t, \dot{\mathbf{q}}_t)\dot{\mathbf{q}}_t + \mathbf{f}_g(\mathbf{q}_t)$$

$$\mathrm{H}(q)\ddot{q} = \mathrm{W}\ \mathrm{K}_c i_m - \mathrm{B}(q)\dot{q}\dot{q} - \mathrm{C}(q)\dot{q}^2 - g(q) - \Gamma_{f_t}, \quad \text{with :} \begin{cases} \mathrm{H}(q) = \mathrm{A}(q) + \mathrm{W}\ (\mathrm{J}_m + \mathrm{J}_{r1})\mathrm{W} + \mathrm{J}_{r2} \\ \Gamma_{f_t} = \mathrm{W} f_m \mathrm{W}\dot{q} + f_s\dot{q} + \Gamma_{fs} \end{cases}$$



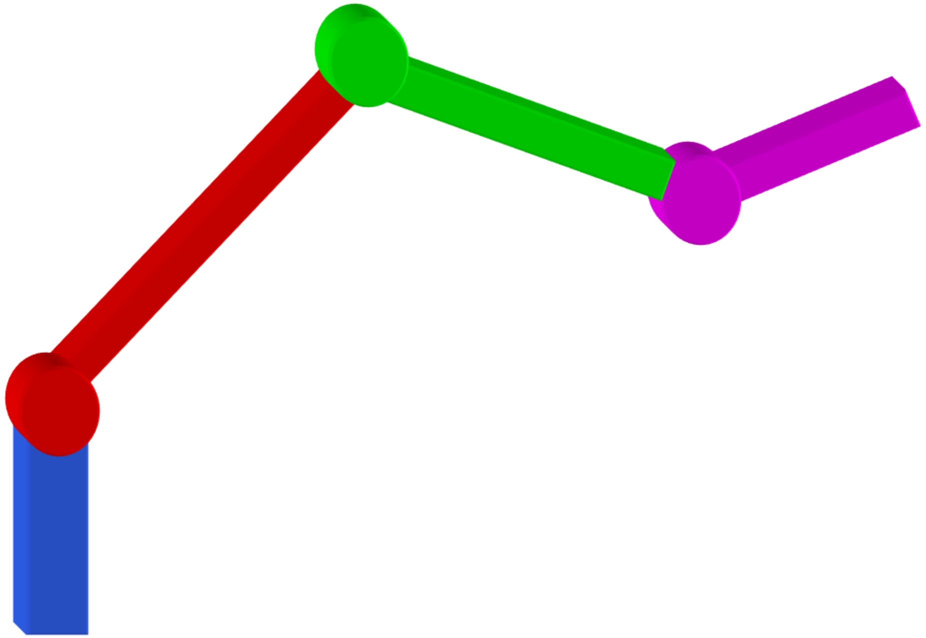
Kinematic modelling



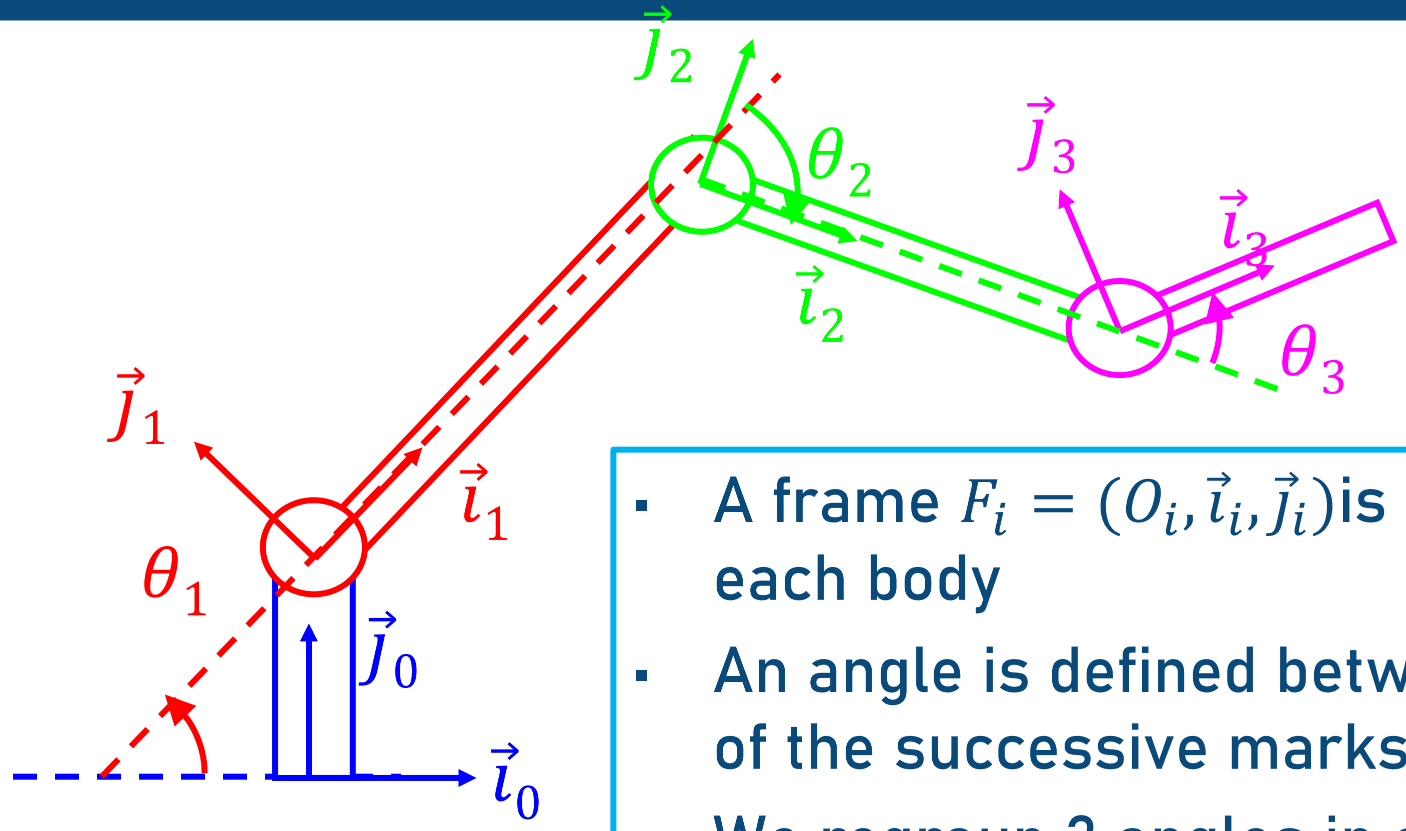
Example

How to:

- Define q ?
- Define x ?
- Compute $x = f(q)$?
- Compute $q = f^{-1}(x)$?

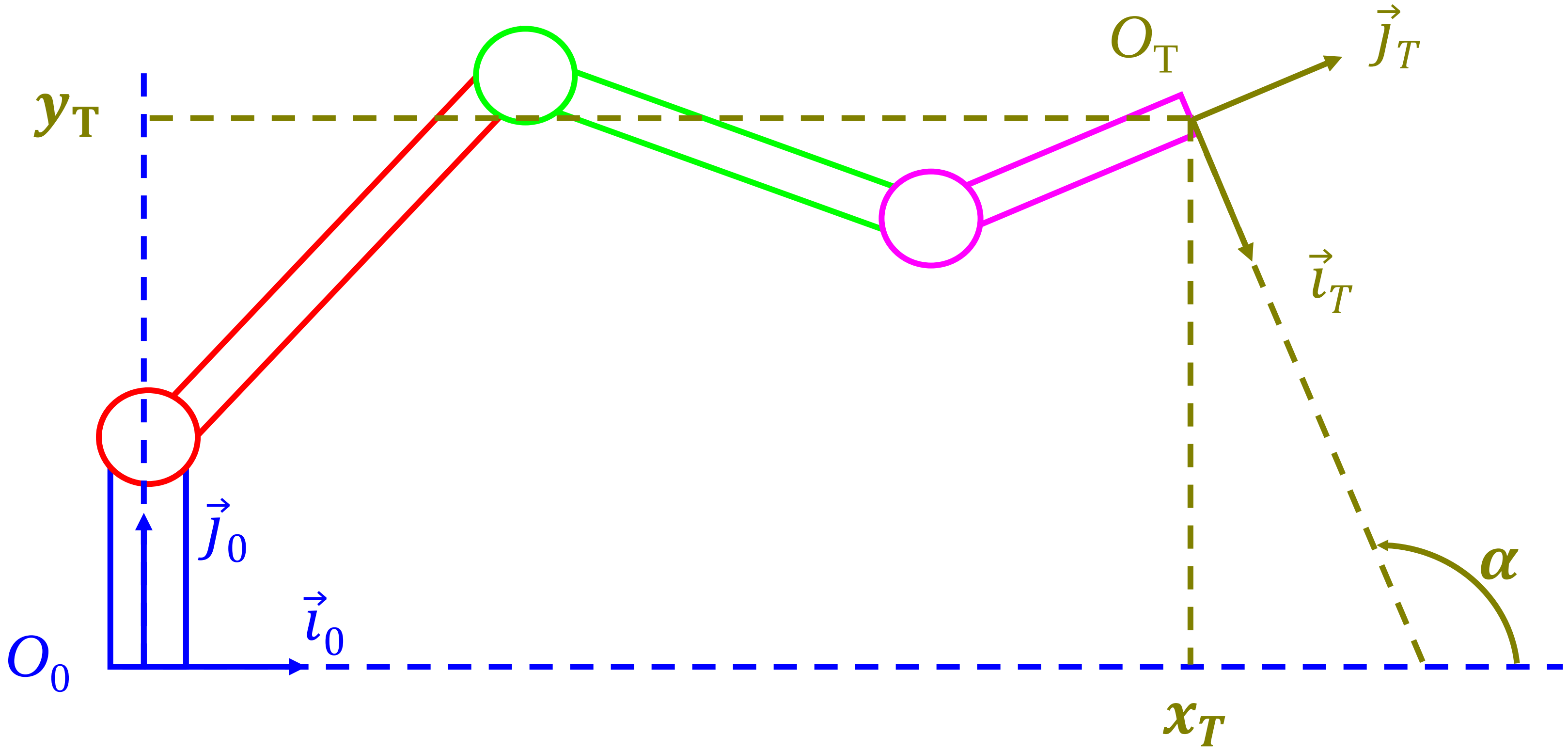


JOINT PARAMETERS (q)



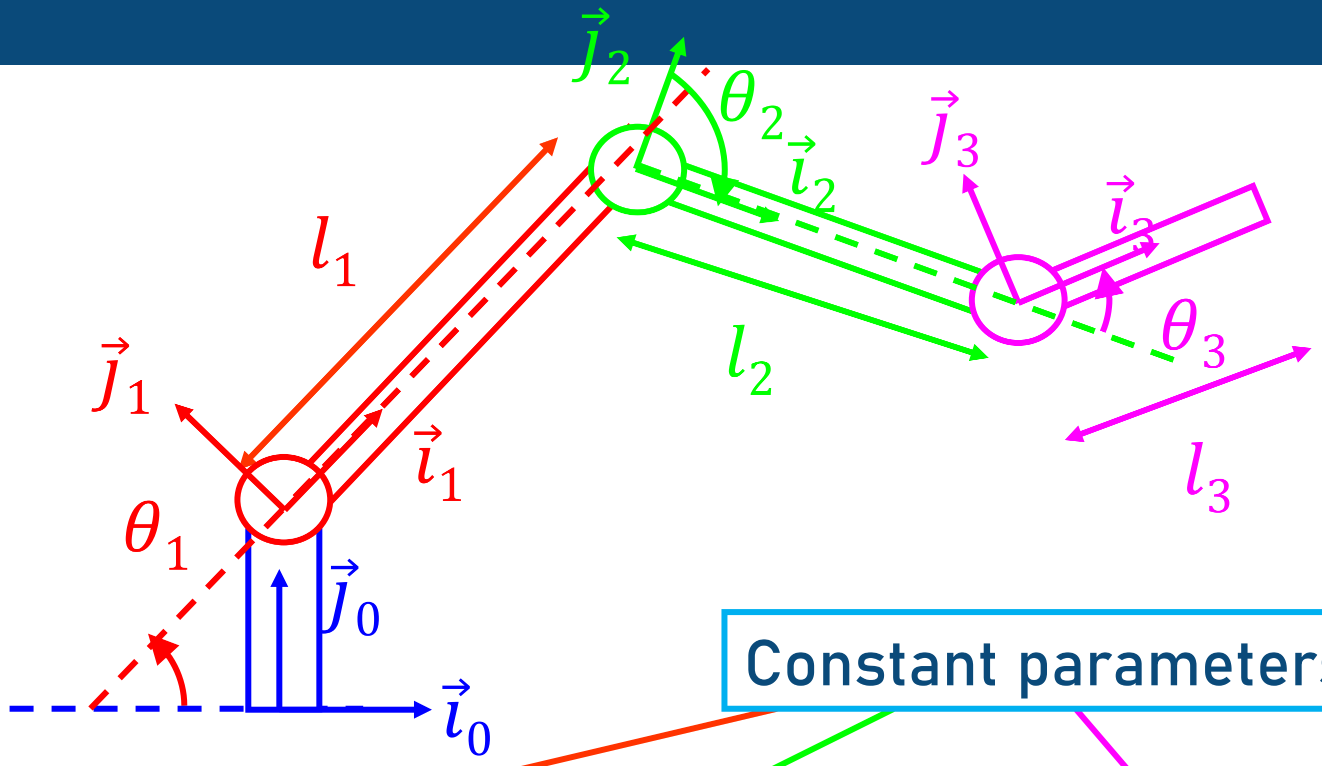
- A frame $F_i = (O_i, \vec{i}_i, \vec{j}_i)$ is attached to each body
- An angle is defined between the axes of the successive marks
- We regroup 3 angles in a vector of joint parameters: $q = [\theta_1\ \theta_2\ \theta_3]^T$

Operational parameters (x)



$$x = [x_T \; y_T \; \alpha]^T$$

Direct kinematic model



Constant parameters

$$x = \begin{bmatrix} x_T \\ y_T \\ \alpha \end{bmatrix} = \begin{bmatrix} l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}$$

From Hélène Chanal's talk

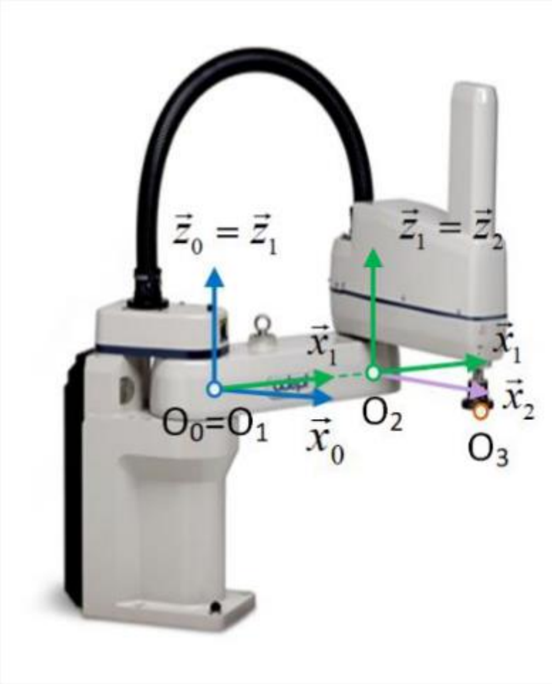
Kinematic modeling

- Example: 2 first joints of SCARA:
  - DKM:

$$e^{\hat{\omega}_i q_i} = \mathbf{I} + \hat{\boldsymbol{\omega}}_i \sin(q_i) + \hat{\boldsymbol{\omega}}_i^2 (1 - \cos(q_i)) = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 \\ \sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

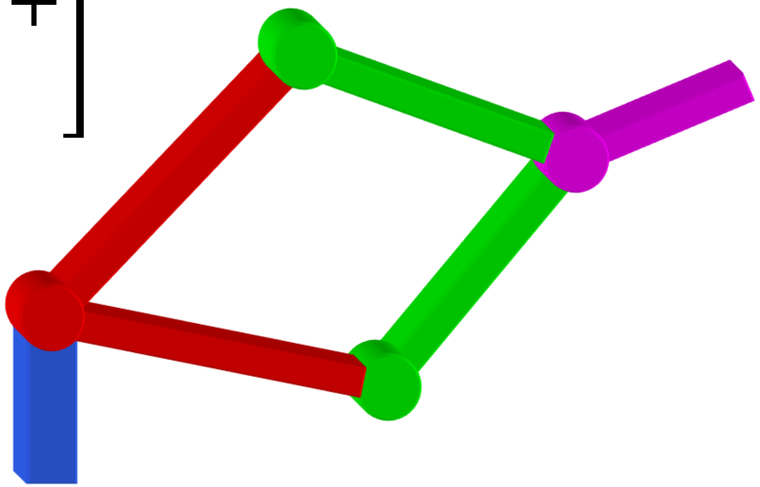
$$\boldsymbol{t}_1 = \left(\mathbf{I} - e^{\hat{\omega}_1 q_1}\right)(\boldsymbol{\omega}_1 \times \boldsymbol{v}_1) + \boldsymbol{\omega}_1 \boldsymbol{\omega}_1^T \boldsymbol{v}_1 q_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{t}_2 = \begin{bmatrix} -a_1(\cos(q_2) - 1) \\ -a_1 \sin(q_2) \\ 0 \end{bmatrix}$$

$$\boldsymbol{g}_d = \begin{bmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 & a_2 \cos(q_1 + q_2) + a_1 \cos(q_1) \\ \sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 & a_2 \sin(q_1 + q_2) + a_1 \sin(q_1) \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Inverse kinematic model

$$\begin{bmatrix} x' \\ y' \\ \alpha \end{bmatrix} = \begin{bmatrix} x_T - l_3 \cos\alpha \\ y_T - l_3 \sin\alpha \\ \alpha \end{bmatrix} = \begin{bmatrix} l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}$$



$$\begin{cases} \cos\theta_2 = \dfrac{x'^2 + y'^2 - l_1^2 - l_2^2}{2 l_1 l_2} \\ \theta_1 = \operatorname{atan2}(y', x') - \operatorname{atan2}(l_1 \sin\theta_2, l_1 + l_2 \cos q_2) \\ \theta_3 = \alpha - \theta_1 - \theta_2 \end{cases}$$

Inverse kinematic model

- No general and systematic method
- In general, no unique solution.
- When a formal expression is wanted, one usually proceed "manually". When possible, decouple orientation from position (see e.g. Pieper algorithm).
- One usually proceed to a numerical inversion.

Numerical inversion

$$q_{sol} = \underset{q}{\operatorname{argmin}} \left(\left(\operatorname{dist}(x, f(q)) \right)^2 \right)$$

Example solution = Newton-Euler algorithm, whose skeleton is:

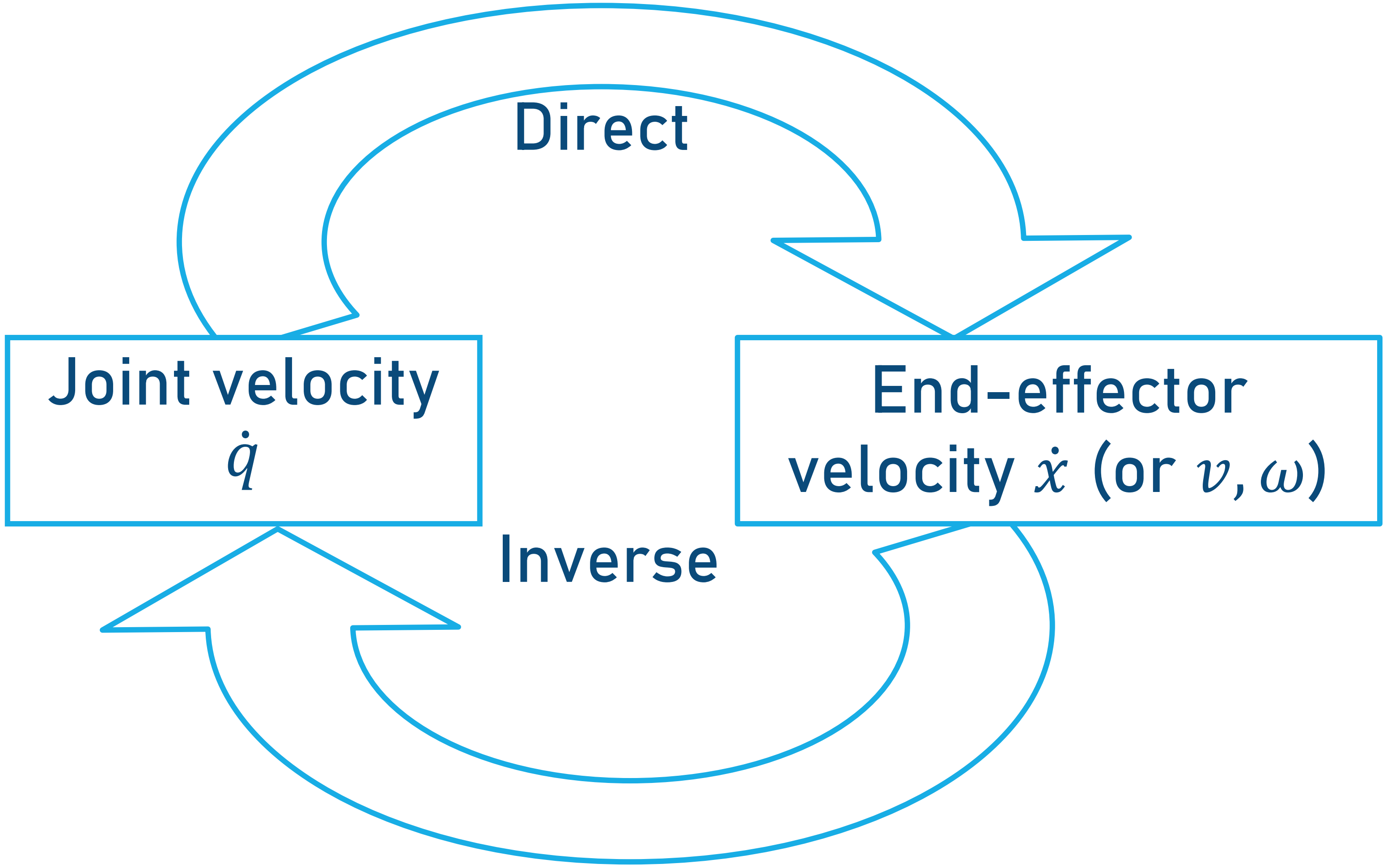
1. $q_{sol} \leftarrow q_0$ (initial best guess)
2. While $\|x - f(q_{sol})\| > \varepsilon$ do:
   - $\delta q \leftarrow \lambda J^{-1}(q).[x - f(q_{sol})]$
   - $q_{sol} \leftarrow q_{sol} + \delta q$

where $J(q)$ is the jacobian matrix of $f(q)$ whose element (i, j) is defined by $\frac{\partial f_i}{\partial q_j}$ and λ is a gain.

Note that the algorithm can be coded in such a way that the convergence is guaranteed (e.g. adapting λ).

Of course, only one local solution is found (but frequently, from one solution, one can compute all of them).

Velocity transmission model



VELOCITY TRANSMISSION MODEL

- Defined either by differentiating the kinematic model:

$$\dot{x} = \left(\frac{\partial f_i}{\partial q_j}\right)\dot{q} =: J(q)\dot{q}$$

- Or by velocity composition at endpoint O_n (natural Jacobian):

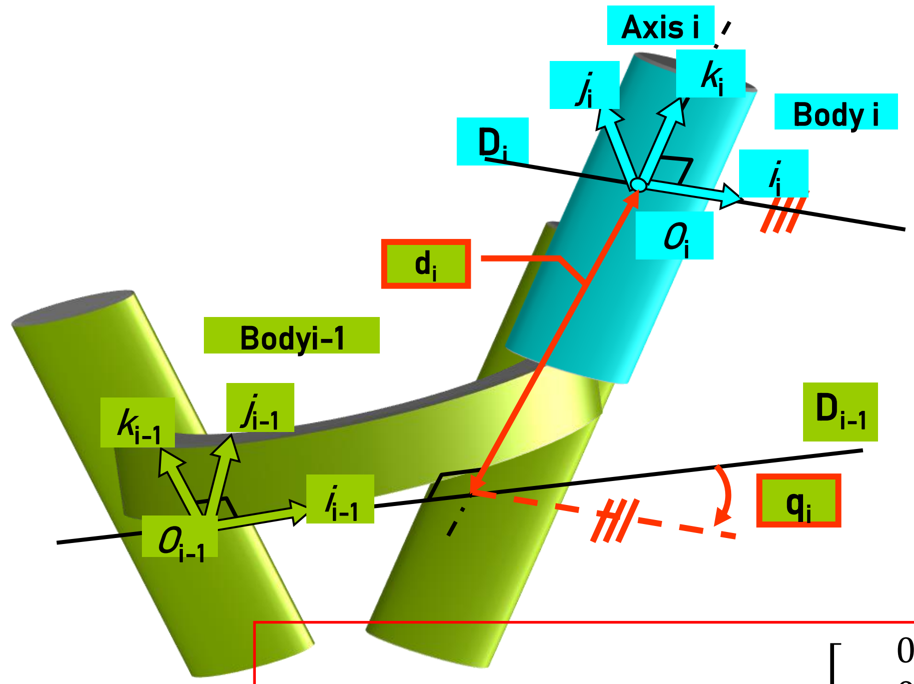
$$\begin{pmatrix} {}^0\omega_{n/0} \\ {}^0v^{O_n}_{n/0} \end{pmatrix} = J_{NE}(q)\dot{q}$$

- These two jacobian matrices are linked by the parameterization jacobian:

$$\dot{x} = J_{Px}(x)\begin{pmatrix} {}^0\omega_{n/0} \\ {}^0v^{O_n}_{n/0} \end{pmatrix} \Rightarrow J(q) = J_{Px}\big(f(q)\big)J_{NE}(q)$$

METHOD: VELOCITY COMPOSITION

a) Velocities between two successive joints of a robot.



Rotational Joint	Prismatic Joint
$\begin{Bmatrix} \vec{\omega}_{i/i-1} \\ \vec{v}^{O_i}_{i/i-1} \end{Bmatrix} = \begin{Bmatrix} \dot{\theta}_i \vec{k}_i \\ \vec{0} \end{Bmatrix}$	$\begin{Bmatrix} \vec{\omega}_{i/i-1} \\ \vec{v}^{O_i}_{i/i-1} \end{Bmatrix} = \begin{Bmatrix} \vec{0} \\ \dot{d}_i \vec{k}_i \end{Bmatrix}$
$\begin{Bmatrix} {}^i\omega_{i/i-1} \\ {}^iv^{O_i}_{i/i-1} \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_i \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{Bmatrix} {}^i\omega_{i/i-1} \\ {}^iv^{O_i}_{i/i-1} \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dot{d}_i \end{bmatrix}$

Homogeneously: $\begin{Bmatrix} {}^i\omega_{i/i-1} \\ {}^iv^{O_i}_{i/i-1} \end{Bmatrix} = \dot{q}_i \begin{bmatrix} 0 \\ 0 \\ \delta_i \\ 0 \\ 0 \\ (1-\delta_i) \end{bmatrix}$

With:

$\begin{cases} \delta_i = 1 \\ q_i = \theta_i \end{cases}$ for a R joint

$\begin{cases} \delta_i = 0 \\ q_i = d_i \end{cases}$ for a P joint

VELOCITY COMPOSITION

b) Composition of velocities : $\begin{Bmatrix} \vec{\omega}_{n/0} \\ \vec{v}^{O_n}_{n/0} \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^{n} \vec{\omega}_{i/i-1} \\ \sum_{i=1}^{n} \vec{v}^{O_n}_{i/i-1} = \sum_{i=1}^{n} \vec{v}^{O_i}_{i/i-1} + \sum_{i=1}^{n} \vec{\omega}_{i/i-1} \times \vec{t}_{O_i O_n} \end{Bmatrix}$

Projecting into R_0 : $\begin{Bmatrix} {}^0\omega_{n/0} \\ {}^0v^{O_n}_{n/0} \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^{n} {}^0\omega_{i/i-1} \\ \sum_{i=1}^{n} {}^0v^{O_i}_{i/i-1} + \sum_{i=1}^{n} {}^0\omega_{i/i-1} \times {}^0t_{O_i O_n} \end{Bmatrix}$

Using result previous slide $\begin{Bmatrix} {}^0\omega_{n/0} \\ {}^0v^{O_n}_{n/0} \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^{n} \delta_i \dot{q}_i \, {}^0k_i \\ \sum_{i=1}^{n} (1-\delta_i)\dot{q}_i \, {}^0k_i + \sum_{i=1}^{n} \delta_i \dot{q}_i \left({}^0k_i \times \left({}^0t_{O_0 O_n} - {}^0t_{O_0 O_i} \right) \right) \end{Bmatrix}$

Finally : $\begin{Bmatrix} {}^0\omega_{n/0} \\ {}^0v^{O_n}_{n/0} \end{Bmatrix} = \sum_{i=1}^{n} \dot{q}_i \begin{Bmatrix} \delta_i \, {}^0k_i \\ (1-\delta_i) \, {}^0k_i + \delta_i \left({}^0k_i \times \left({}^0t_{O_0 O_n} - {}^0t_{O_0 O_i} \right) \right) \end{Bmatrix}$

Robot Jacobian Matrix

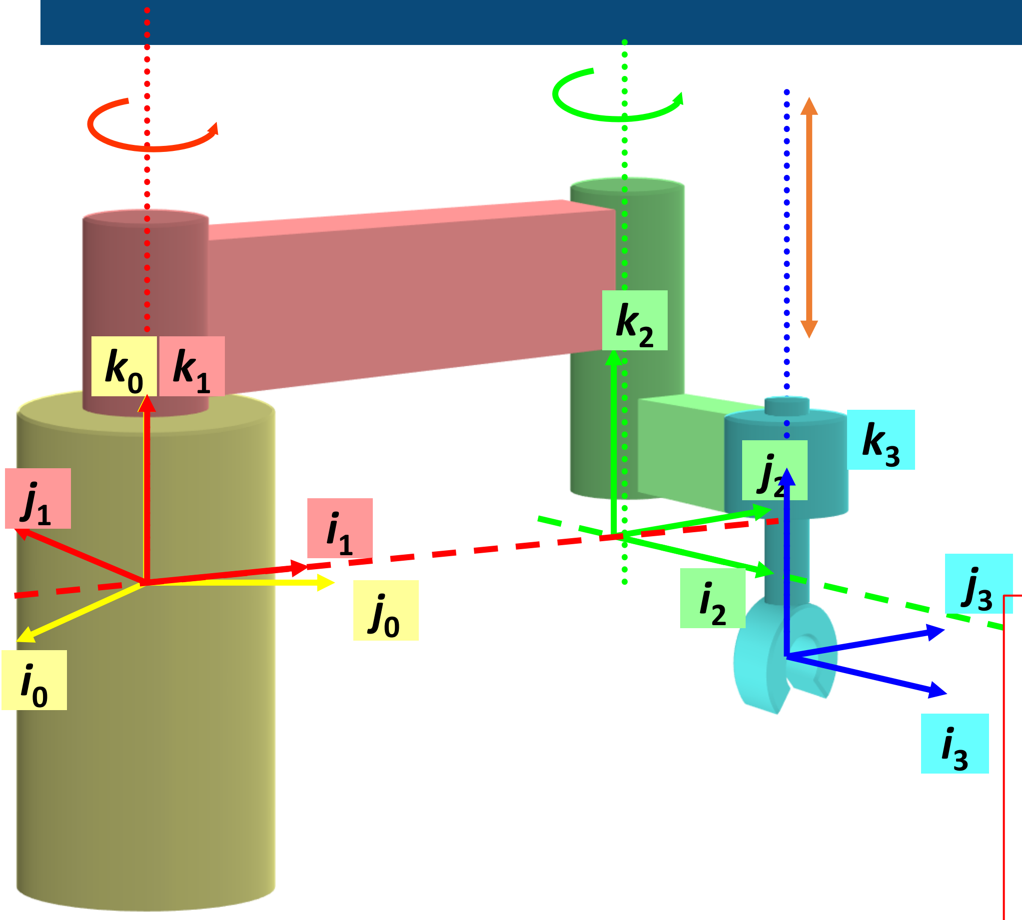
$$\begin{Bmatrix} {}^{0}\omega_{n/0} \\ {}^{0}v^{O_n}_{n/0} \end{Bmatrix} = \sum_{i=1}^{n} \dot{q}_i \begin{Bmatrix} \delta_i \, {}^{0}k_i \\ (1-\delta_i)\, {}^{0}k_i \; + \delta_i \left({}^{0}k_i \; \times \left({}^{0}t_{O_0O_n} - {}^{0}t_{O_0O_i} \right) \right) \end{Bmatrix}$$

Finally : $\begin{Bmatrix} {}^{0}\omega_{n/0} \\ {}^{0}v^{O_n}_{n/0} \end{Bmatrix} = J_{NO_n}(q)\dot{q} = J_N(\dot{q}) \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_i \\ \vdots \\ \dot{q}_n \end{bmatrix}$

With : $J_{NO_n}(q) = \begin{bmatrix} \delta_1 \, {}^{0}k_1 & \cdots & \delta_i \, {}^{0}k_i & \cdots & \delta_n \, {}^{0}k_n \\ (1-\delta_1)\, {}^{0}k_1 \; + \delta_1 \, {}^{0}k_1 \; \times \; {}^{0}p_1 & \cdots & (1-\delta_i)\, {}^{0}k_i \; + \delta_i \, {}^{0}k_i \; \times \; {}^{0}p_i & \cdots & (1-\delta_n)\, {}^{0}k_n \end{bmatrix}$

And : ${}^{0}p_i$ = ${}^{0}t_{O_0O_n} - {}^{0}t_{O_0O_i}$

EXAMPLE : SCARA



i	α_i	a_i	θ_{i+1}	d_{i+1}
0	0	0	θ_1	0
1	0	a_1	θ_2	0
2	0	a_2	0	d_3

$$\begin{Bmatrix} {}^{0}\omega_{3/0} \\ {}^{0}v^{O_3}_{3/0} \end{Bmatrix} = J_{NO_3}(q)\dot{q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ -a_1s_1 - a_2s_{12} & -a_2s_{12} & 0 \\ a_1c_1 + a_1c_{12} & a_1c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \end{bmatrix}$$

Inverse velocity mapping

- One now wants to compute $\dot{q}$ from the twist $\begin{Bmatrix} {}^{0}\omega_{n/0} \\ {}^{0}v^{O_n}_{n/0} \end{Bmatrix}$
- To this aim, it is required to invert the direct mapping:

$$\begin{Bmatrix} {}^{0}\omega_{n/0} \\ {}^{0}v^{O_n}_{n/0} \end{Bmatrix} = J_{NO_n}(q)\dot{q}$$

- It all comes down to study the properties of Jacobian matrix $J_{NO_n}(q)$.

INVERTING THE ROBOT JACOBIAN MATRIX

We here consider that $J_{NO_n}(q) \in \mathbb{R}^{6\times 6}$ (robot with $\boldsymbol{n = 6}$ joints)

In general, the inverse mapping writes : $\dot{q} = \left(J_{NO_n}(q)\right)^{-1} \begin{Bmatrix} {}^0\omega_{n/0} \\ {}^0v^{O_n}_{n/0} \end{Bmatrix}$

However a problem arises when the Jacobian matrix is not invertible. 2 cases :

- If $\det(J_N(q)) \neq 0$ in general, but $\exists q_S$ / $\det(J_{NO_n}(q_s)) = 0$, then q_S is called a singular configuration of the robot.
- If $\forall q \in \mathbb{R}^6, \det\left(J_{NO_n}(q)\right) = 0$, then the robot does not have 6 degrees of freedom (DOF). Only $m < 6$ DOF can be controlled. To determine a control model $(6 - m)$ lines of $J_N(q)$ shall be removed. One then has a non squared Jacobian, with more lines than columns. This corresponds to the redundant case (see later).

Inverting the robot Jacobian matrix

When $J_{NO_n}(q) \in \mathbb{R}^{6\times n}$ (with $\boldsymbol{n < 6}$ joints), then the matrix is non squared and thus invertible.

Method :

1. Build a reduced jacobian matrix by removing $6-n$ lines off the jacobian matrix: $V_r = J_{N_r}(q)\,\dot{q}$

   where $V_r \in \mathbb{R}^n$ contains n controllable components of $\begin{Bmatrix} {}^0\omega_{n/0} \\ {}^0v^{O_n}_{n/0} \end{Bmatrix}$.

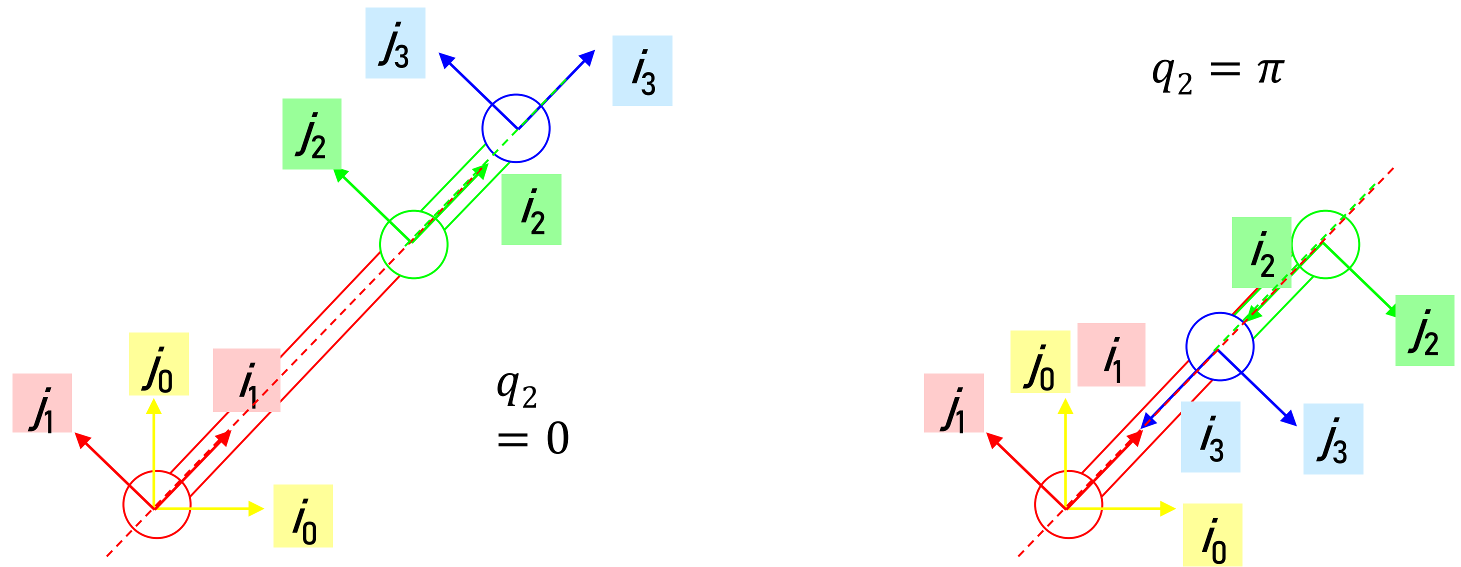
2. Invert: $\dot{q} = \left(J_{N_r}(q)\right)^{-1} \dot{x}$

Remark : one still can encounter a singularity if in general $\det\left(J_{N_r}(q)\right) \neq 0$, although $\exists q_S / \det\left(J_{N_r}(q_S)\right) = 0$.

EXAMPLE : SINGULARITIES OF A SCARA ROBOT

$$\begin{Bmatrix} {}^0\omega_{3/0} \\ {}^0v^{O_3}_{3/0} \end{Bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 \\ a_1 c_1 + a_1 c_{12} & a_1 c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J_N(q)} \begin{bmatrix} \dot\theta_1 \\ \dot\theta_2 \\ \dot d_3 \end{bmatrix} \Rightarrow V_r = {}^0v^{O_3}_{3/0} = \underbrace{\begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 \\ a_1 c_1 + a_1 c_{12} & a_1 c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J_{N_r}(q)} \begin{bmatrix} \dot\theta_1 \\ \dot\theta_2 \\ \dot d_3 \end{bmatrix}$$

$$\det\left(J_{N_r}(q)\right) = (a_1 s_1 + a_2 s_{12}) a_2 c_{12} - a_2 s_{12}\,(a_1 c_1 + a_2 c_{12}) = \boldsymbol{a_1 a_2 s_2}$$



Inversion of the robot Jacobian

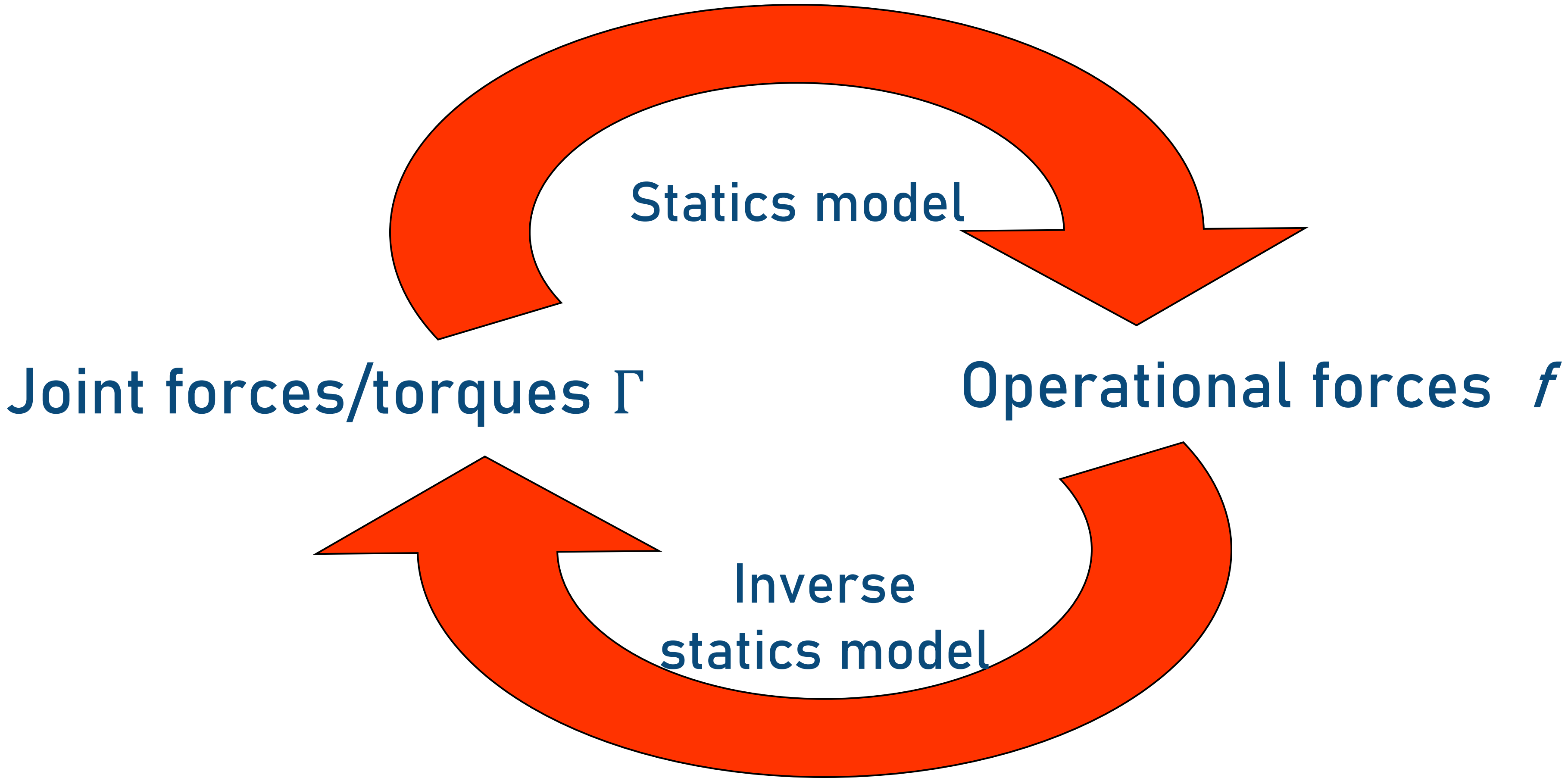
- When $J_{NO\_n}(q) \in \mathbb{R}^{6\times n}$ (robot with $\boldsymbol{n} > \mathbf{6}$ axes : redundancy), then the matrix is non square, non invertible.
- There exists an infinite number of solutions :

$$\dot{q} = \left(J_{NO_n}(q)\right)^{+} \begin{Bmatrix} {}^{0}\omega_{n/0} \\ {}^{0}v^{O_n}_{n/0} \end{Bmatrix} + \left[I_n - \left(J_{NO_n}(q)\right)^{+} J_{NO_n}(q)\right]\xi$$

Where: $\left(J_{NO_n}(q)\right)^{+} = \left(J_{NO_n}(q)\right)^{T} \left[J_{NO_n}(q)\left(J_{NO_n}(q)\right)^{T}\right]^{-1}$ is called the *pseudoinverse* of $J_N O_n(q)$ and $\xi \in \mathbb{R}^n$ random (e.g. : 0, or secondary task)

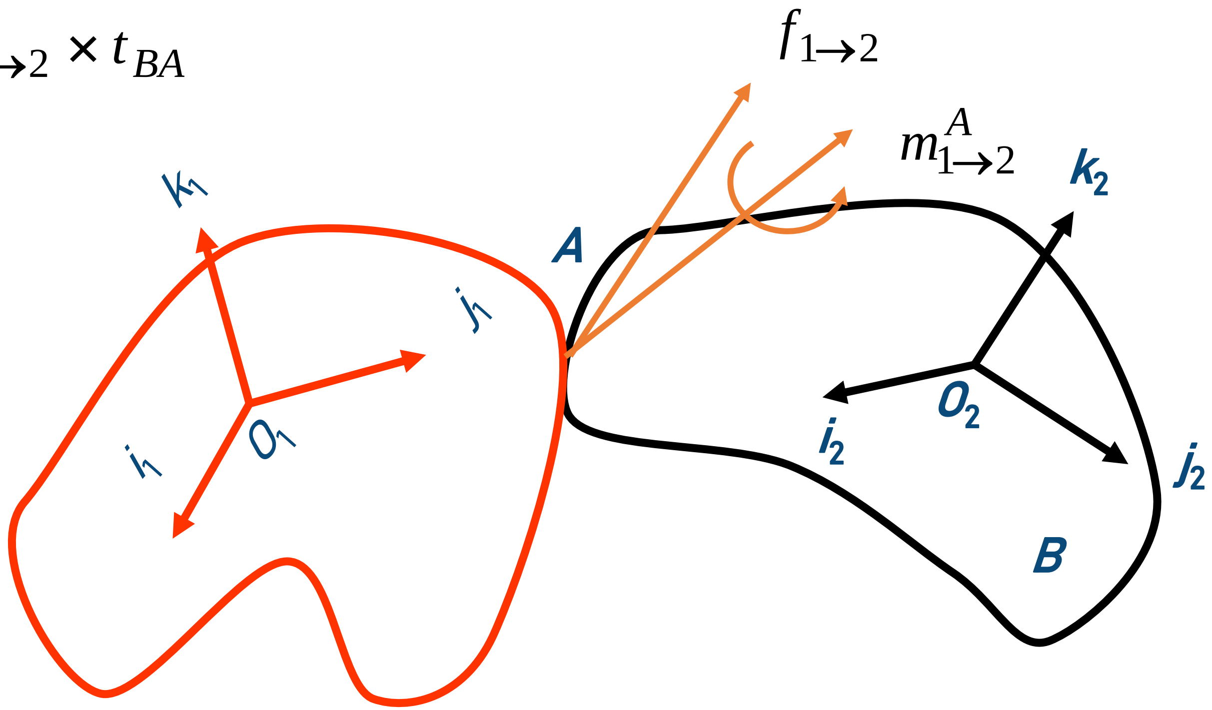
- Remark : if $\det\left(J_{NO_n}(q)\left(J_{NO_n}(q)\right)^{T}\right) \neq 0$ in general, although $\exists q_S / \det\left(J_N(q_s)\left(J_N(q_s)\right)^{T}\right) = 0$, then q_S is a singular configuration.

Force transmission



Wrenches between two rigid bodies

$$m^{A}_{1\to2} = m^{B}_{1\to2} + f_{1\to2} \times t_{BA}$$



Mechanical Power :

$$\mathbf{p} = m^{A}_{1\to2} \cdot \boldsymbol{\omega}_{2/1} + v^{A}_{2/1} \cdot f_{1\to2}$$

Force transmission model

- Considering a robot with n joints, with a velocity transmission model:

$$\begin{Bmatrix} {}^{0}\omega_{n/0} \\ {}^{0}v^{O_n}_{n/0} \end{Bmatrix} = J_{NO_n}(q)\dot{q}$$

- When a torque Γ is applied at the joint level, the mec²hanical power writes:

$$p_j = \Gamma^T \dot{q}$$

- Suppose now that a wrench $\left({}^{0}f_{0\to n}, \; {}^{0}m^{O_n}_{0\to n} \right)$ is applied directly to the end effector, then the mechanical power that it produces writes:

$$p_0 = \left({}^{0}m^{O_n}_{0\to n} \right)^T \, {}^{0}\omega_{n/0} + \left({}^{0}f_{0\to n} \right)^T \, {}^{0}v^{O_n}_{n/0}$$

- The two force systems are said to be equivalent if and only if they produce the same power whatever the velocity, i.e.: $\forall \dot{q}, \; p_j = p_0$
- This straightforwardly leads to:

$$\Gamma = J^T_{NO_n}(q) \begin{Bmatrix} {}^{0}m^{O_n}_{0\to n} \\ {}^{0}f_{0\to n} \end{Bmatrix}$$

Operational space dynamics

- Combing:

$$H(q)\ddot{q} = \underbrace{W\ K_c\ i_m}_{\Gamma_m} - B(q)\dot{q}\dot{q} - C(q)\dot{q}^2 - g(q) - \ \Gamma_{f_t}$$

$$\begin{Bmatrix} {}^{0}\omega_{n/0} \\ {}^{0}v^{O_n}_{n/0} \end{Bmatrix} = J_{NO_n}(q)\dot{q}$$

and $\Gamma = J^T_{NO_n}(q)\begin{Bmatrix} {}^{0}m^{O_n}_{0\to n} \\ {}^{0}f_{0\to n} \end{Bmatrix}$

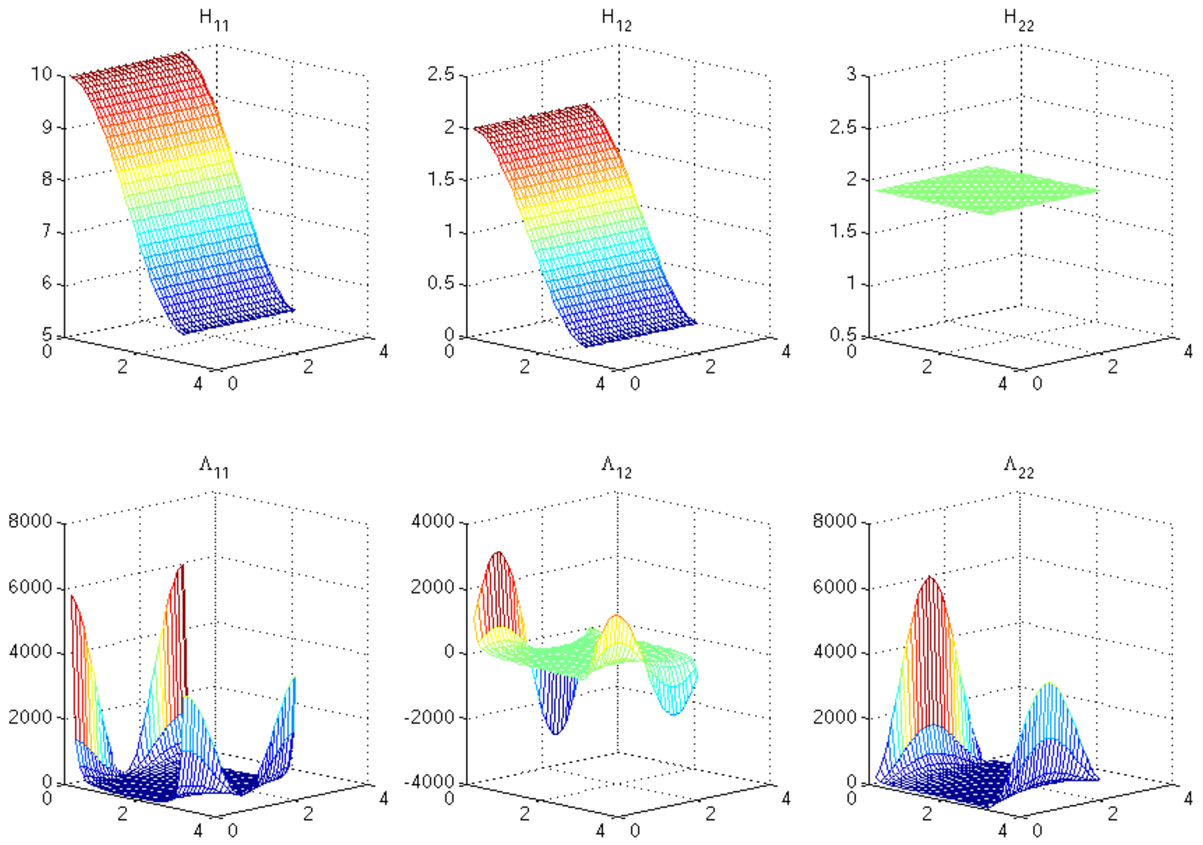
- One gets the operational space dynamics.

Operational space dynamics

$$\Lambda(q)\begin{Bmatrix} {}^{0}\dot{\omega}_{n/0} \\ {}^{0}\dot{v}^{O_n}_{n/0} \end{Bmatrix} = \begin{Bmatrix} {}^{0}m^{O_n}_{0\to n} \\ {}^{0}f_{0\to n} \end{Bmatrix}_c - \mu(q,\dot{q}) - p(q) - \mathrm{f}_{ft}$$

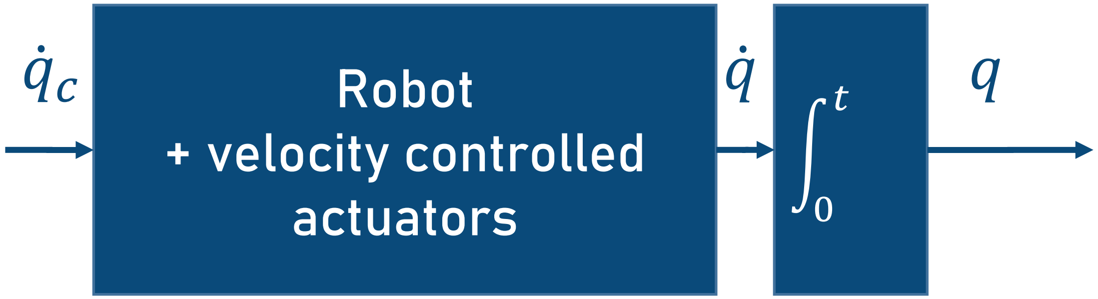
- $\begin{Bmatrix} {}^{0}m^{O_n}_{0\to n} \\ {}^{0}f_{0\to n} \end{Bmatrix}_c = J^{-T}_{NO_n}(q)W\ K_c\, i_m \Leftrightarrow i_m = K_c^{-1}W^{-1}\begin{Bmatrix} {}^{0}m^{O_n}_{0\to n} \\ {}^{0}f_{0\to n} \end{Bmatrix}_c$
- $\Lambda(q) = J^{-T}_{NO_n}(q)H(q)J^{-1}_{NO_n}(q)$
- $\mu(q,\dot{q})$= $J^{-T}_{NO_n}(q)\big(B(q)\dot{q}\dot{q} + C(q)\dot{q}^2 + \dot{J}_{NO_n}(q)\,\dot{q}\big)$
- $p(q\,)$= $J^{-T}_{NO_n}(q)g(q)$ and $\mathrm{f}_{ft} = J^{-T}_{NO_n}(q)\Gamma_{ft}$

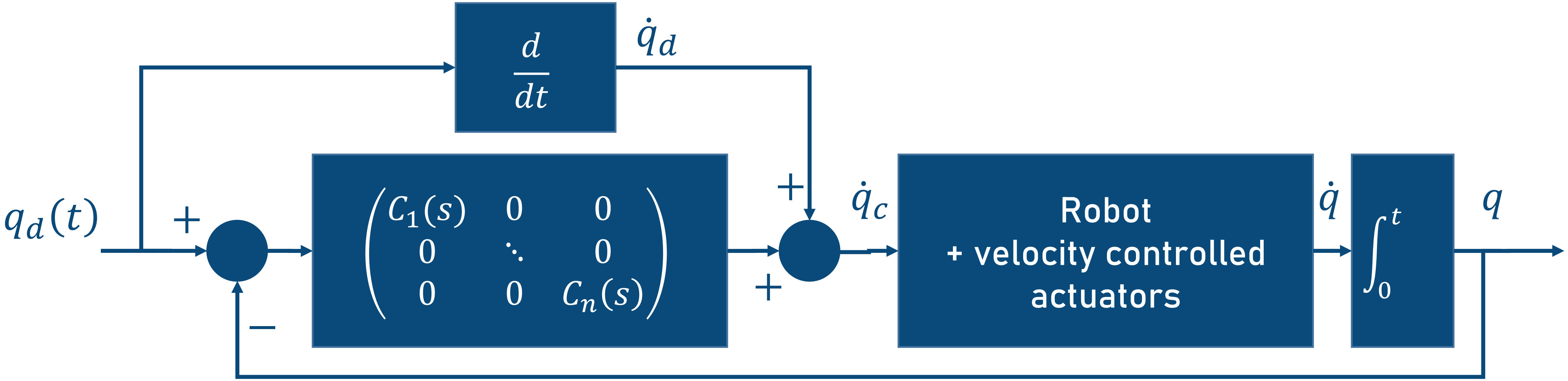
Example: the two first joints of an industrial SCARA robot (IBM 7576)



PART 2: CONTROLLING THE MOVEMENTS OF MULTI-JOINT ROBOTS IN THE FREE SPACE

2.1 WITH VELOCITY CONTROLLED JOINTS





- Compensator $C_1(s)$: a simple gain often suffices :

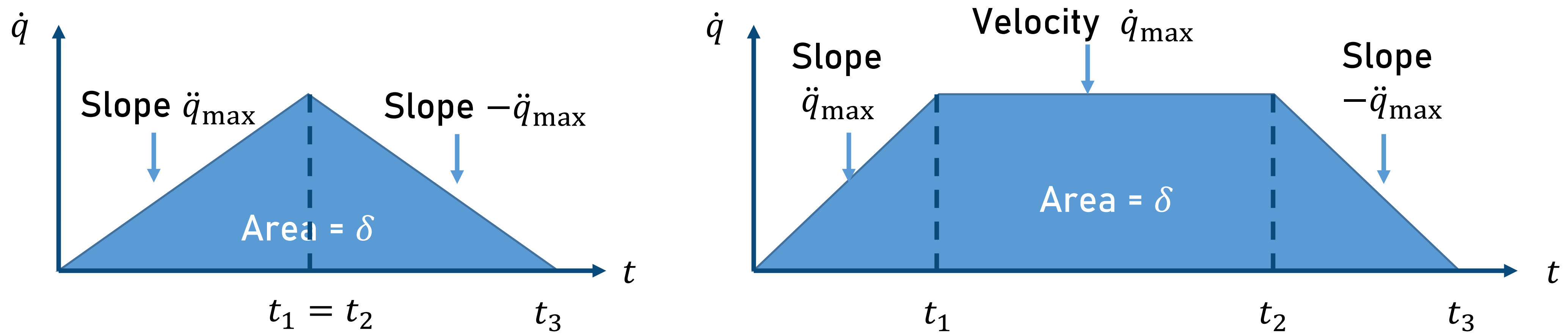
$$\dot{q}_c = \dot{q}_d + \lambda(q_d - q)$$

Computing a trajectory in joint space

- Input data:
  - Initial joint position q_i
  - Final joint position q_f
  - Maximal joint velocity $\dot{q}_{\max}$
  - Maximal joint acceleration $\ddot{q}_{\max}$
- Objective : compute the fastest trajectory given the limits

For 1 joint

- Compute $\delta = q_f - q_i$
- If $|\delta|$ small enough then the fastest trajectory consists in accelerating maximally then decelerating maximally (triangle velocity profile)
- Else, the velocity is saturated to its max value, leading to a trapeze profile.



- Question : knowing δ, $\ddot{q}_{max}$ and $\dot{q}_{max}$, compute t_1 (end of acceleration phase), t_2 (end of constant velocity phase) and t_3 (end of movement)

TRAJECTORY PARAMETERS COMPUTATION

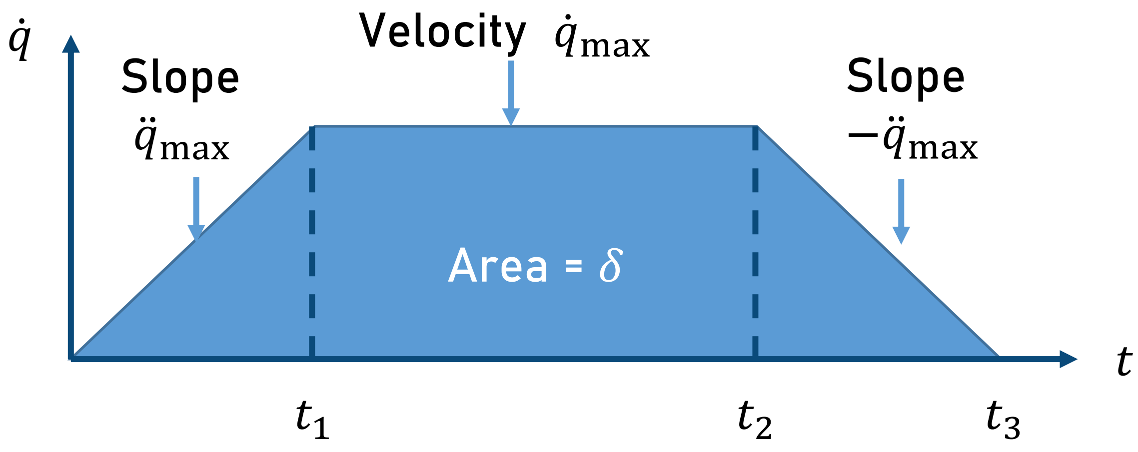
➔ Knowing δ, $\ddot{q}_{max}$ & $\dot{q}_{max}$, compute t_1, t_2 & t_3

If $|\delta| \leq \frac{\dot{q}_{max}{}^2}{\ddot{q}_{max}}$ (triangle)

Then $t_1 = t_2 = \sqrt{\frac{2\,|\delta|}{\ddot{q}_{max}}} = \frac{t_3}{2}$

Else (trapeze):

- $t_1 = \frac{\dot{q}_{max}}{\ddot{q}_{max}}$
- $t_2 = t_1 + \frac{|\delta| - \frac{\dot{q}_{max}{}^2}{\ddot{q}_{max}}}{\dot{q}_{max}}$
- $t_3 = t_2 + t_1$



Trajectory generation

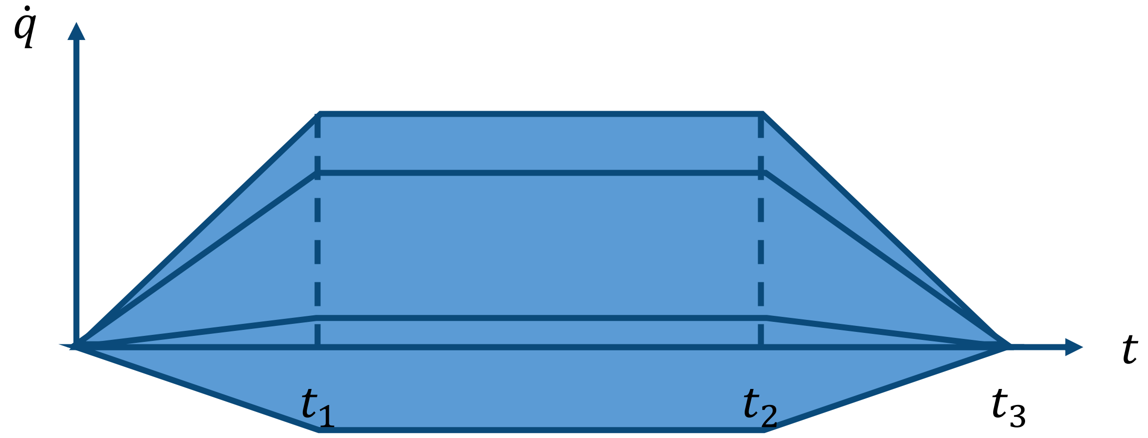
➔ Compute the desired position at a given time t, knowing the initial time t_0 (here set to zero for simplicity) and the trajectory parameters : q_i, q_f, $\ddot{q}_{\max}$, $\dot{q}_{\max}$, t_1, t_2 & t_3

- $\delta \leftarrow q_f - q_i$
- If $t < t_1$, then $\quad q \leftarrow q_i + \mathrm{sgn}(\delta q)\ddot{q}_{\max}\frac{t^2}{2}$
- Else if $t < t_2$ then $\quad q \leftarrow q_i + \mathrm{sgn}(\delta q)\ddot{q}_{\max}\frac{t_1^2}{2} + \mathrm{sgn}(\delta q)\dot{q}_{\max}(t - t_1)$
- Else if $t < t_3$ then $\quad q \leftarrow q_i + \mathrm{sgn}(\delta q)\ddot{q}_{\max}\frac{t_1^2}{2}$

  $+\mathrm{sgn}(\delta q)\dot{q}_{\max}(t - t_1) - \mathrm{sgn}(\delta q)\ddot{q}_{\max}\frac{(t-t_2)^2}{2}$
- Else $\quad q \leftarrow q_f$

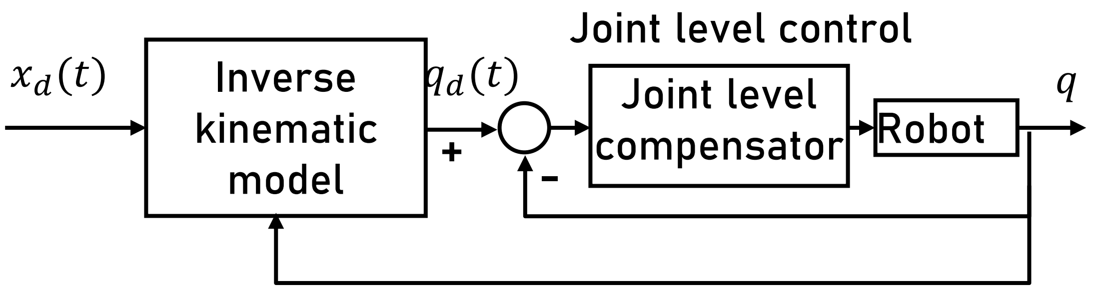
Multi-joint interpolation

- Compute the profile for each joint,
- Select for $t_1, (t_2 - t_1)$ and $(t_3 - t_2)$ the slowest joint.
- Re-compute velocities and accelerations for all joints from t_1, t_2 and t_3



OPERATIONAL SPACE CONTROL BY EXPLOITING THE INVERSE KINEMATICS

- In general, we don't specify q_d. Rather, we specify $x_d = f(q_d)$. How to deal with that?



- The closed loop dynamic is tuned at the joint level (e.g. joint by joint).
- Very easy to implement (used in many industrial controllers)

Computing a trajectory in operational space.

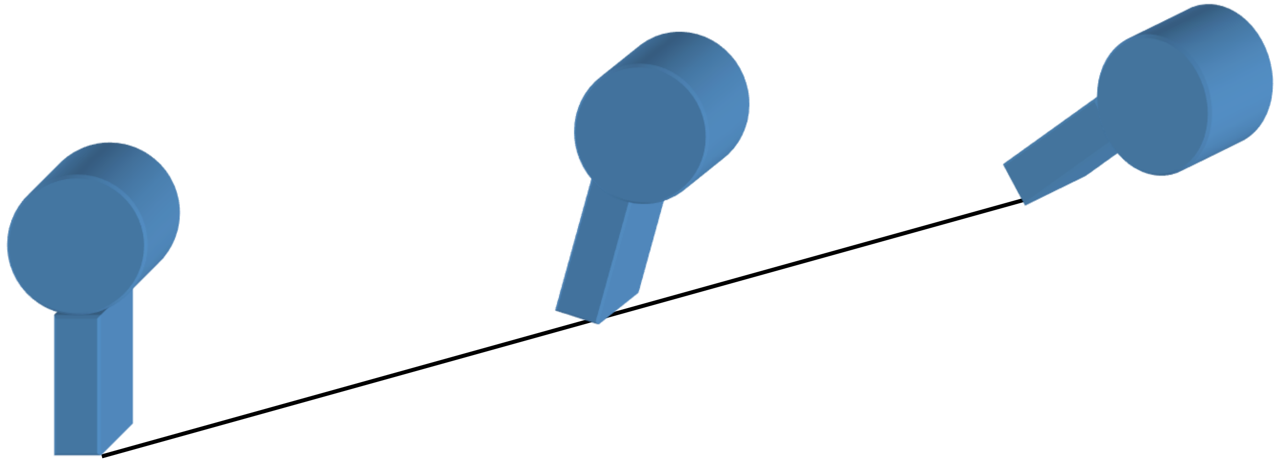
- Operational parameters: $x = [x_p, x_o] = f(q)$.
- General case : $\dim(x) \; = \; 6$.
- Input : Starting point x_i, End point x_f.
- General principles:
  - separate orientation from position.
  - Interpolate geometrically (path without time)
  - Select a velocity and acceleration along the path.
- Interpolation = find $x(\kappa)$,$\kappa \in [0,1]$ such that:
  - Limit conditions are respected, i.e. $x(0) = x_i$ and $x(1) = x_f$.
  - $x(\alpha)$ is continuous (*a minima*), or (twice) continuously differentiable.

Interpolation for positions

- Rather simple: x_p usually consists in the coordinates of a point M in the robot base frame, one then linearly interpolates, i.e.:

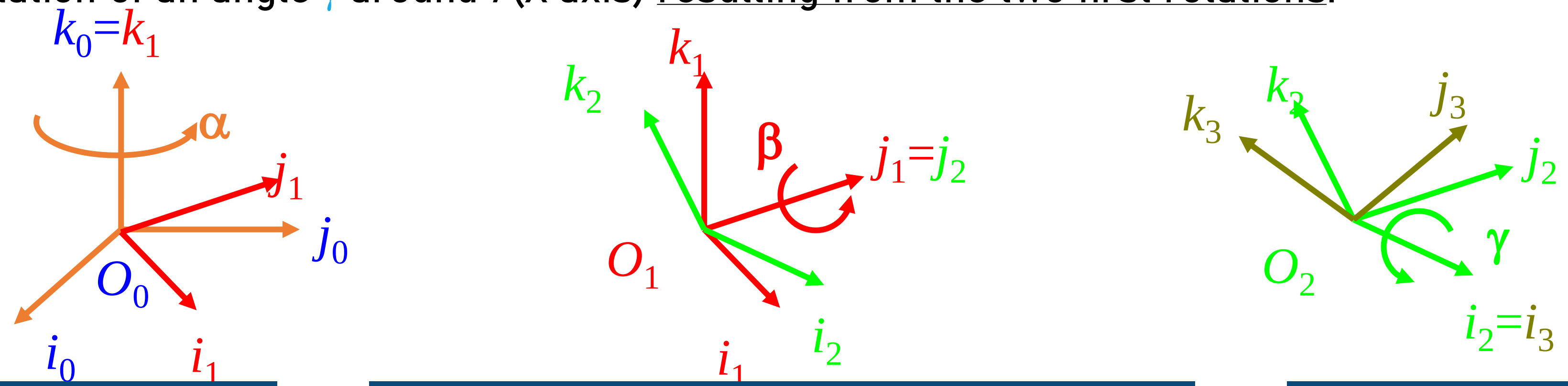
$$x_p(\kappa) = \kappa x_p(1) + (1-\kappa)x_p(0)$$

- The resulting path is a straight line for point M, independently from the orientation of the end-effector that "turns around M".



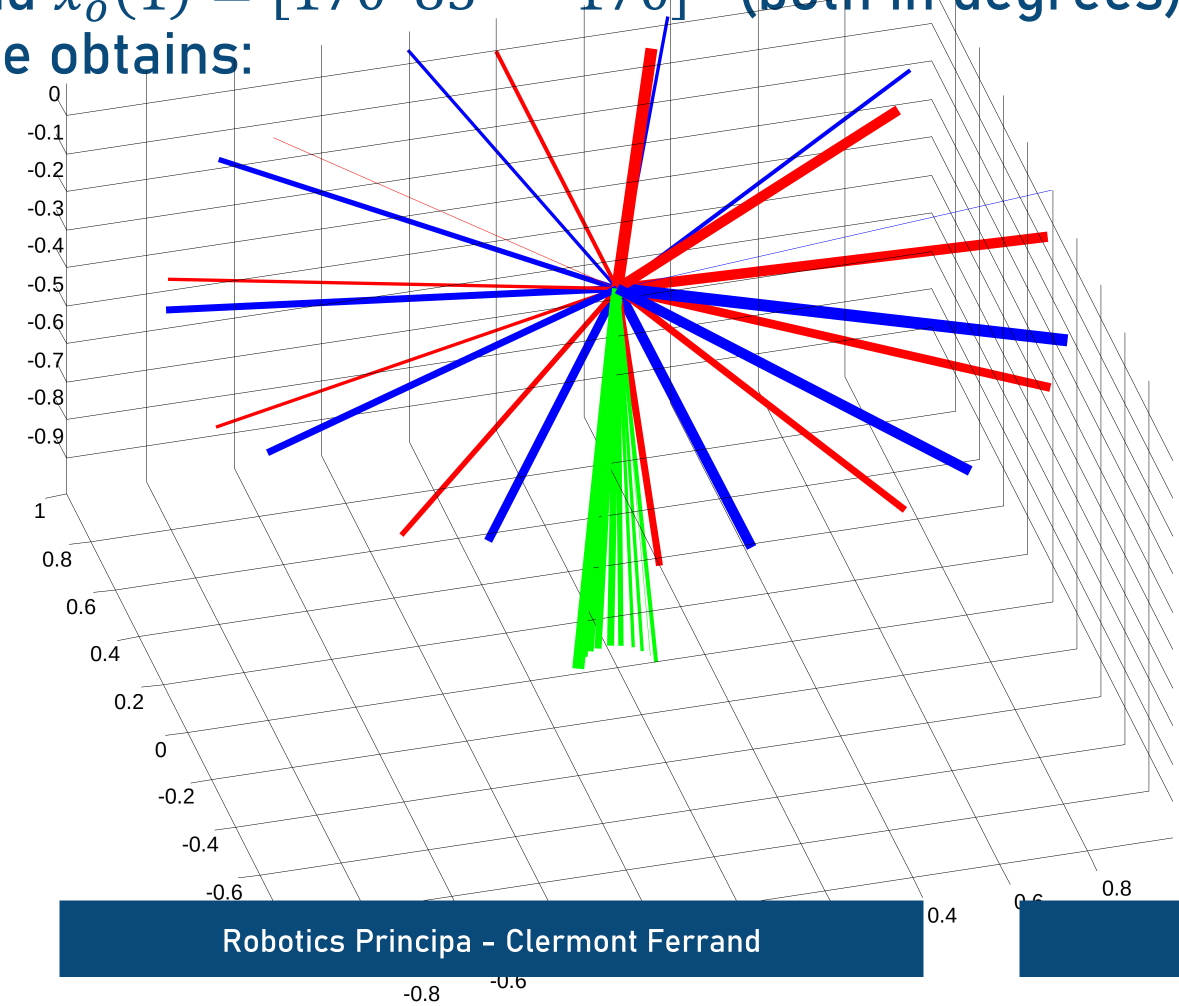
Interpolation for orientations

- Much more complex than for positions
- Usually, x_o contains 3 angles that parameterize the orientation of the end-effector w.r.t. to the robot base.
- One can not interpolate linearly: ~~$x_o(\kappa) = \kappa x_o(1) + (1-\kappa)x_o(0)$~~
- Example : ZYX Euler angle, consisting in:
  1. A rotation of an angle α around *k* (Z axis)
  2. A rotation of an angle β around *j* (Y axis) <u>resulting from the first rotation</u>.
  3. A rotation of an angle γ around *i* (X axis) <u>resulting from the two first rotations</u>.



Example of a linear interpolation on ZYX Euler angles

If $x_o(0) = [0\ \ 85\ \ 0]^T$ and $x_o(1) = [170\ \ 85\ \ -170]^T$ (both in degrees), then, through linear interpolation, one obtains:



Interpolation on a geodesic path

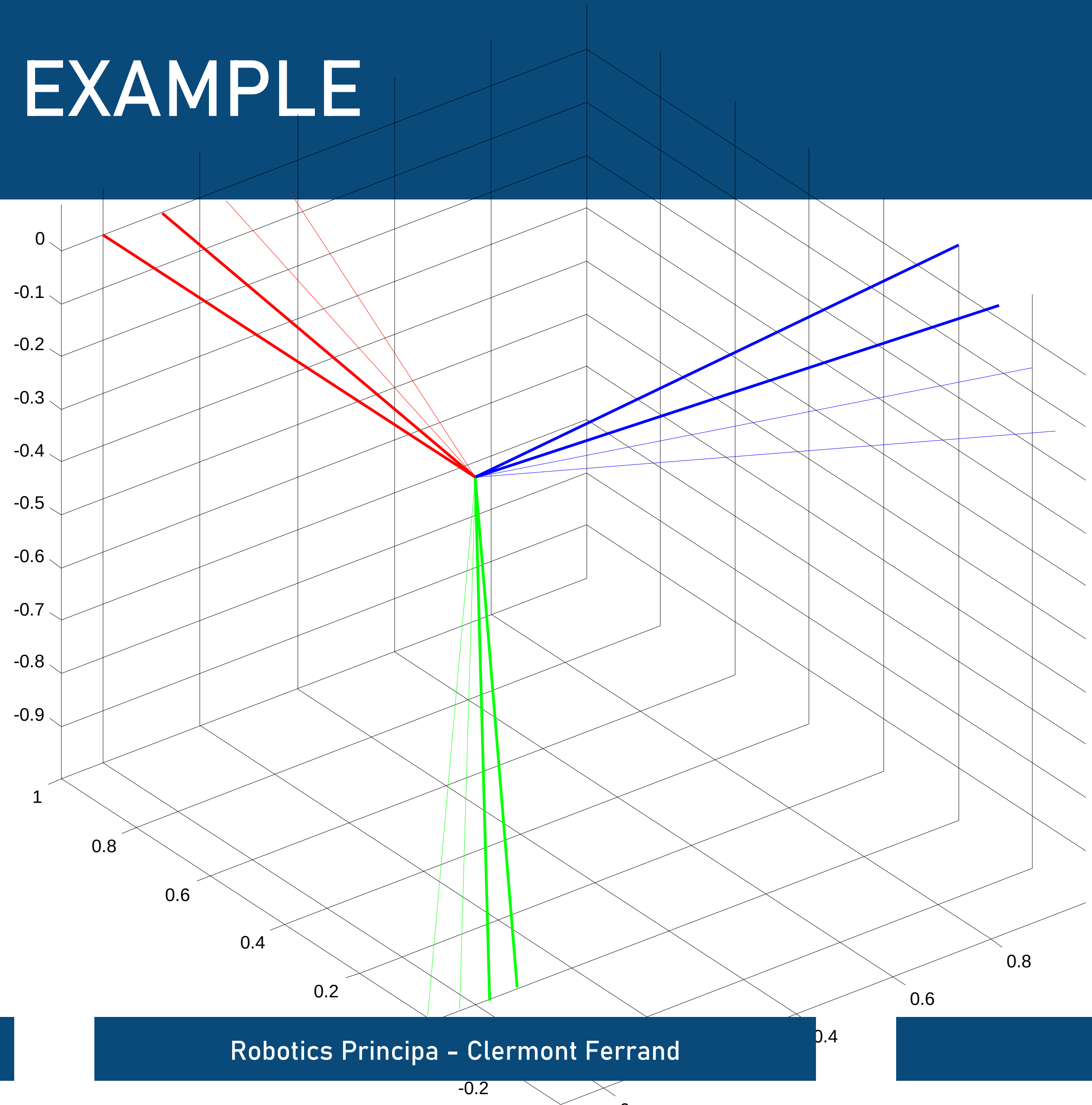
- Between two orientations, a distance can be defined: a **geodesic distance**.
- The geodesic distance is defined by the angle measured around the rotation axis separating the two orientations.
- Following a geodesic path consists in rotating around a fixed axis.
- **It is the shortest path between two orientations**.
- When following a geodesic path around a fixed axis u, the rotational velocity is, at any time, collinear to u. This underlines the similarity between the straight line path for positions and geodesic path for orientations, from a kinematic point of view: in both cases, the shortest path is obtained when the instantaneous velocity vector has a constant direction during the movement (independent from the time evolution of the magnitude).

INTERPOLATION ON A GEODESIC PATH

General principle:

- To determine a geodesic path between two known orientations $x_o(0)$ and $x_o(1)$, it is first required to extract angle θ and unit vector $\boldsymbol{u}$ between the two orientations.
- Then the interpolation consists in rotating around $\boldsymbol{u}$ (which is kept constant) with an angle varying from 0 to θ.

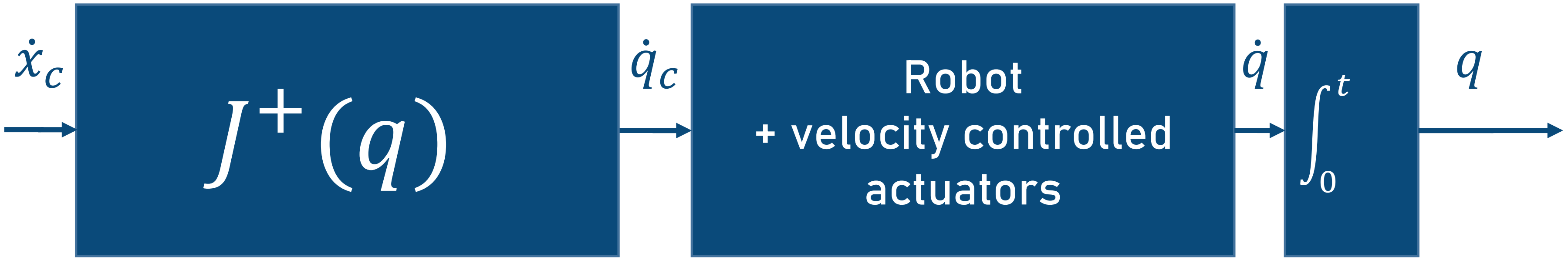
Numerical example



$x_o(0) = [0\ \ 85\ \ 0]^T$ and
$x_o(1) = [170\ \ 85\ \ -170]^T$

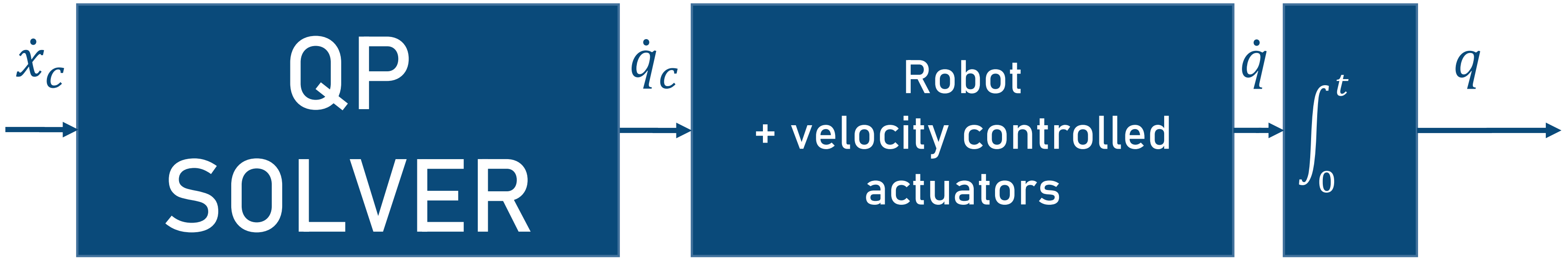
Operational space velocity control

- Just use the (pseudo) inverse jacobian



- Used for:
  - Control loop with external end-effector sensors (e.g. visual servoing, as explained by Youcef Mezouar yesterday)
  - Teleoperation
  - Providing that the movement are not too “fast” (i.e. with inertial coupling effects becoming so large that they significantly affect dynamic tracking precision).

A MORE MODERN WAY: SOLVING A LINEAR QUADRATIC PROBLEM

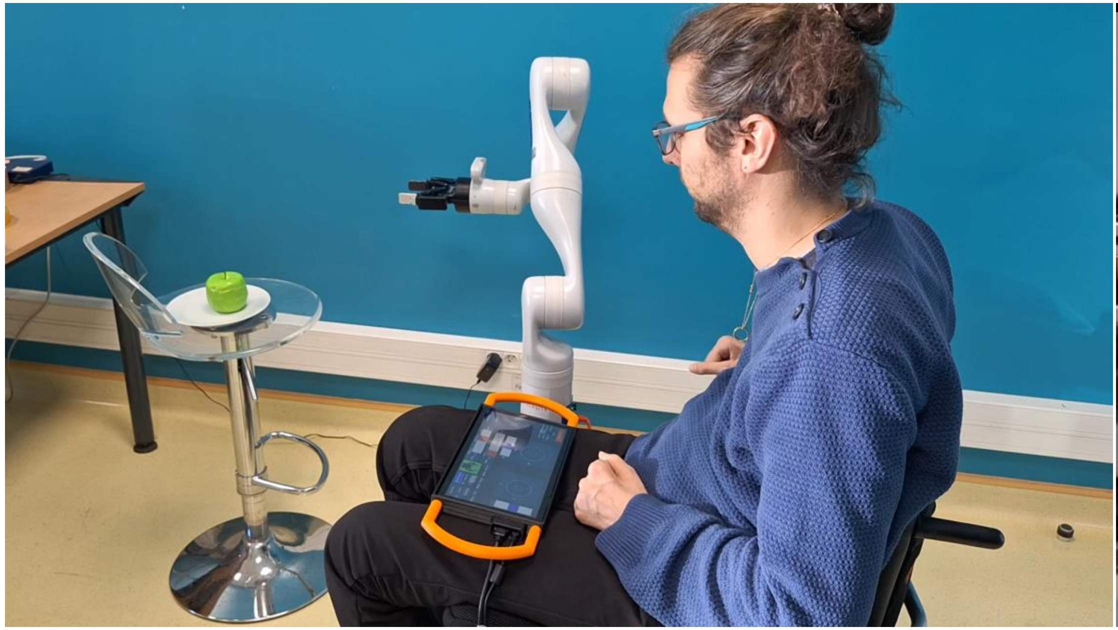


- Provides $\dot{q}_c$ as the solution of an optimization problem:

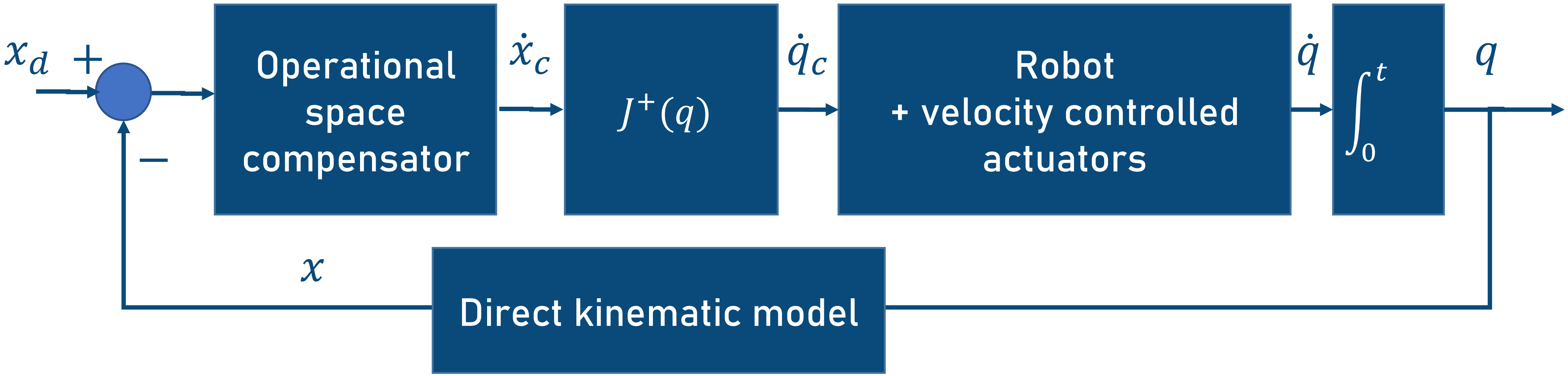
$$\dot{q}_c = \underset{\dot{q}}{\operatorname{argmin}} \left(\|W_x(J(q)\,\dot{q} - \dot{x}_c)\|^2 + \|W_q \dot{q}\|^2 \right)$$

- While strictly respecting a set of constraints, such as
  - Joint position limits
  - Joint velocity limits
  - Joint acceleration limits
  - Any sort of constraint that can be expressed as a linear constraint on the joint velocity.

Video examples



Motion rate control

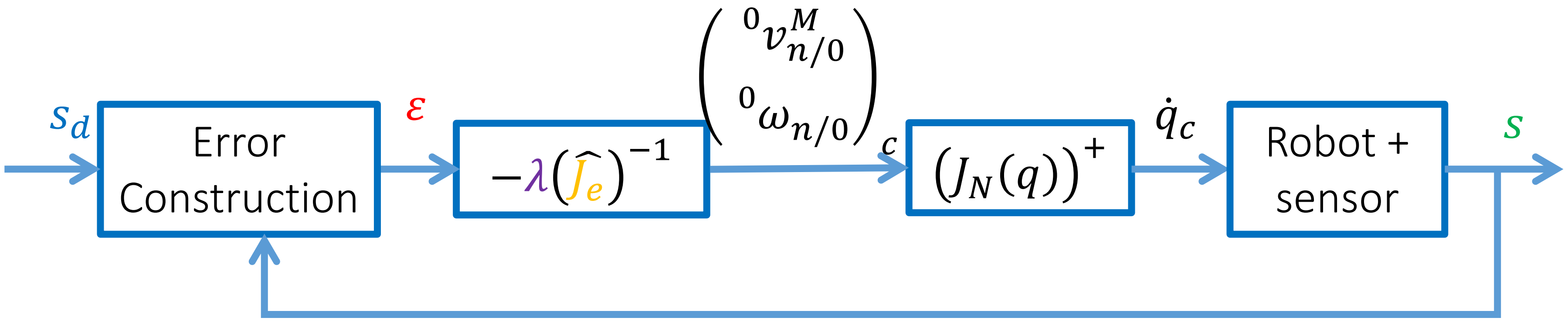


- Compensator : $\dot{x}_c = \lambda(x_d - c) + \dot{x}_d$
- Robust, precise, simple

Motion rate control with an external sensor: see Youcef Mezouar's talk

- The control law writes:

$$\begin{pmatrix} {}^0v^{O_n}_{n/0} \\ {}^0\omega_{n/0} \end{pmatrix}_c = -\lambda\left(\widehat{J_e}\right)^{-1}\varepsilon$$



Incredibly robust and precise

- IF: $\begin{pmatrix} {}^0v^{O_n}_{n/0} \\ {}^0\omega_{n/0} \end{pmatrix} = \begin{pmatrix} {}^0v^{O_n}_{n/0} \\ {}^0\omega_{n/0} \end{pmatrix}_c$, i.e. if the commanded velocity is properly executed by the robot, then:

$$\dot{\varepsilon} = J_e \begin{pmatrix} {}^0v^{O_n}_{n/0} \\ {}^0\omega_{n/0} \end{pmatrix} = J_e \begin{pmatrix} {}^0v^{O_n}_{n/0} \\ {}^0\omega_{n/0} \end{pmatrix}_c = -\lambda J_e \left(\widehat{J_e}\right)^{-1} \varepsilon.$$

- Moreover, IF : $\widehat{J_e} = J_e$, i.e. $\widehat{J_e}$ is a perfect bonne estimate of J_e, then:

$$\dot{\varepsilon} = -\lambda\varepsilon \quad \Rightarrow \varepsilon(t) = e^{-\lambda t}\varepsilon_{t=0}$$

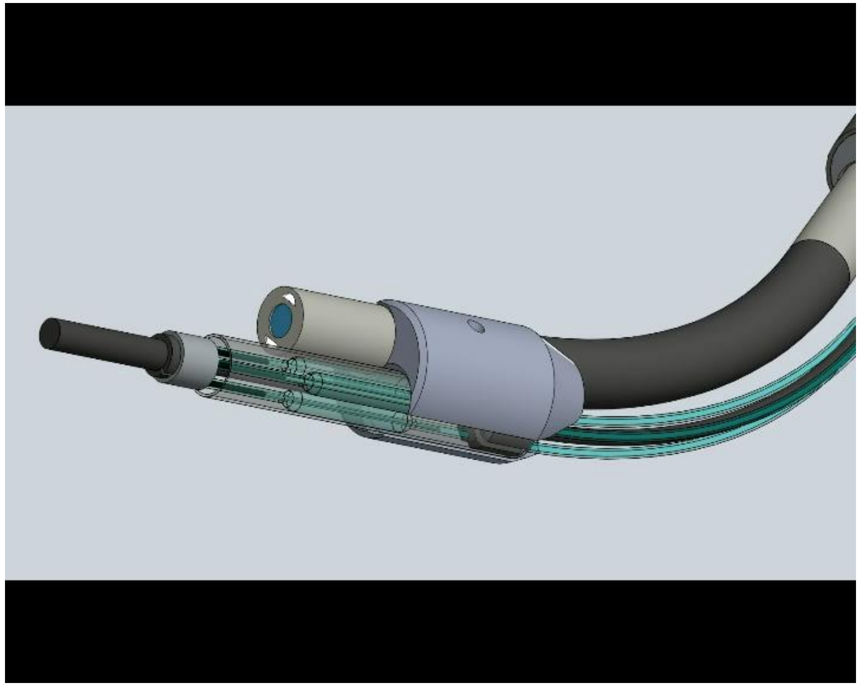
  which exponentially converges towards 0.

- In general $\widehat{J_e} \neq J_e$ but the solution of equation $\dot{\varepsilon} = -\lambda J_e \left(\widehat{J_e}\right)^{-1} \varepsilon$ still converges asymptotically towards zero if and only if $J_e \left(\widehat{J_e}\right)^{-1}$ is positive definite.

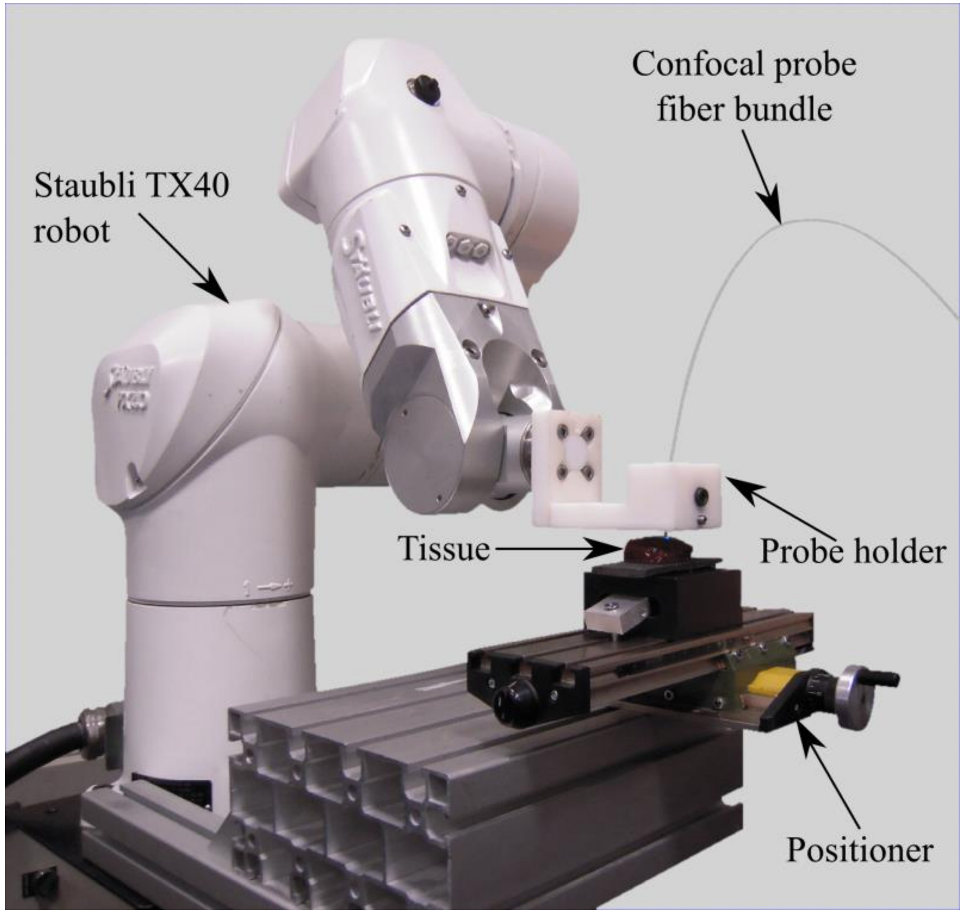
Example: Scanning the surface of soft tissues with a micrometer precision thanks to endomicroscopy based control

Context and aims

- Confocal endomicroscopy
  - Real time imaging
  - Micrometer resolution
  - Small field of view as compared to needs of pathologist => mosaicing
- Illustrated on this video:
  - Laparoscopic device
  - Stabilization
  - Micro-hydraulic actuation.
  - Mosaicing
- Gola to go further : building large mosaics (typically > 1000 images)



Ex-vivo experimental setup to study mosaicing



Mosaicing

- Maximizing a similarity measure between an image I_k and the next one I_{k+1} translated by $T(\zeta)$:

$$\mathrm{sim}\left(I_k, I_{k+1} \odot T(\zeta)\right) = \frac{\sum_j (I_k(j) - \overline{I_k})(I_{k+1}(j+\zeta) - \overline{I_{k+1}})}{\sqrt{\sum_j (I_k(j) - \overline{I_k})^2 + (I_{k+1}(j+\zeta) - \overline{I_{k+1}})^2}}$$

$$\hat{\zeta}(I_k, I_{k+1}) = \mathrm{argmax}\left(\mathrm{sim}\left(I_k, I_{k+1} \odot T(\zeta)\right)\right)$$

$$\hat{\zeta}^s_{k+1} = \frac{\hat{\zeta}(I_k, I_{k+1}) + \hat{\zeta}(I_{k+1}, I_k)}{2}$$

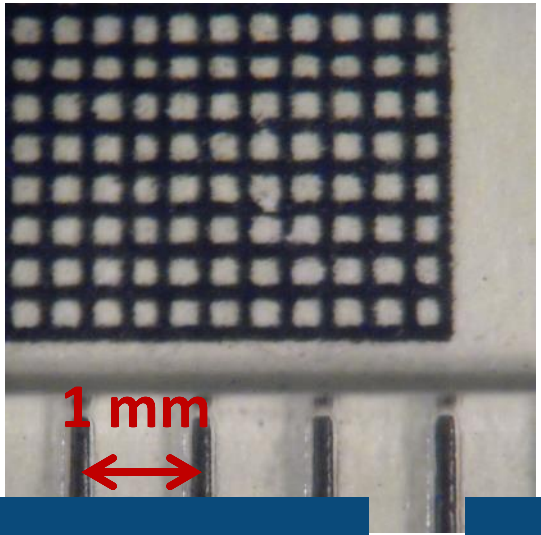
- By summing, this provides a **real-time** measure of the position of I_k with respect to I_0:

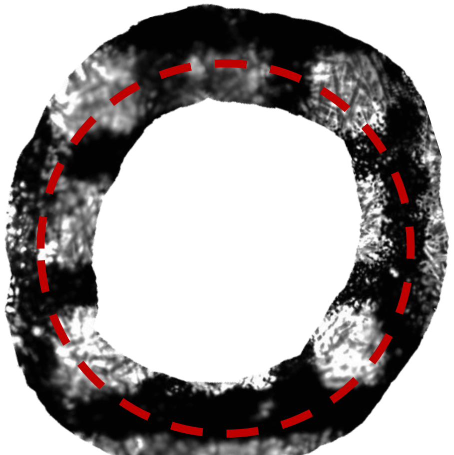
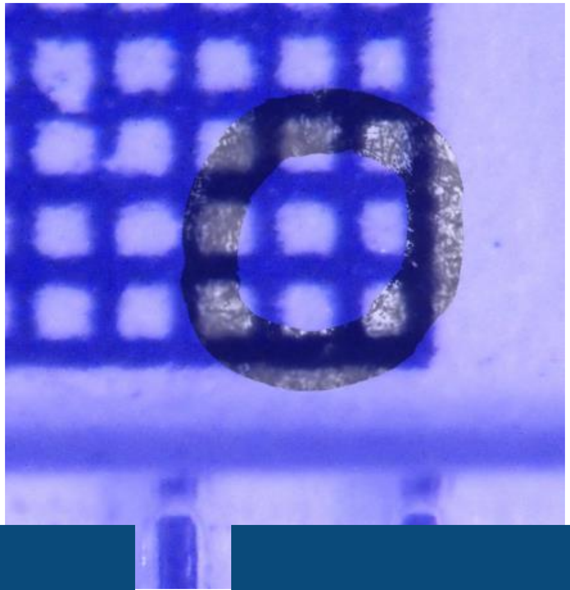
$$\hat{X}_{p/a}(k) = \lambda \sum_{i=0}^{k} \hat{\zeta}^s_k$$

- **Off-line** optimization allows to compensate for the drift.

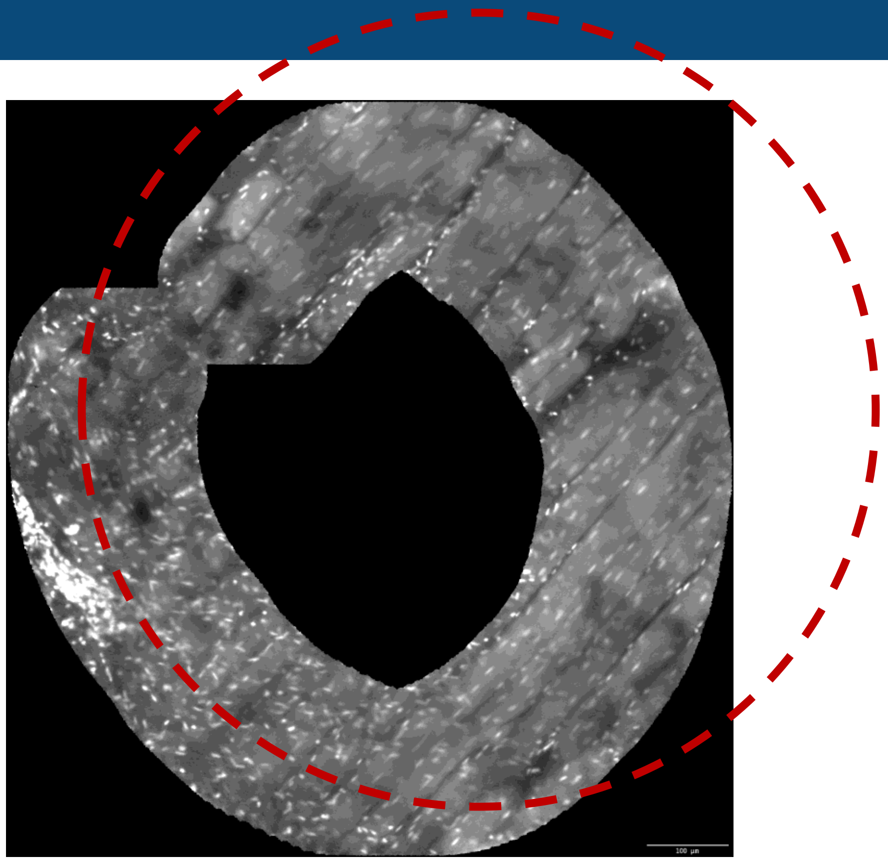
RESULTS (PRINTED PAPER)

1. A 0.3 mm pitch grid laser printed on paper (and pictured – left)
2. The Stäubli robot is asked to draw a 1 mm diameter circle.
3. The real time mosaic is reconstructed and compared to the circular trajectory (middle)
4. Images are manually superimposed (right)



Results (chicken breast)

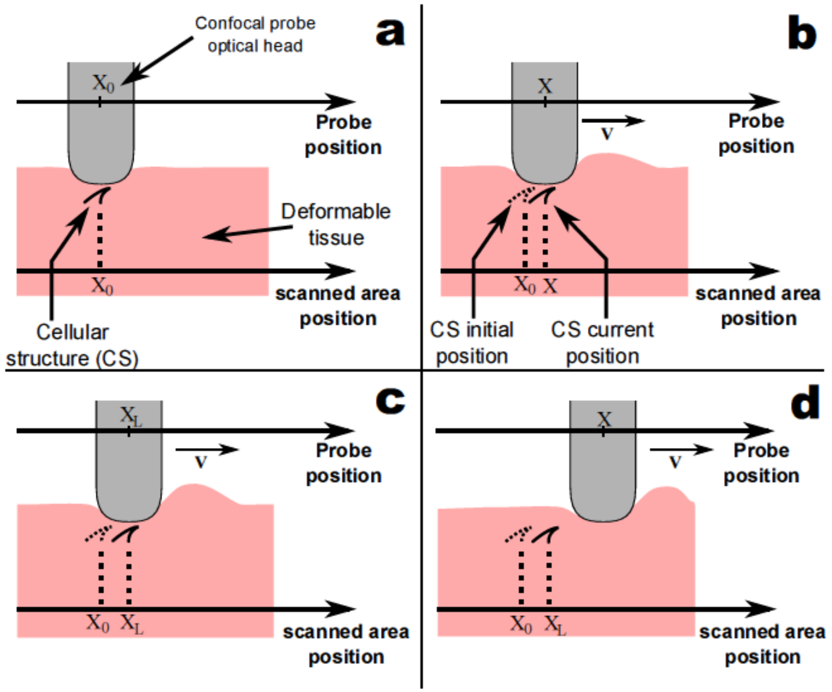


Contrarily to the paper experiment, a large distortion is observed.
This is due to tissue deformation [Erden et al, IROS 2012].

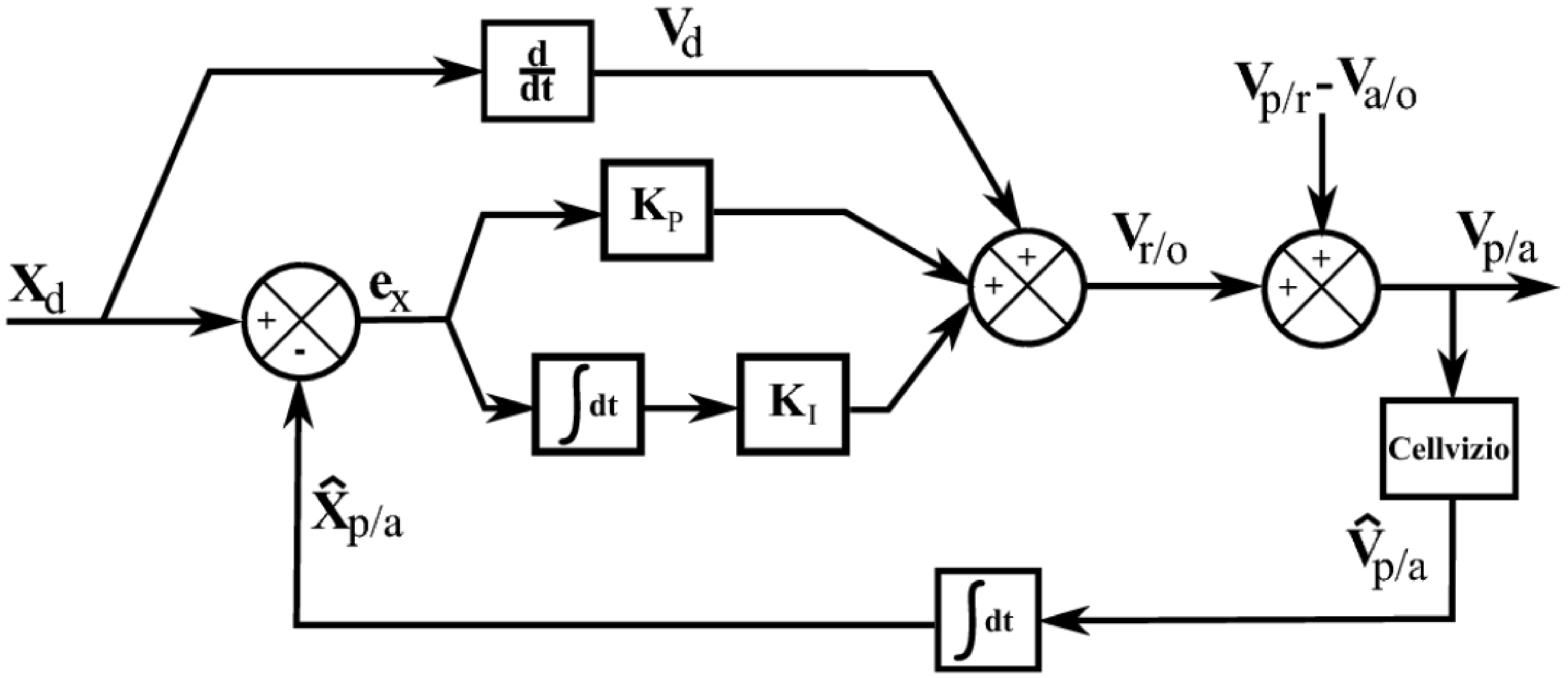
Summary results of open loop mosaicing

- Real-time mosaicing exhibits very low drift under the given experimental conditions (30µm in average for circle experiments with paper).
- Tissue deforms.

$$V_{p/a} = V_{p/r} + V_{r/0} - V_{a/0}$$

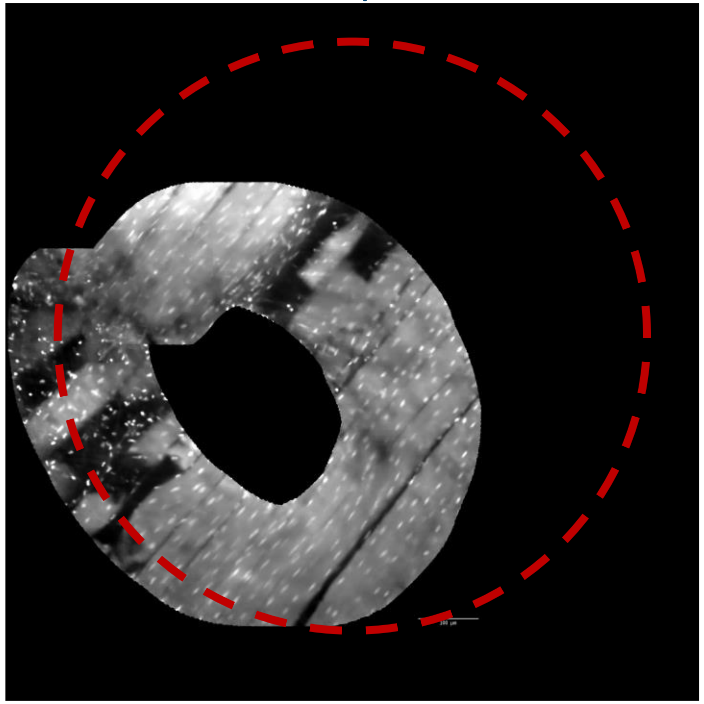


Motion rate control loop

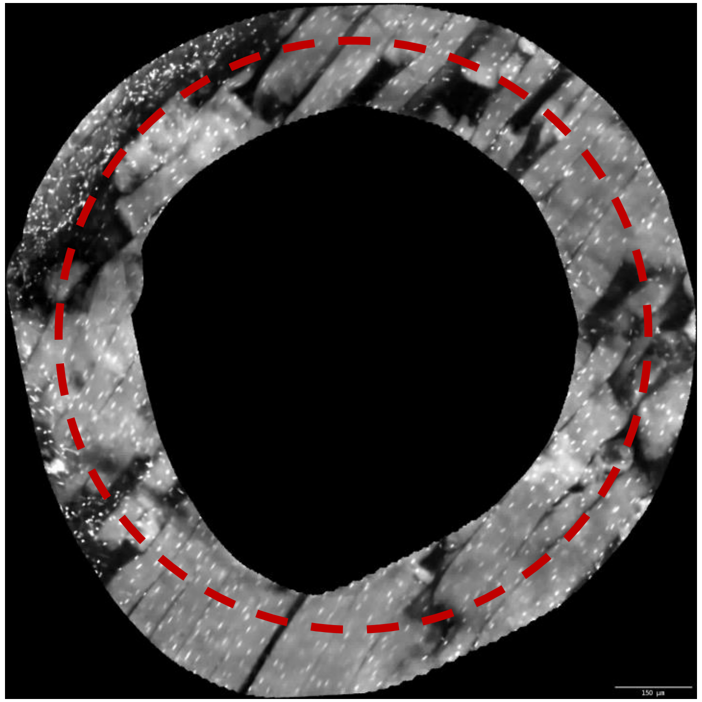


RESULTS

Without microscope based control

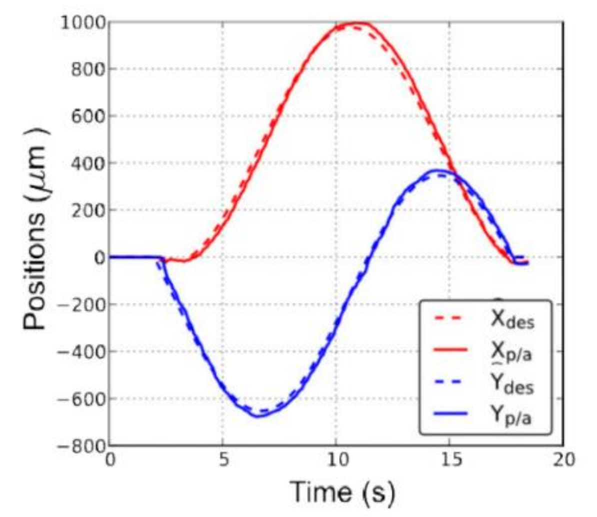


With microscope based control

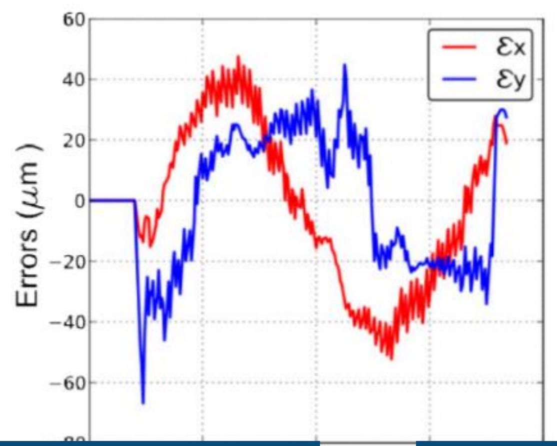


- Integral compensation allows to compensate for tissue deformations distortions
- Mean servoing error:39µm (max: 87µm)

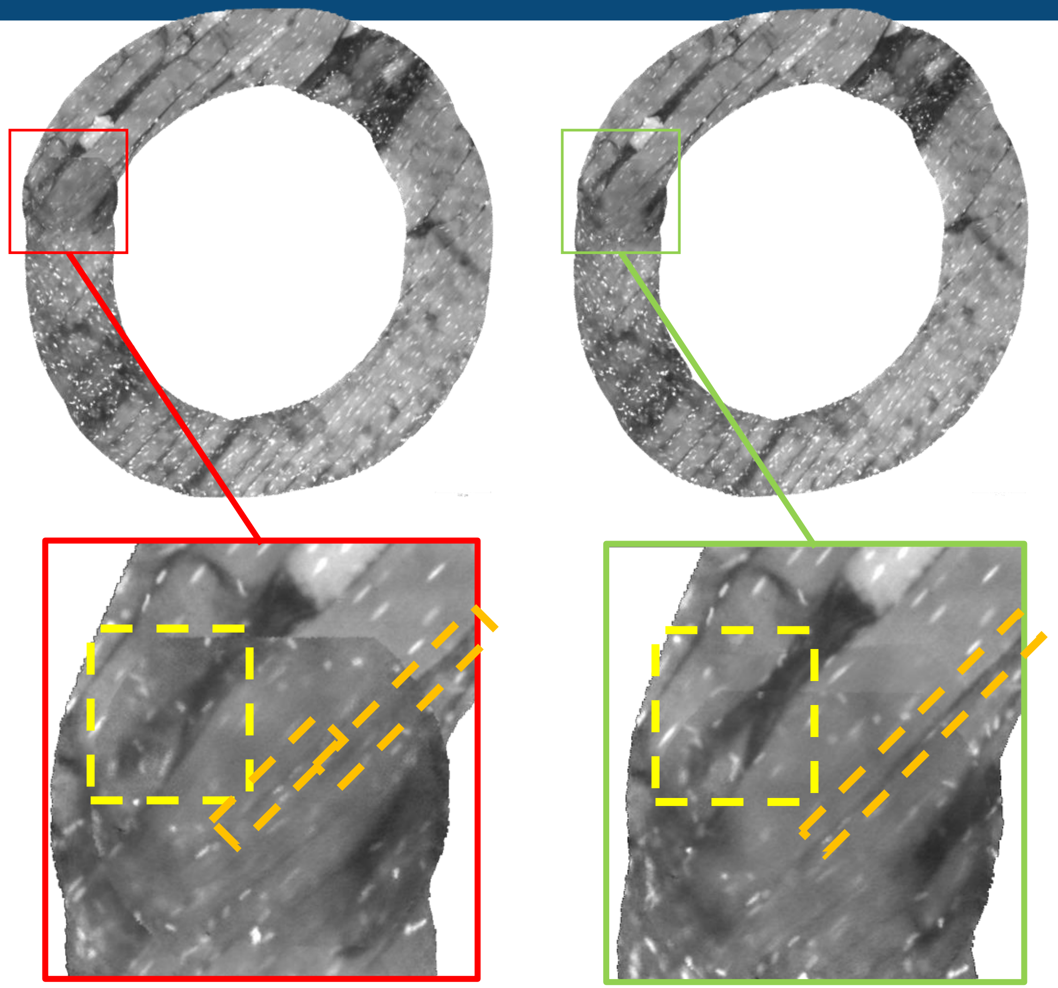
Results (2)



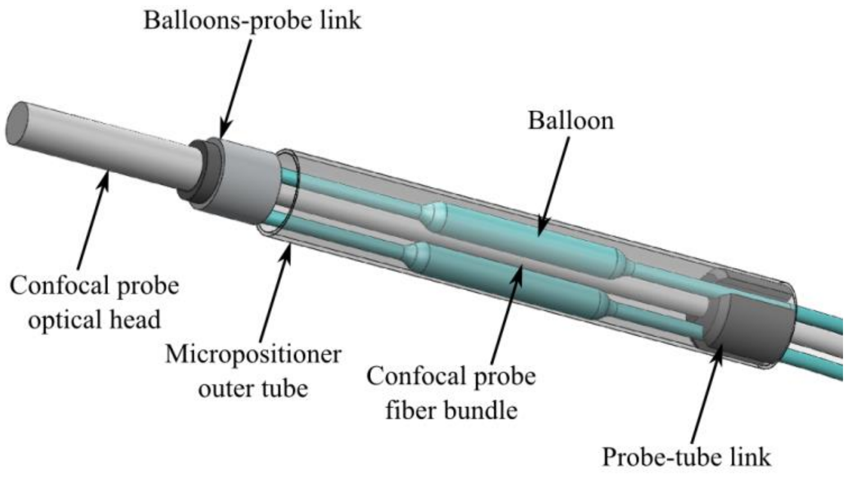
Positions (µm)
Time (s)
1000
800
600
400
200
0
−200
−400
−600
−800
0
5
10
15
20
X_{des}
$X_{p/a}$
Y_{des}
$Y_{p/a}$

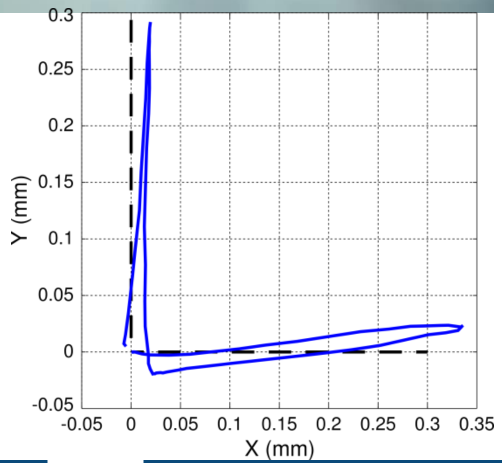


Errors (µm)
60
40
20
0
−20
−40
−60
ε_x
ε_y



APPLICATION TO THE MINIMALLY INVASIVE DEVICE

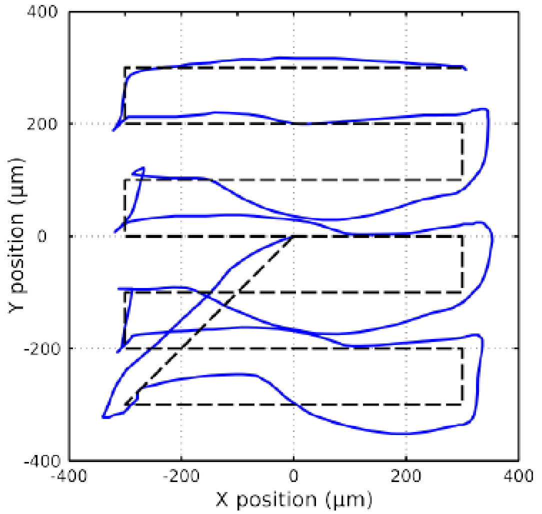


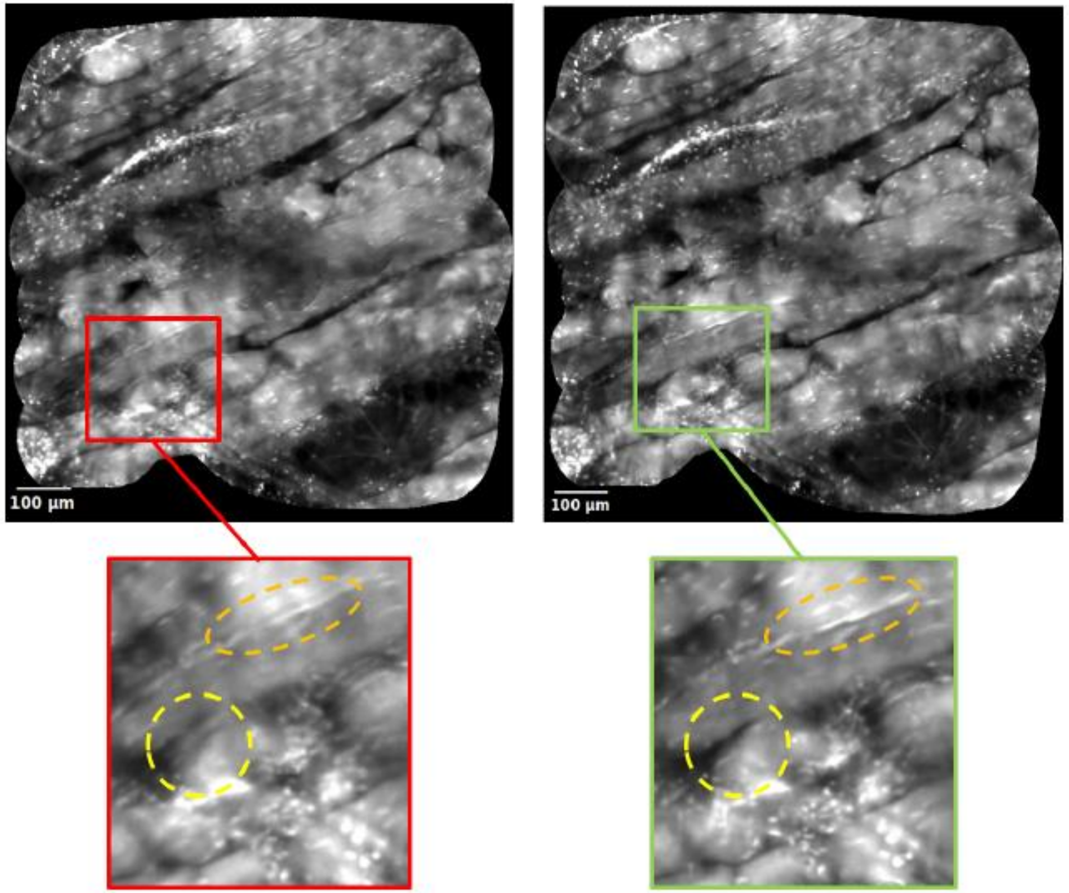


- **Velocity control:**

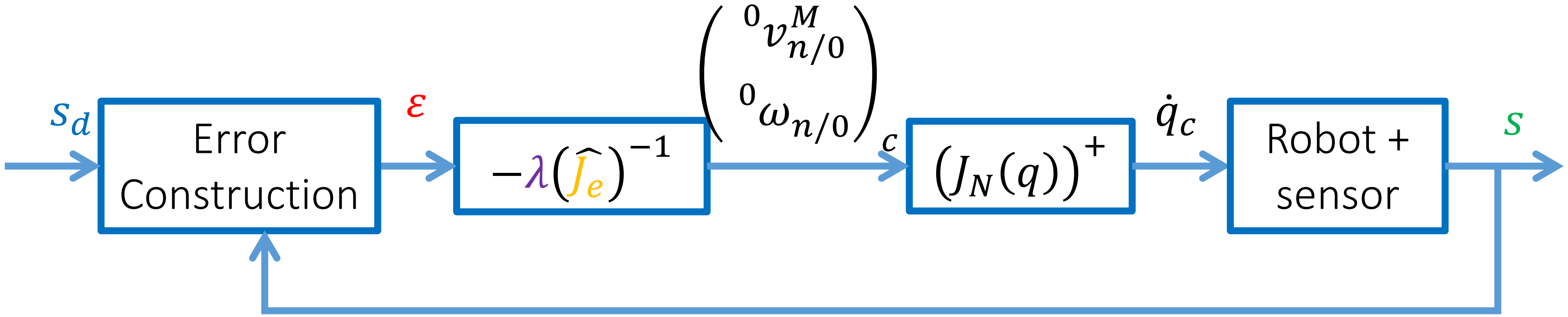
$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \frac{1}{3a} \begin{pmatrix} \sqrt{3} & 1 \\ -\sqrt{3} & 1 \\ 0 & -2 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} V_{r/0}$$

Results (ex vivo)



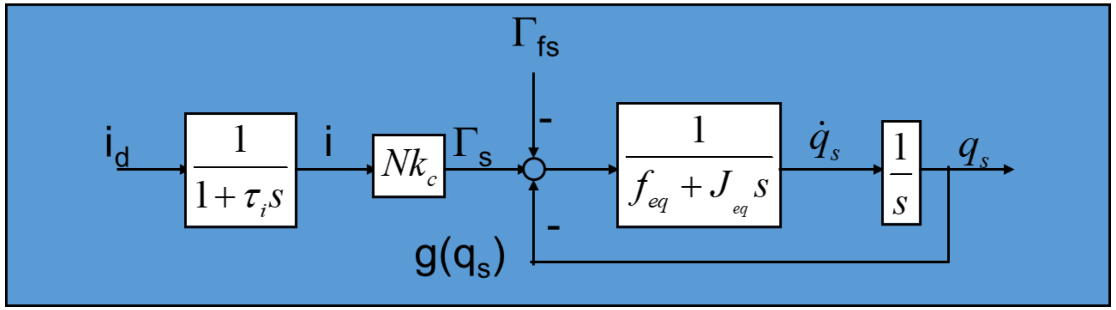


SUMMARY ON MOTION RATE CONTROL

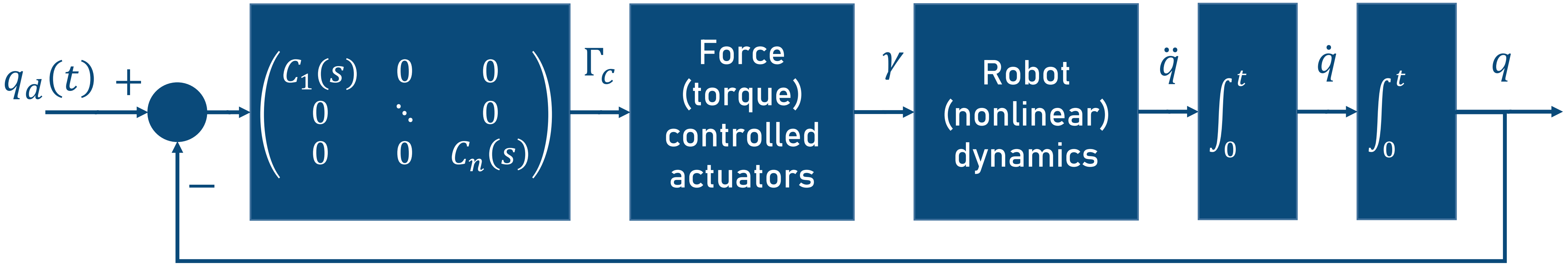


- Condition: being able of modelling J_e such that $\dot{s} = -J_e \begin{pmatrix} {}^0v^{O_n}_{n/0} \\ {}^0\omega_{n/0} \end{pmatrix}$ and computing in real time a reasonable estimate $\widehat{J_e}$ for J_e
- In practice, even with a rough estimate, the system converges:
  - Towards a null error.
  - in the sensor world (in interventional systems, medical images are the ground truth, not 3D coordinates)
  - within a few seconds

2.2 WITH TORQUE CONTROLLED JOINTS



JOINT SPACE POSITION SERVOING WITH INNER CURRENT TORQUE LOOP

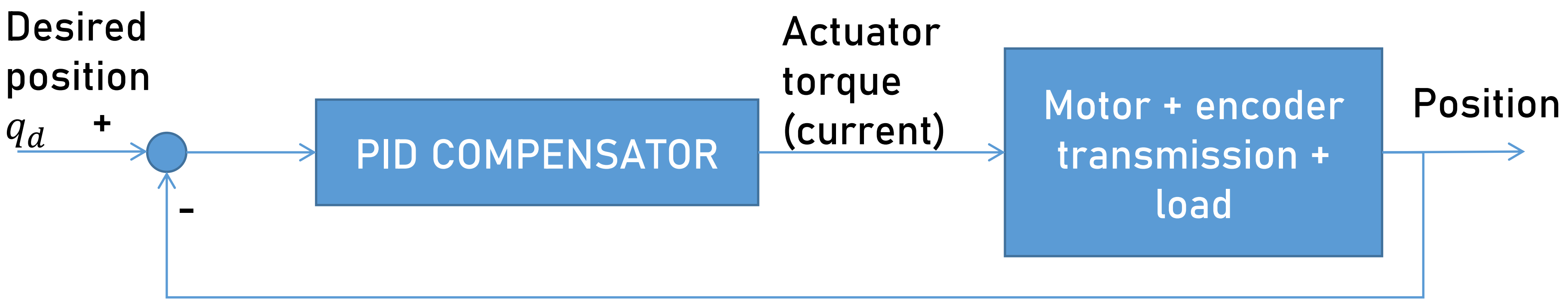


- Compensator $C_1(s)$ = PID:

$$\gamma_c = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + K_i \int_0^t (q_d - q)d\tau$$

- K_p= stiffness, K_d= viscosity, K_i: no mechanical analogue, the torque varies (e.g. increases) as long as the error is not zero (thus compensating for disturbances like friction or weight).

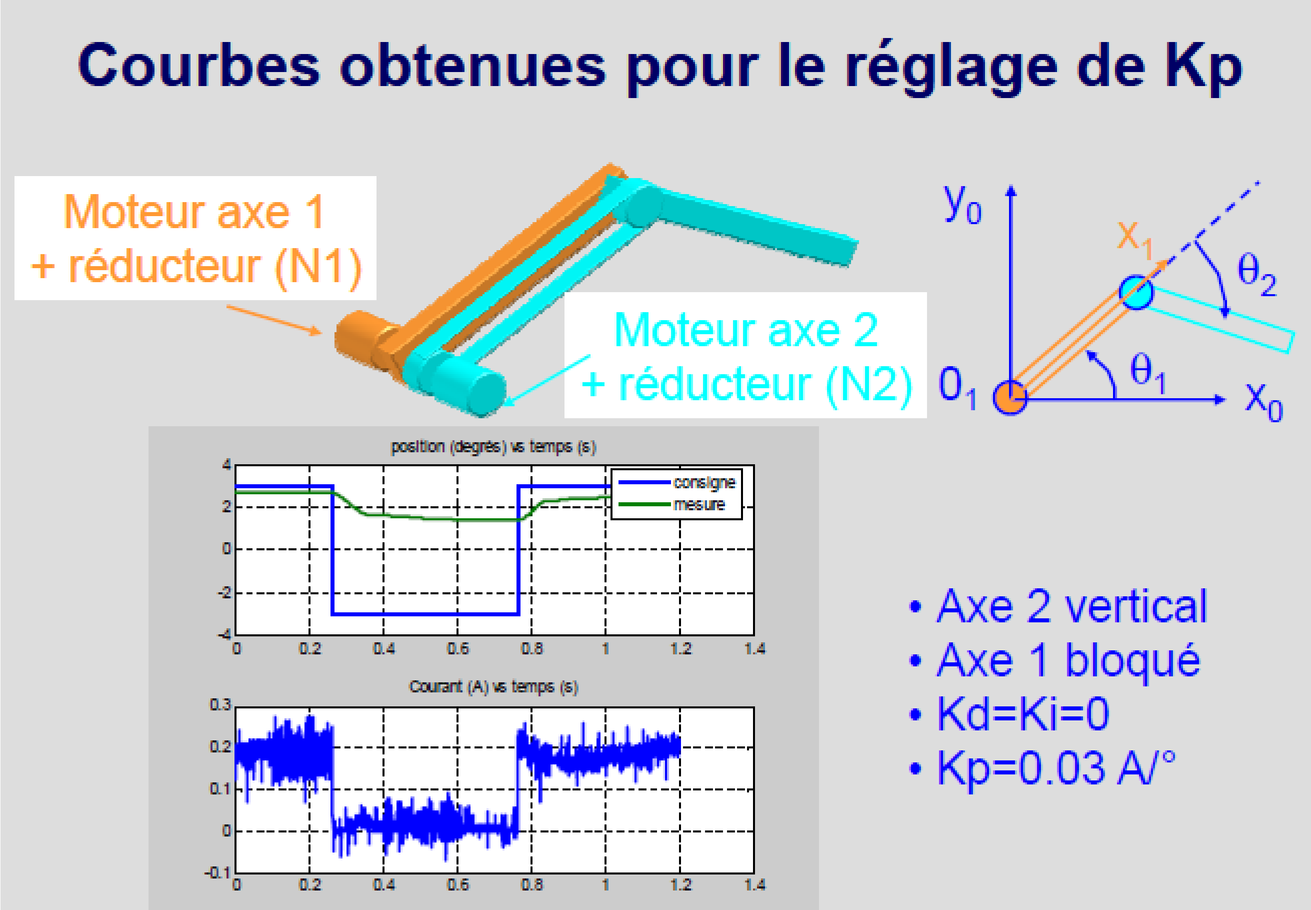
Principle of joint position control



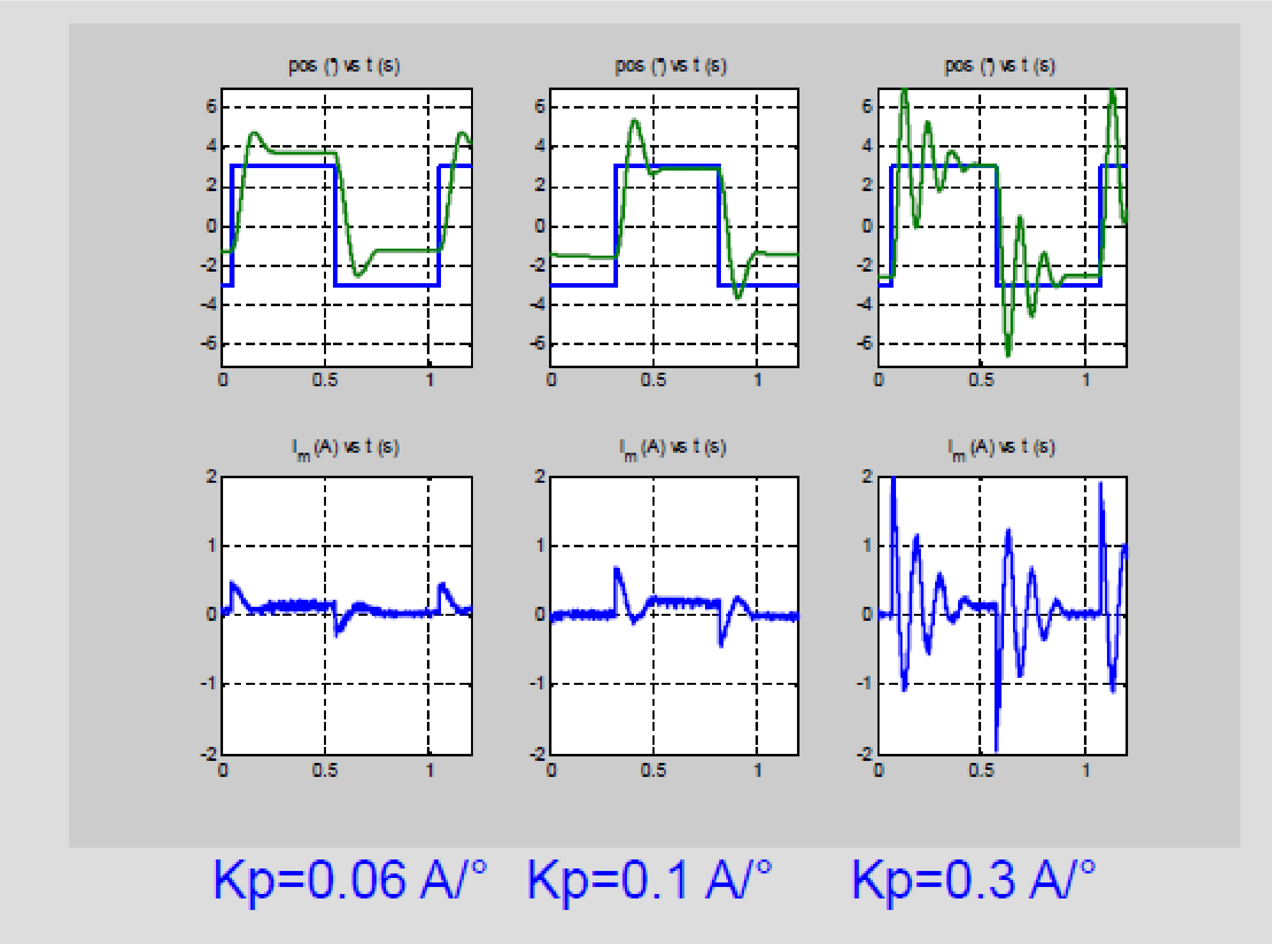
PID compensator:

$$\Gamma = K_p\left(q_d(t) - q(t)\right) + K_d \frac{d\left(q_d(t) - q(t)\right)}{dt} + K_i \int_0^t \left(q_d(\tau) - q(\tau)\right) d\tau$$

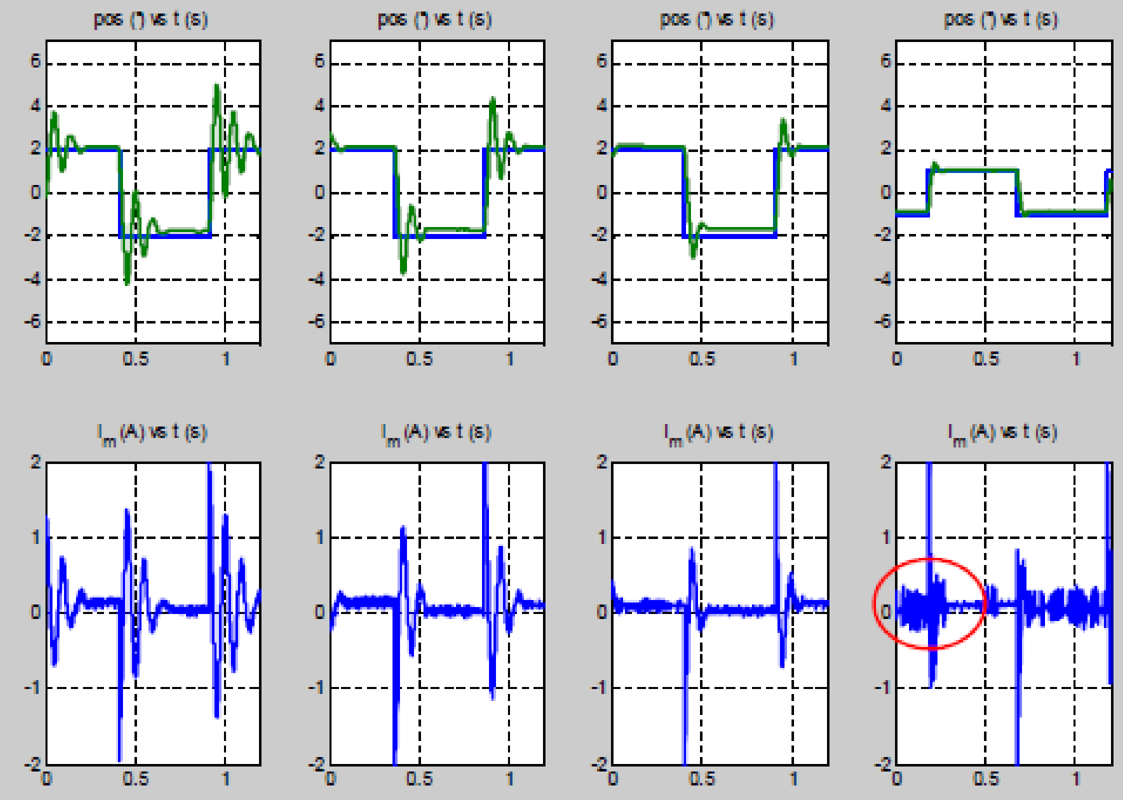
Tuning: a practical approach, case of a robot with gearboxes (i.e. friction)



Increasing K_p

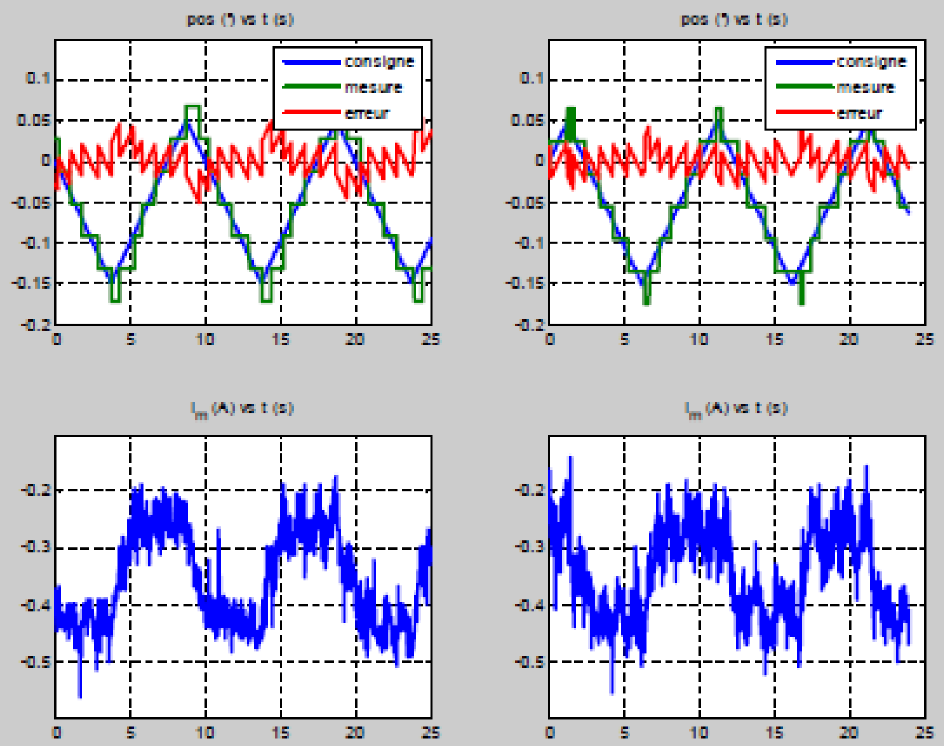


Then increasing K_d



Kp=0.5	Kp=0.5	Kp=0.5	Kp=0.5
Kd=0.	Kd=0.001	Kd=0.003	Kd=0.01

Finally increase K_I for slow movements



Kp=0.5
Kd=0.006
Ki=4

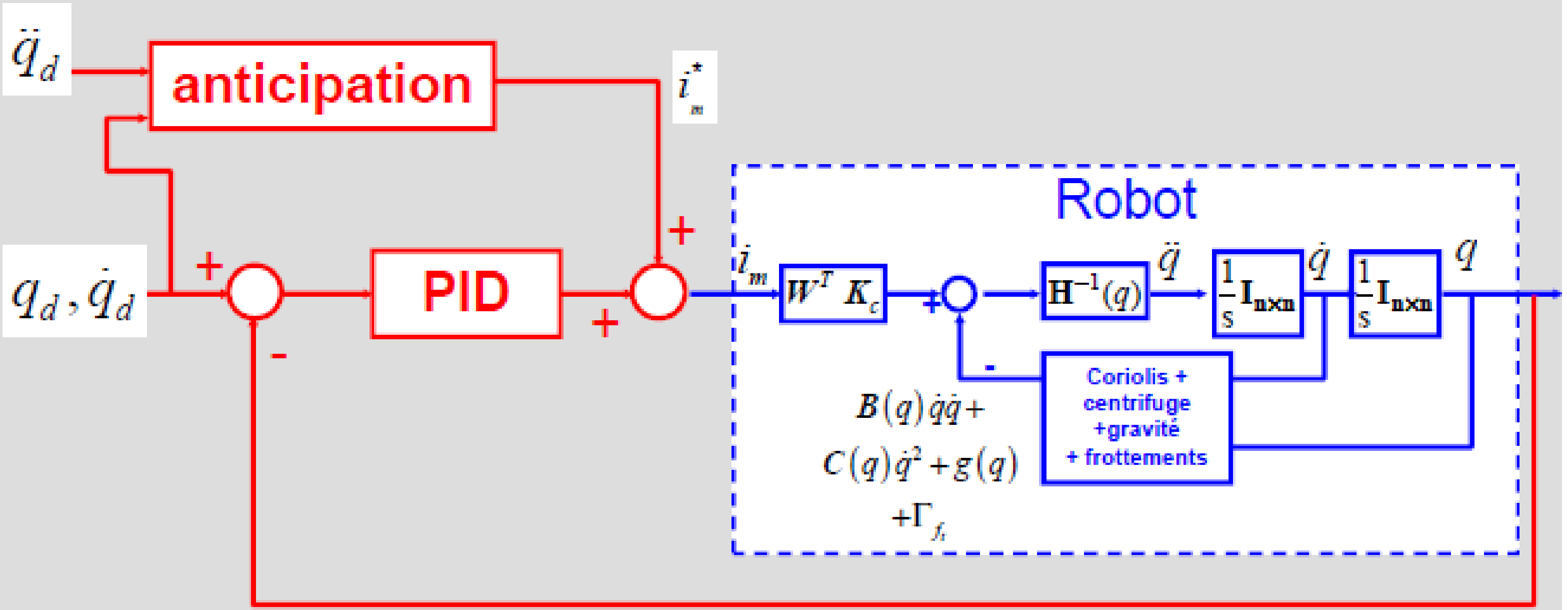
Kp=0.5
Kd=0.006
Ki=10

USING A DYNAMIC MODEL IN THE CONTROL LOOP

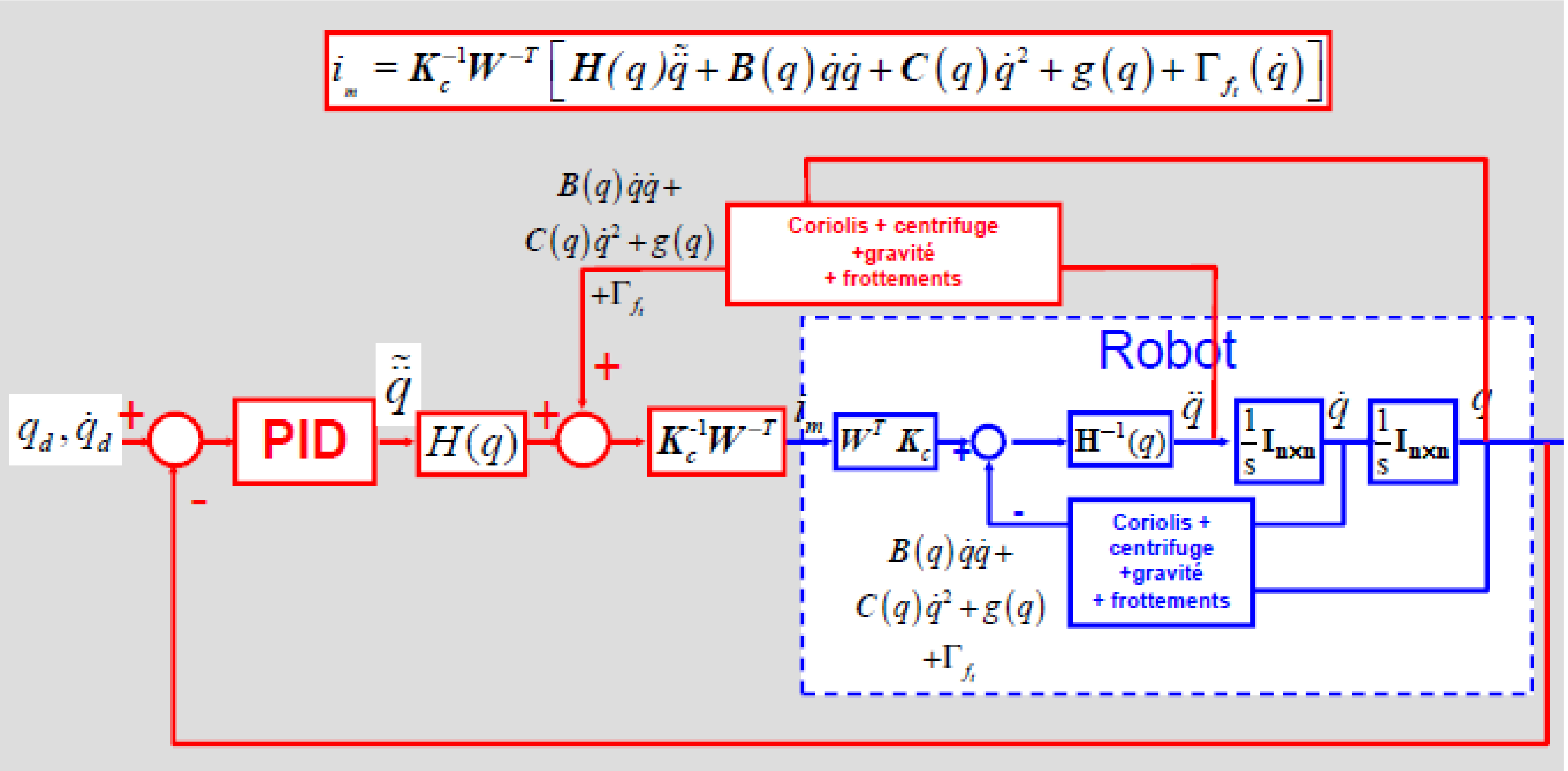
Useful only for fast movements (i.e. movements in which inertial coupling effects affect tracking precision)

Si un modèle dynamique du robot est connu, on peut calculer la commande i_m^* qui correspond à la trajectoire désirée $q_d, \dot{q}_d, \ddot{q}_d$:

$$i_m^* = K_c^{-1} W^{-T} \left[H(q_d)\ddot{q}_d + B(q_d)\dot{q}_d\dot{q}_d + C(q_d)\dot{q}_d^{\,2} + g(q_d) + \Gamma_{f_s}(\dot{q}_d) \right]$$



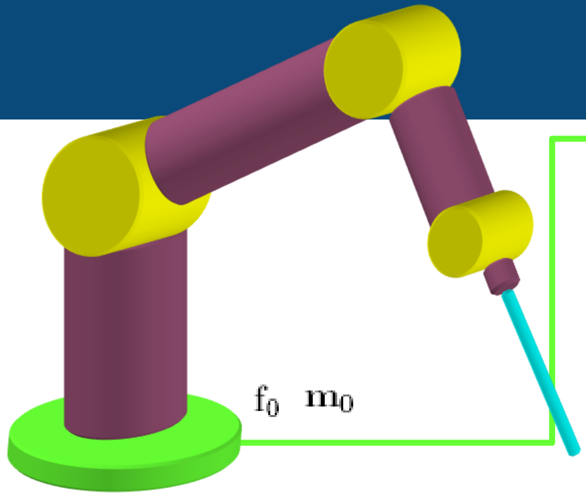
A DECOUPLING APPROACH



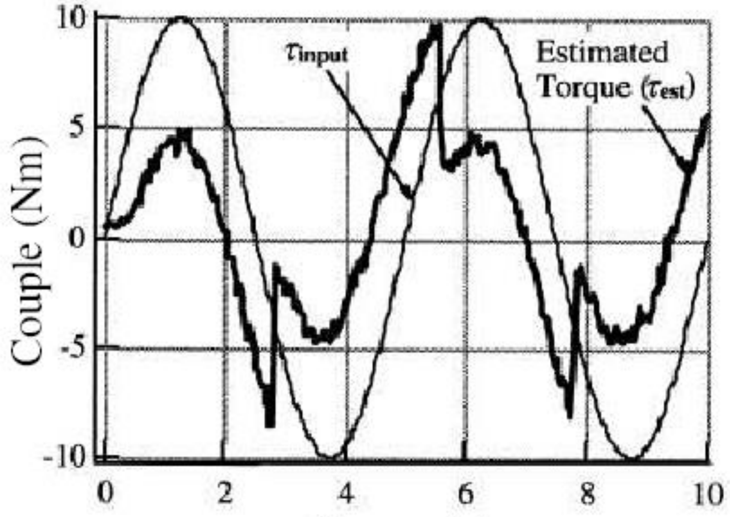
Practical problems in joint space position servoing with inner current torque loop

- Gravity disturbance and coupled dynamics may be easily compensated for (thanks to an identification effort).
- Position sensor resolution may be an issue
- Friction is the real enemy
  - Models very simplistic or very difficult to identify
  - Changes with time and conditions (load, temperature)
  - Inner joint torque loops helps dealing with it

Illustration of the benefits of a torque loop



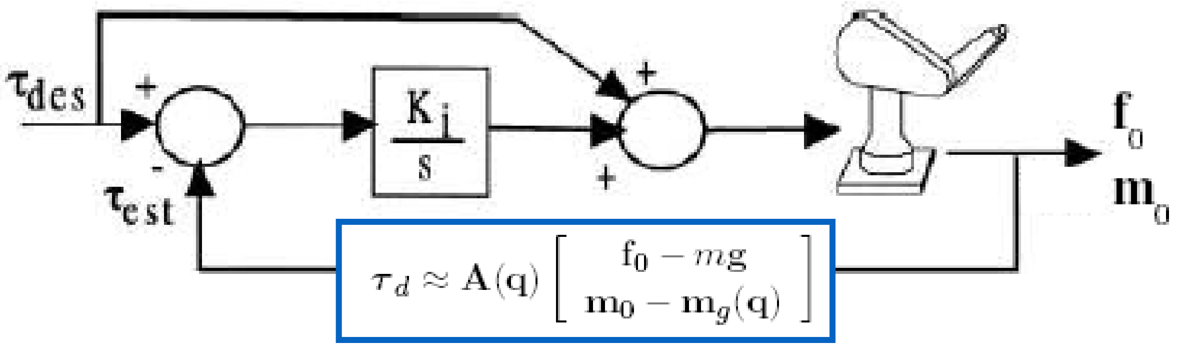
$$\tau_d = \mathbf{A}(\mathbf{q}) \begin{bmatrix} \mathbf{f}_0 - m\mathbf{g} \\ \mathbf{m}_0 - \mathbf{m}_g(\mathbf{q}) \end{bmatrix} - \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$$

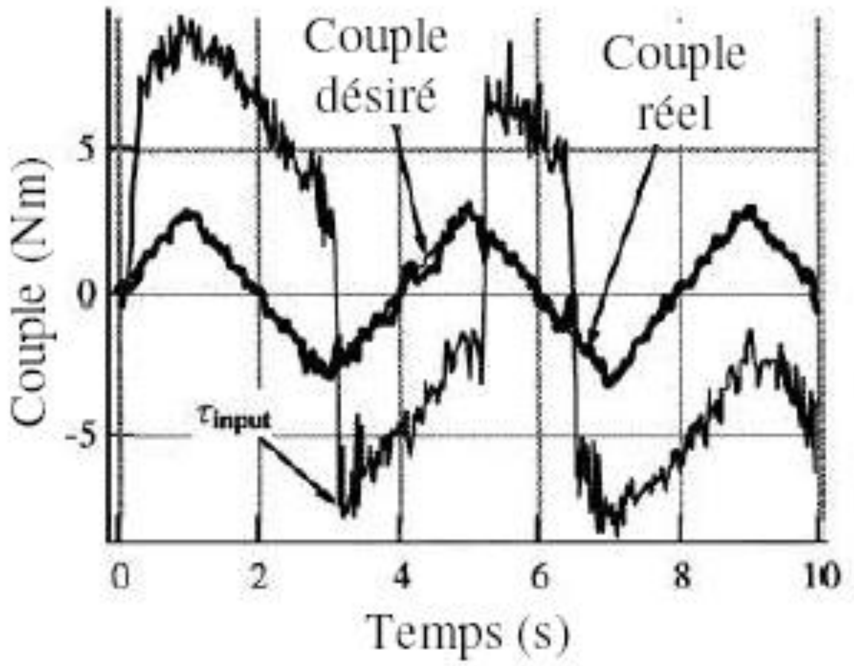


$$\tau_d \approx \mathbf{A}(\mathbf{q}) \begin{bmatrix} \mathbf{f}_0 - m\mathbf{g} \\ \mathbf{m}_0 - \mathbf{m}_g(\mathbf{q}) \end{bmatrix}$$

USING A TORQUE LOOP FOR FINE MANIPULATION

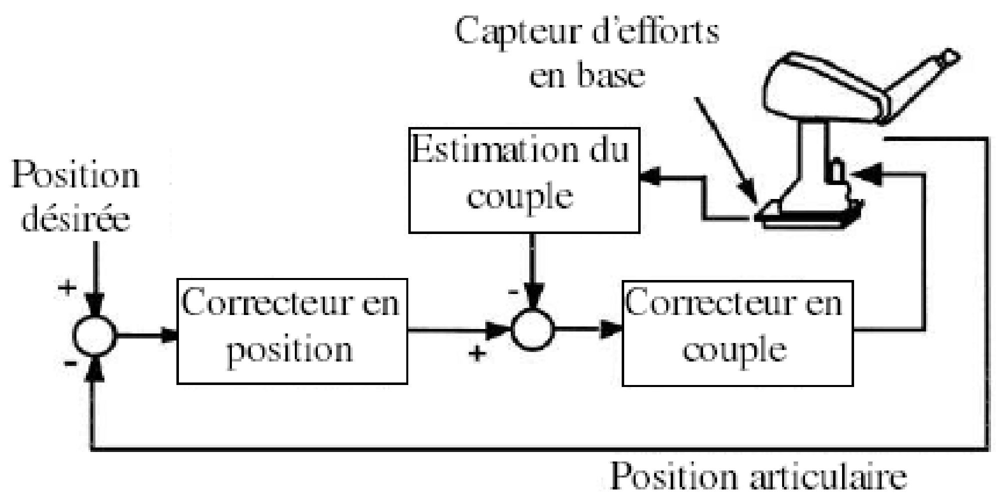
- Inner torque loop:

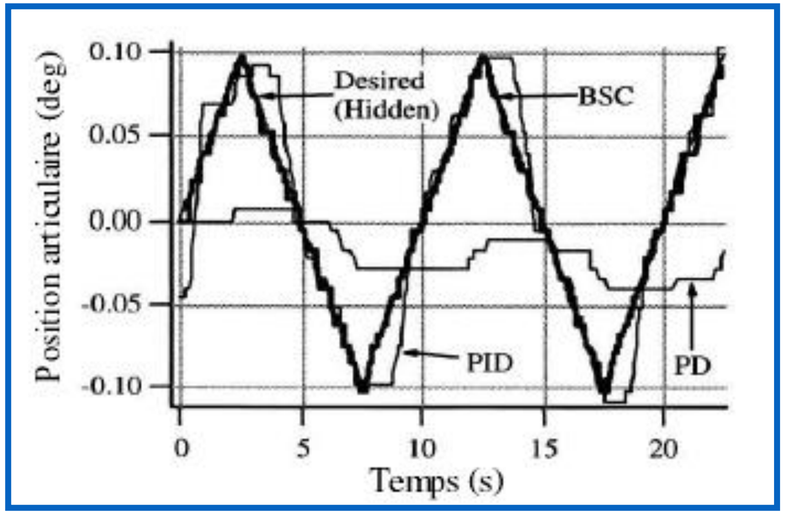




Using a torque loop for fine manipulation

- Outer position loop



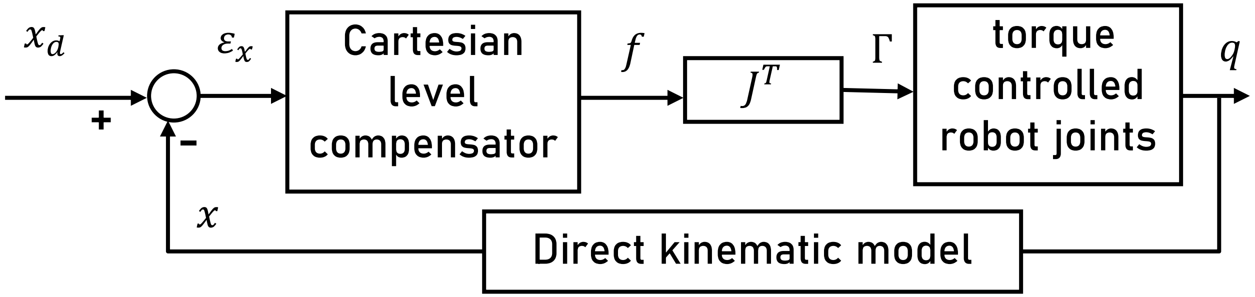


Using a torque loop for fine manipulation

Practical problems in joint space position servoing with inner current torque loop

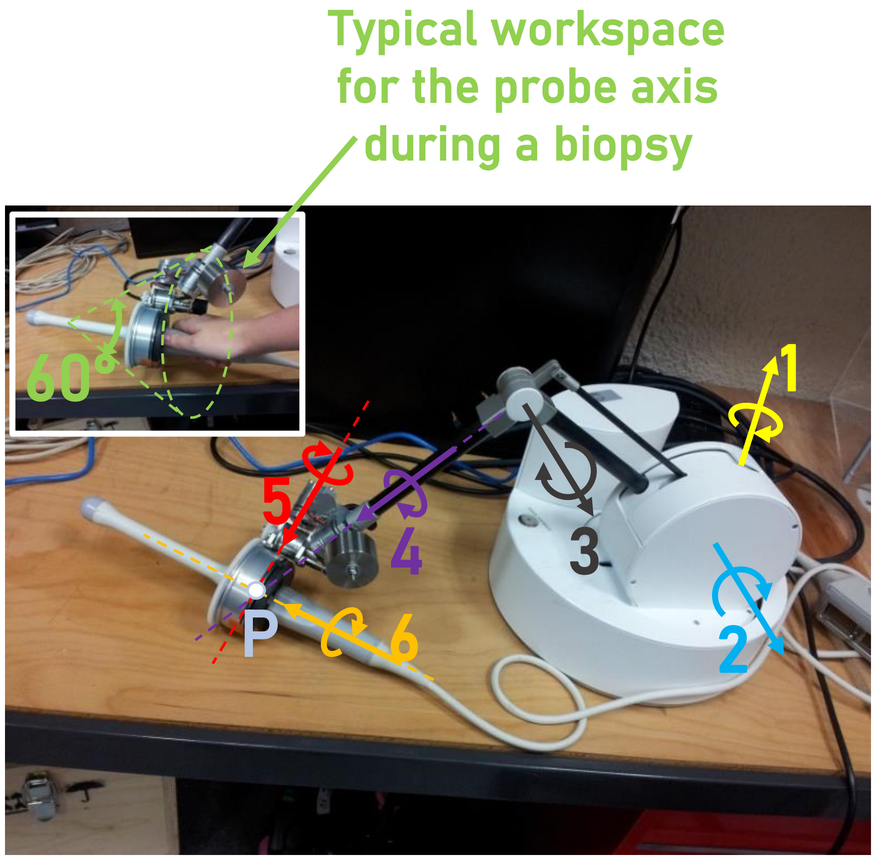
- Gravity disturbance and coupled dynamics may be easily compensated for (thanks to an identification effort).
- Position sensor resolution may be an issue
- Friction is the real enemy
  - Models very simplistic or very difficult to identify
  - Changes with time and conditions (load, temperature)
  - Inner joint torque loops helps dealing with it
  - But it requires a model for gravity (and dynamics) compensation. This can be done with observers. Errors in these models induce position tracking errors

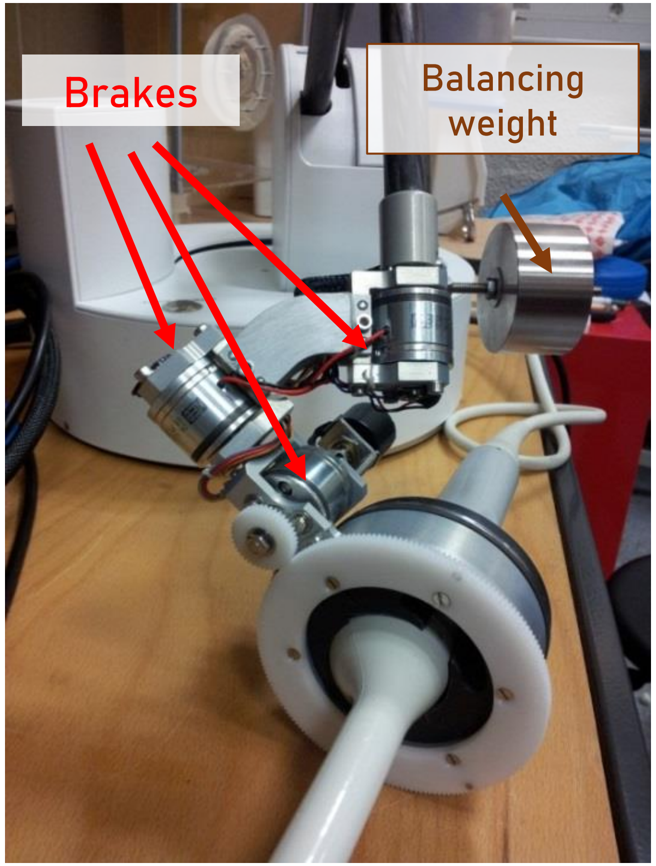
Operational space motion control with torque controlled actuators



- Where we used the fact that : $\Gamma = J^T_{NO_n}(q)\, f$, f begin the force/torque vector grouping the
- The closed loop dynamic is tuned at the operational space level. Quite difficult due to large variation in operational space dynamics ➔ A PI(D) compensator will work only with low gains.
- This approach is sometimes viewed as impedance control, because the compensator is a mechanical impedance : K_p as an operational stiffness and K_d as an operational viscosity

Precise yet soft : the exemple of Apollo



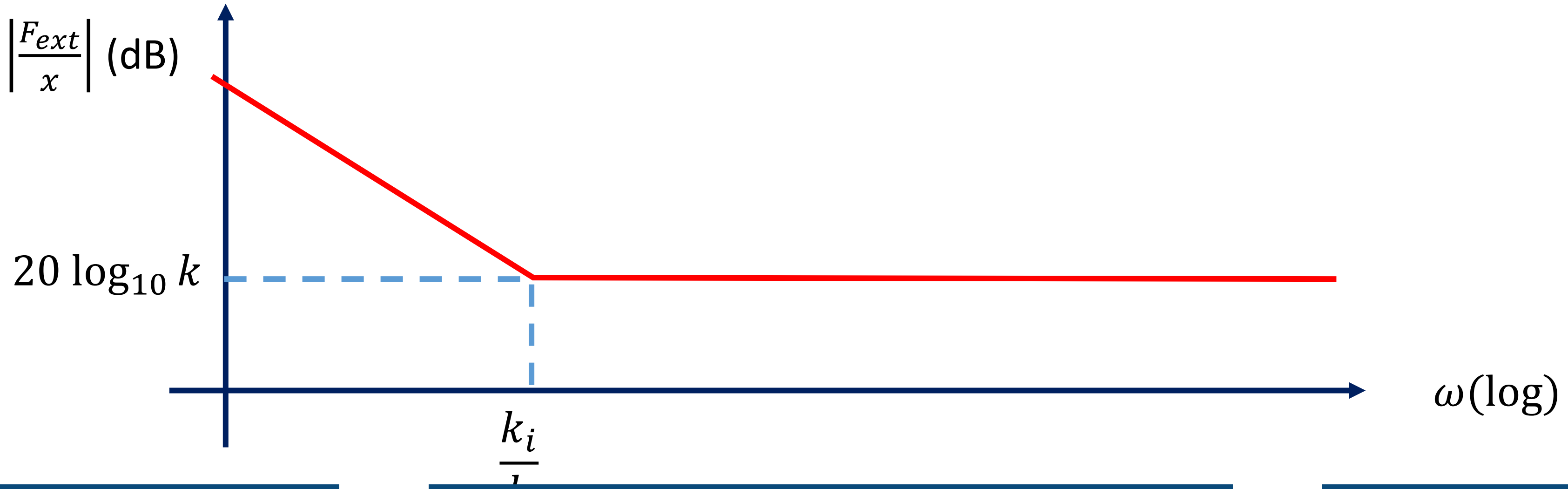


- Endorectal ultrasound probe manipulator (prostate biopsies)
- 6 degrees of freedom
- 3 first joints motorized (haptic interface technology – cable transmission)
- 3 last joints w/ brakes

Poquet et al., IROS 2013

PI CONTROL WITH INNER TORQUE CONTROL: PRECISION – STIFFNESS

- $f_{actuator} + f_{distrubance} = m\ddot{x}$
- PI comp: $f_{actuator} = k(x_d - x) + k_i \int_0^t (x_d - x)d\tau$
- Equilibrium: $\forall f_{distrubance}, x = x_d$
- Apparent stiffness vs frequency (at low freq – inertia neglected):



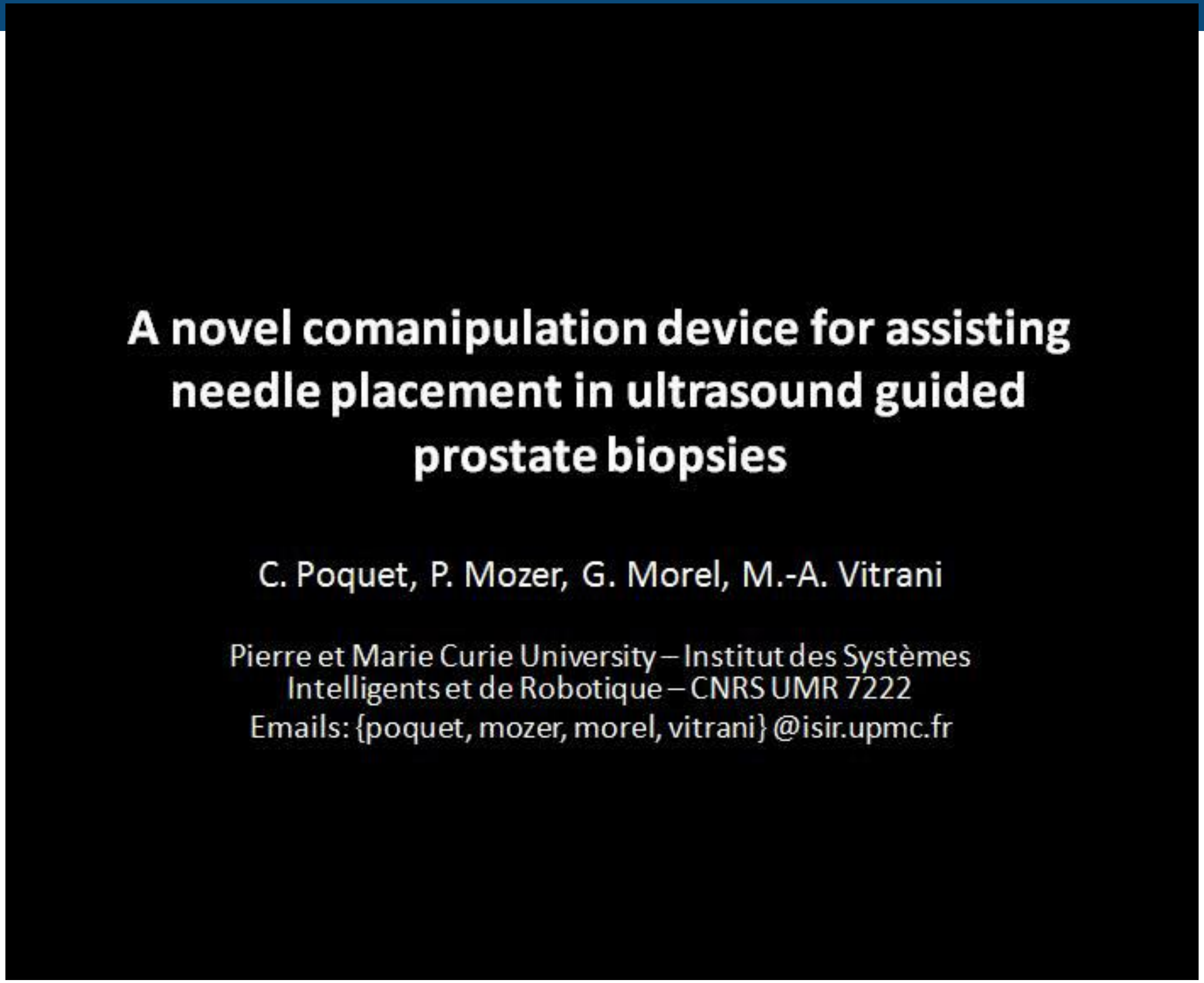
Locking Apollo: soft and precise

- Goal : lock the target site for a prostate biopsy during a TRUS examination.
- The patient is awake and may move: softness required.
- A 1mm precision is required.
- Large external forces due to patient-probe interaction.
- A PI controller tuned with low k and very low k_i for the three fist joints, brakes for the three last joints:

$$k = 200\text{Nm}^{-1} \text{ and } \frac{k_i}{k} = 4 \text{ rad s}^{-1} \ \ (0.6 \text{ Hz})$$

- The three first joints also compensate for the flexion measured in the three last joints (brakes don't exhibit an infinite stiffness).

LOCKING APOLLO: SOFT AND PRECISE

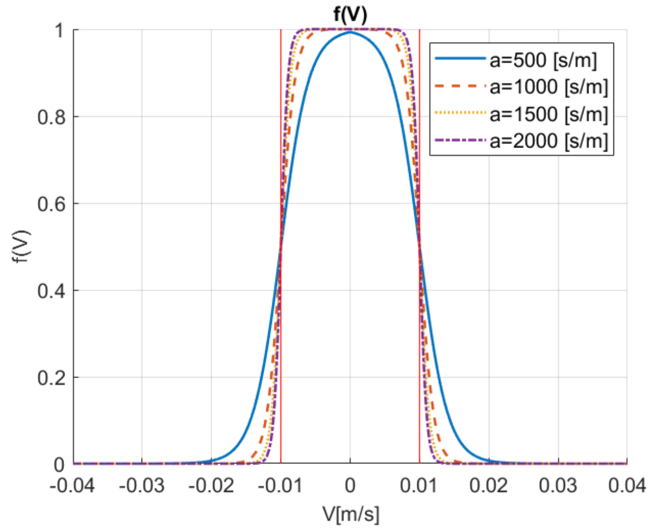


POQUET et al, IROS 2013

DEALING WITH EXTERNAL FRICTION IN TELEOPERATED MODE FOR DYNAMIC PRECISION

- Standard PID $\mathbf{F}_u = K_P \cdot \mathbf{e} + K_D \cdot \dot{\mathbf{e}} + K_I \int_0^t \mathbf{e}\, dt + \mathbf{F}_g$
- When using low gains, the tracking performance is highly degraded with friction, specifically when starting / stopping a movement
- Modified integrator:

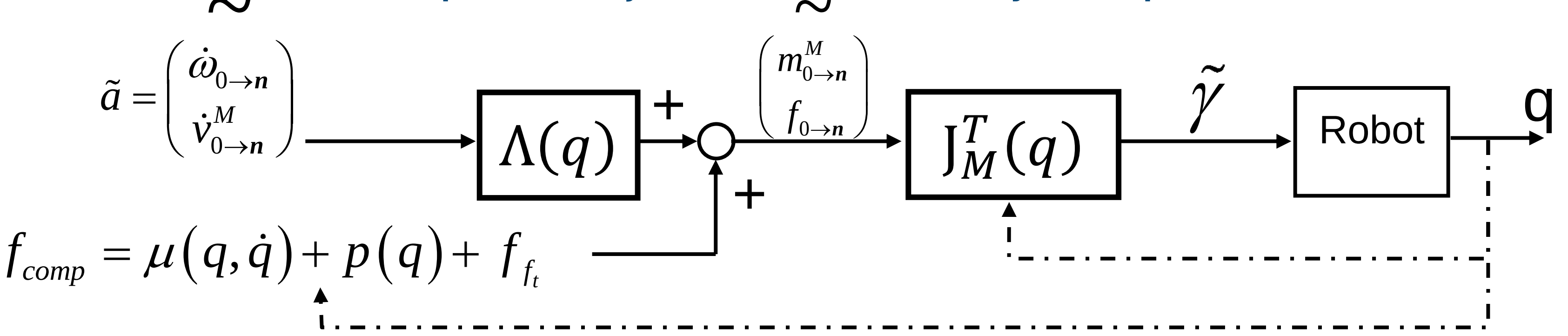
$$\mathrm{I} = \underbrace{K_I \int_0^t \mathbf{e}\, dt}_{\text{Standard integrator}} \underbrace{+\alpha K_I \int_0^t \frac{|\mathbf{e}|\mathbf{e}}{1 + \exp(a(|\mathbf{V}| - V_o))} dt}_{\text{Non-linear integrator}}$$



Result in a video

Operational space dynamic control

- On uses matrix $\Lambda(q)$ to decouple the dynamics, as in the joint space.



- This allows to have a new control in $\tilde{a} = \begin{pmatrix} \dot{\omega}_{0\to n} \\ \dot{v}^M_{0\to n} \end{pmatrix}$

- The compensator then writes: $\tilde{a} = \boldsymbol{K}\begin{pmatrix} \theta\, {}^0\boldsymbol{u} \\ {}^0\boldsymbol{t}_{T\to T_d} \end{pmatrix} - \boldsymbol{B}\begin{pmatrix} {}^0\omega_{n/0} \\ {}^0v^M_{n/0} \end{pmatrix}$ (warning: K is not a stiffness)

- This results in a decoupled operational space dynamics

PART 3: CONTROLLING ROBOTS IN CONTACT WITH THEIR ENVIRONMENT

Why?

- So far we've seen how to position a robot from either proprioceptive sensors (conventional robot position control from joint sensors with registration) or exteroceptive sensors (motion rate control)
- When a contact is to be established with the environment, forces appear. The resolution of a position sensor is insufficient to master forces (if the stiffness is high).
- Force control consists in controlling the force applied by a robot rather than the position

Controlling a robot in contact with its environment

- Impedance control
  - In fact a position controller : $f = K_d(x_d - x) - B_d\dot{x}$
  - But K_d and B_d are the focus.
  - Works only for low stiffness.
  - Initially (Hogan, 1985) formulated as: $f = M_d\ddot{x} + K_d(x_d - x) - B_d\dot{x}$, but in practice, rather unstable

$$
\begin{aligned}
\mathrm{Tact} = {} & I(\theta)\mathbf{J}^{-1}(\theta)M^{-1}K[\mathbf{X}_0 - L(\theta)] + S(\theta) \text{ (position terms)} \\
& + I(\theta)\mathbf{J}^{-1}(\theta)M^{-1}B[\mathbf{V}_0 - \mathbf{J}(\theta)\omega] + V(\omega) \text{ (velocity terms)} \\
& + I(\theta)\mathbf{J}^{-1}(\theta)M^{-1}\mathbf{Fint} - \mathbf{J}^{t}(\theta)\mathbf{Fint} \text{ (force terms)} \\
& - I(\theta)\mathbf{J}^{-1}(\theta)G(\theta,\omega) + C(\theta,\omega) \text{ (inertial coupling terms)}
\end{aligned}
$$

  - To stabilize the system across the workspace, B_d can be made variable with respect to inertial variations in order to keep a constant damping.

Applying a desired force under impedance control

- With an impedance controller $f = K_d(x_d - x) - B_d\dot{x}$, one can apply a desired force f_d only if x_e is known.
- Option 1:
  - When an environment is contacted, a rest: $K_d(x_d - x) = f$ (up to gravity compensation errors & static friction distrubances)
  - Set

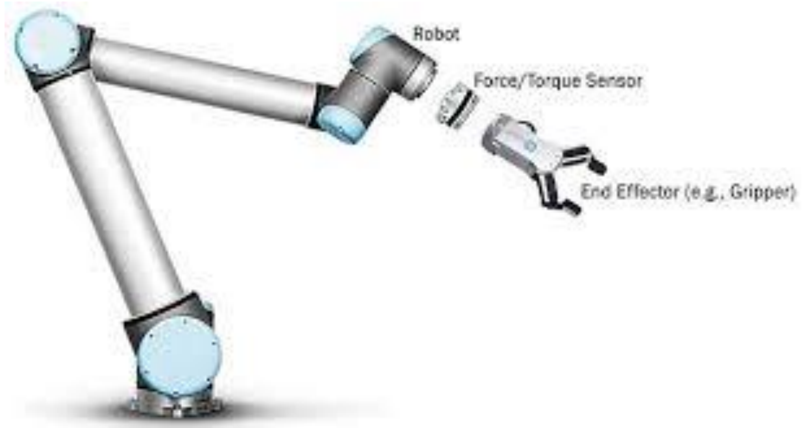
$$x_d = K_d^{-1} f_d + x_e$$

- Option 2:
  - Add a feedforward term in the :$f = K_d(x_d - x) - B_d\dot{x} + f_d$
  - Set

$$x_d = x_e$$

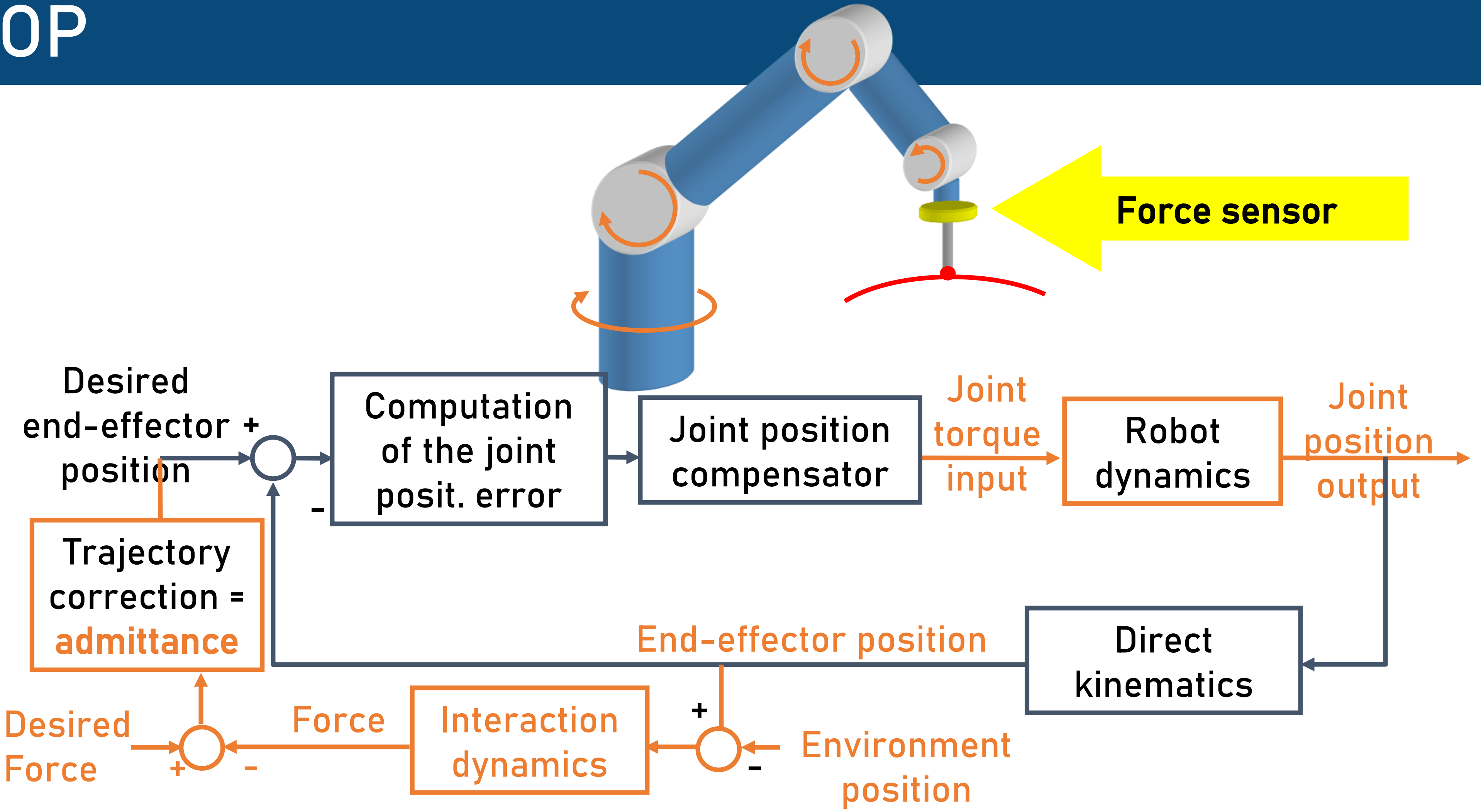
Closed-loop force control

- One usually exploits a distal force sensor to measure the interaction.

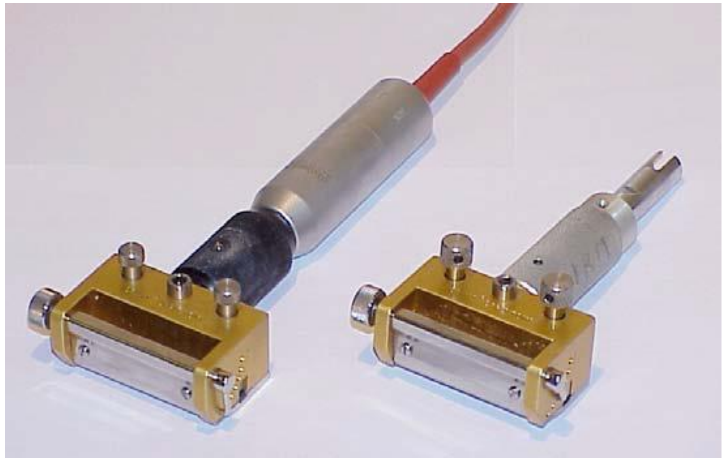


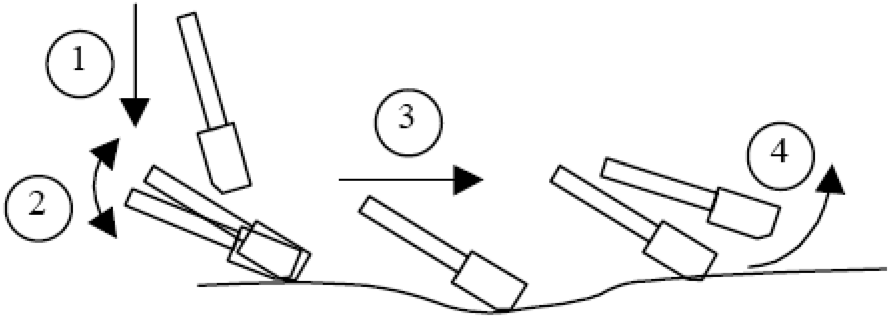
- Two approaches
  - Indirect:
    - The robot is a positioner (*i.e.* there is a position control loop)
    - An external force loop changes the desired position according to the measured force
  - Direct
    - No inner position loop
    - A force error generates a joint torque command, directly

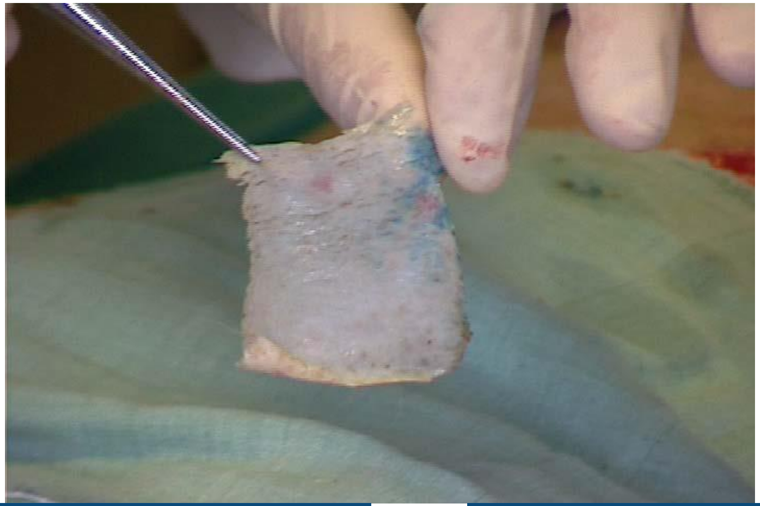
Force control with an internal position loop



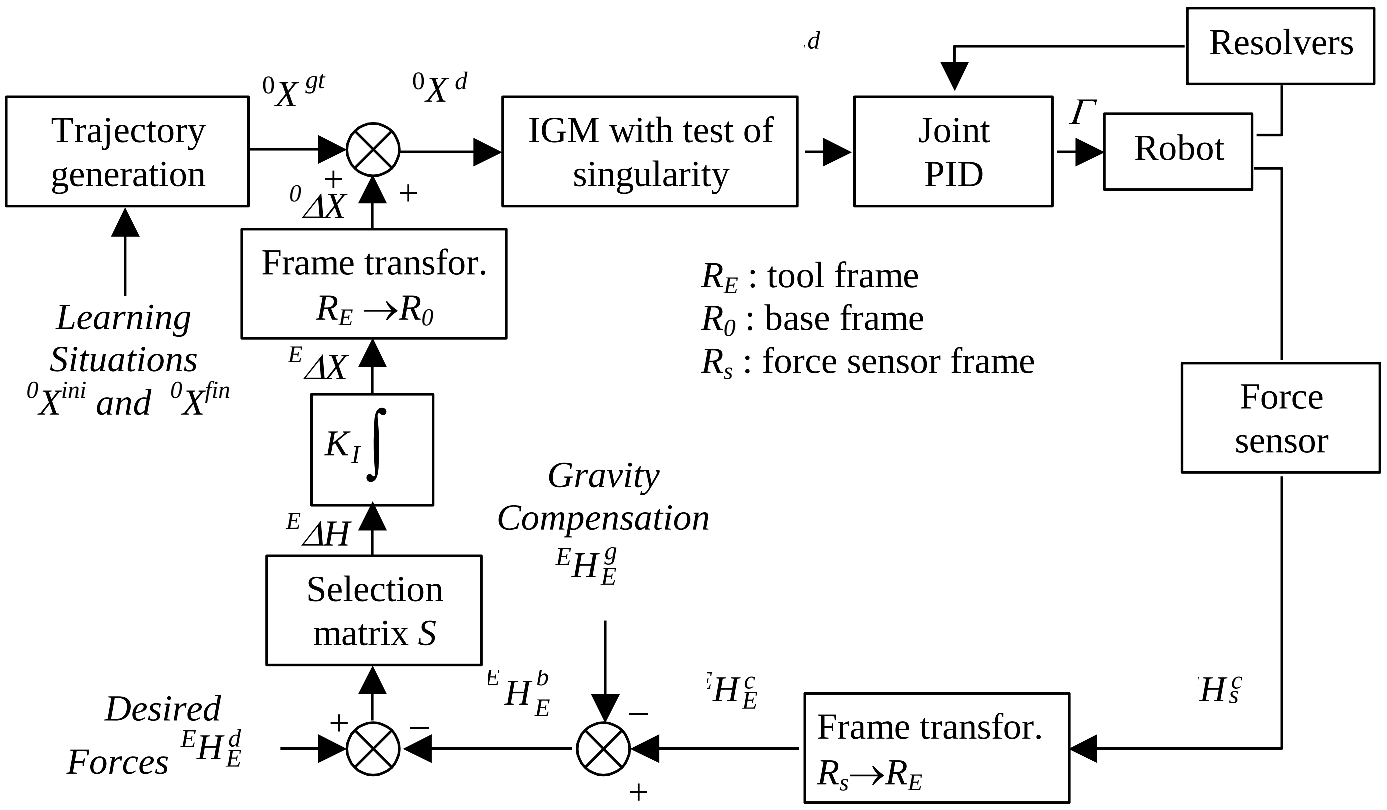
Example: Dermarob







Control law



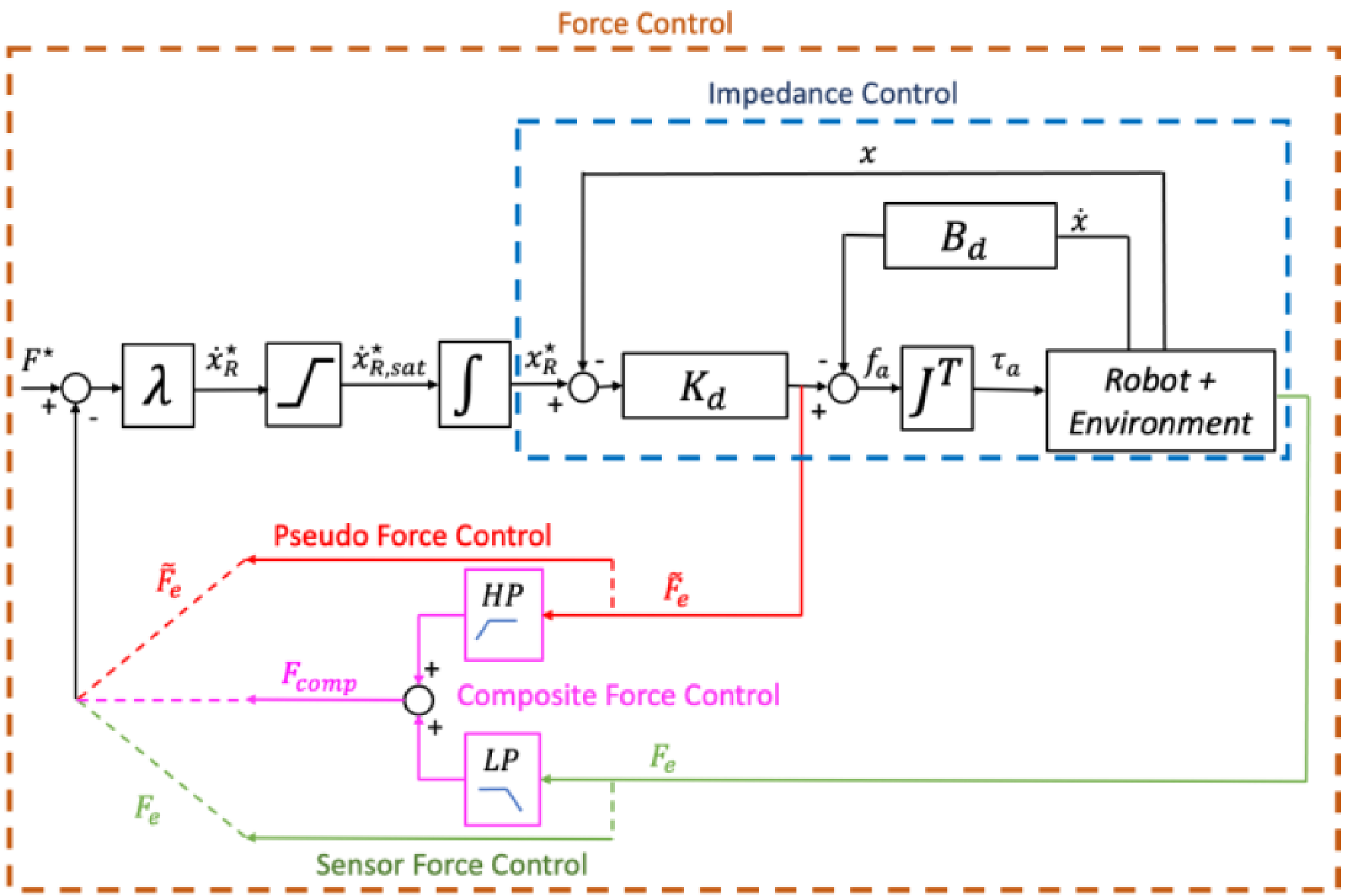
Credit: E. Dombre – Montpellier.

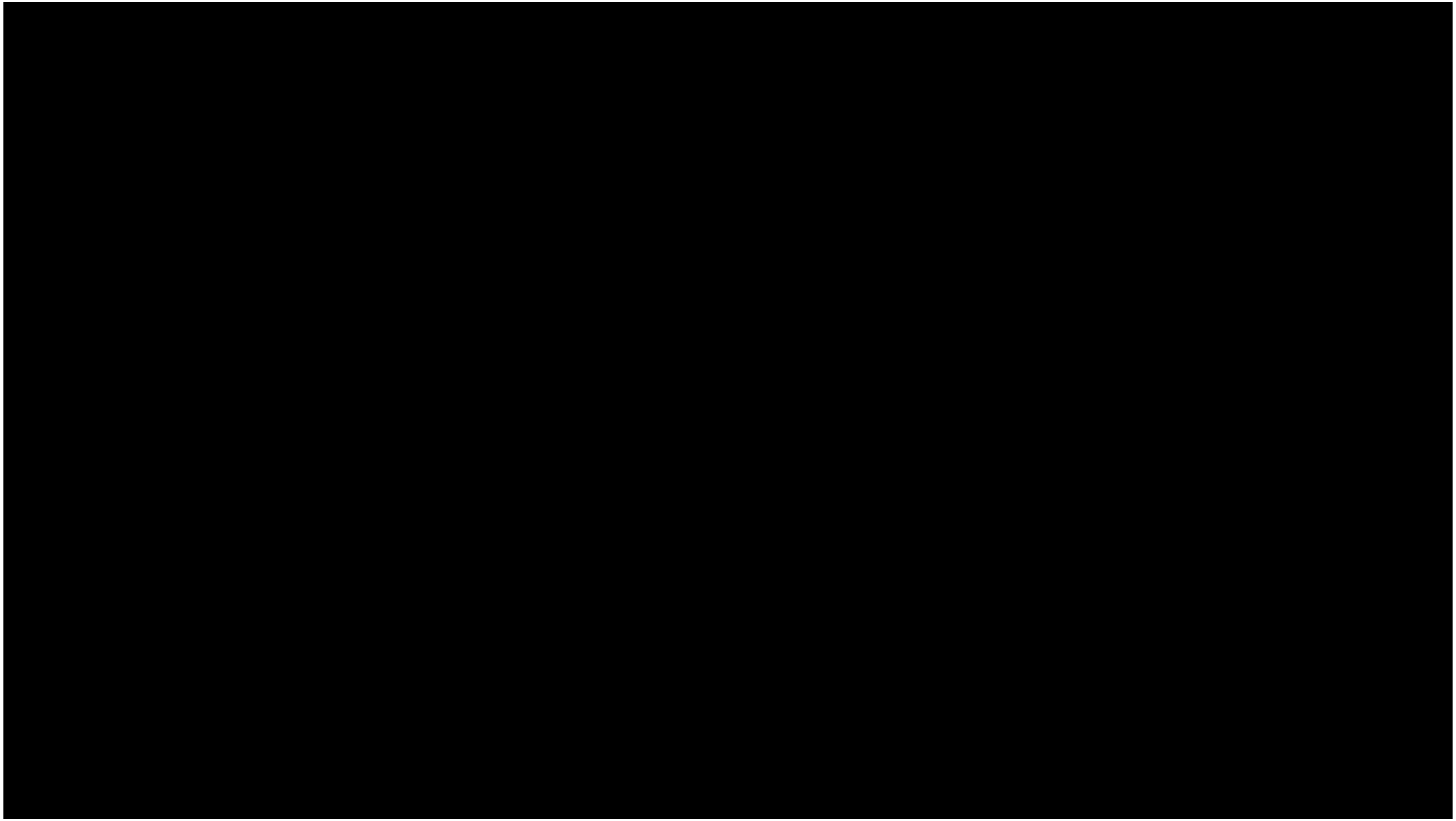
Video

Stability issues

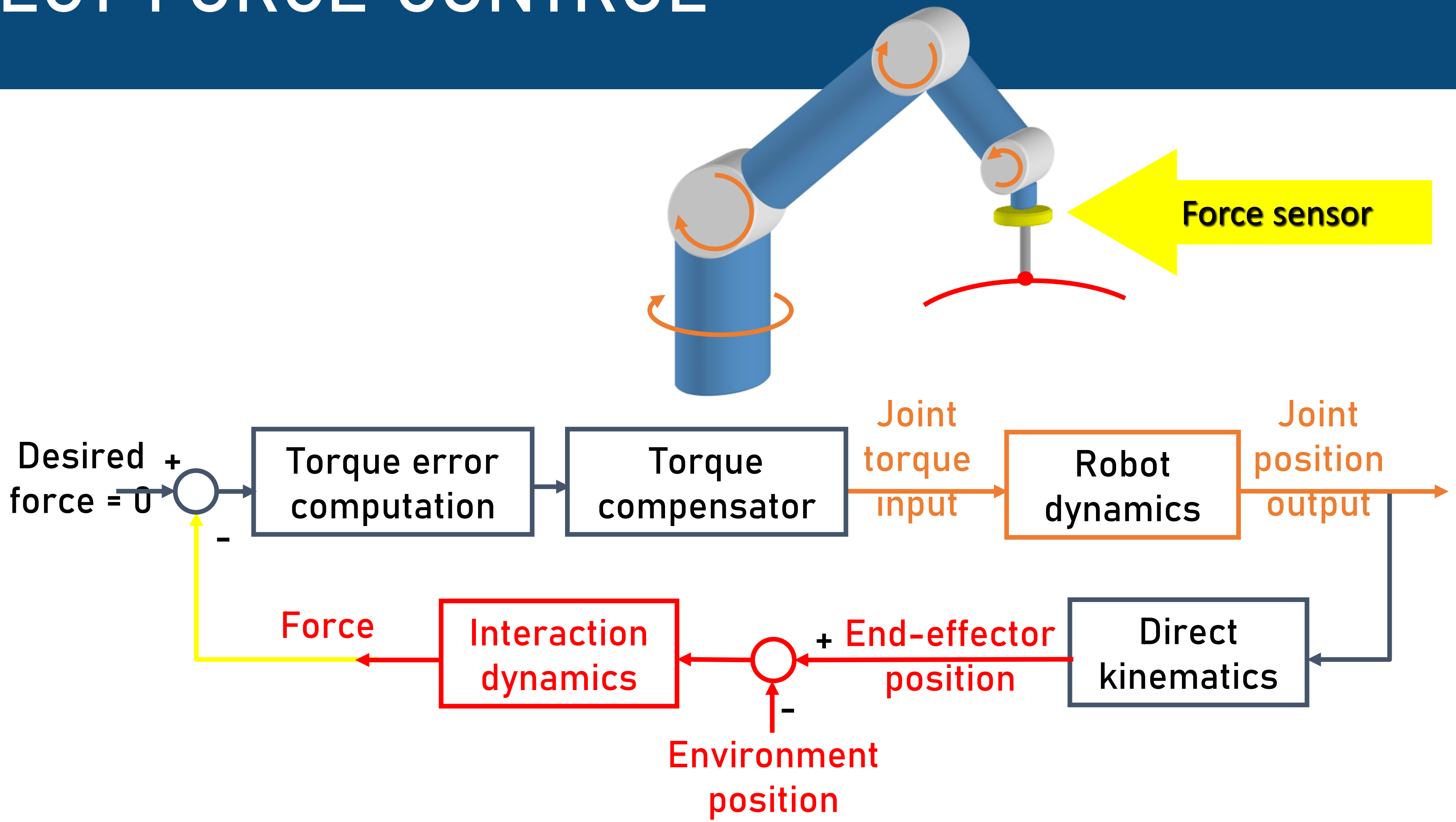
- Adding a distal force sensor allows for finely control the interaction
- But it creates so-called *non collocated vibration modes*. This induces bandwidth limitations
- Using joint torque sensors or pseudo force estimation does not produce this problem
- But it is not precise for estimating the distal force

Exploiting a complementary filter





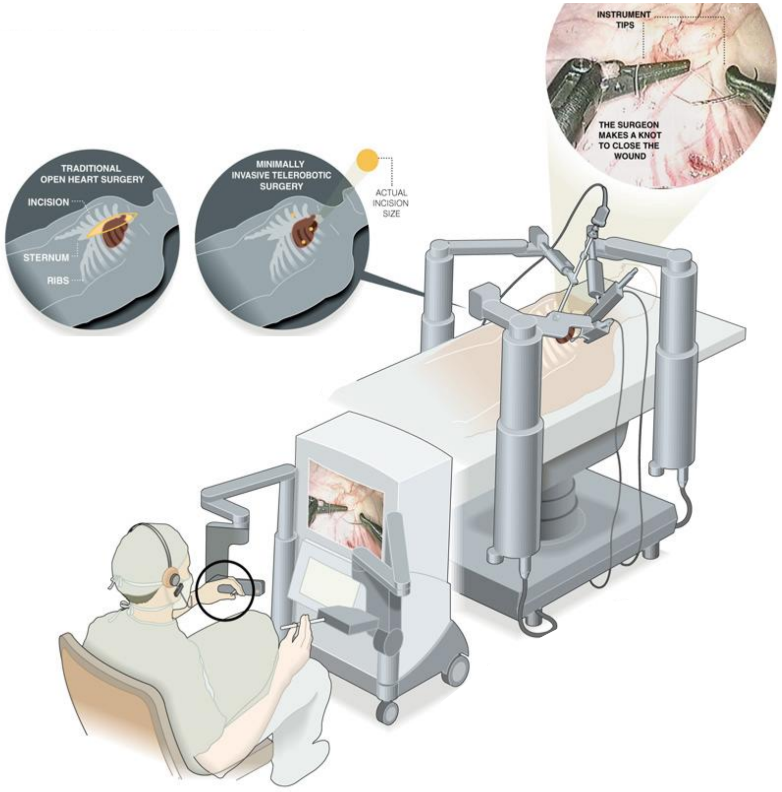
DIRECT FORCE CONTROL

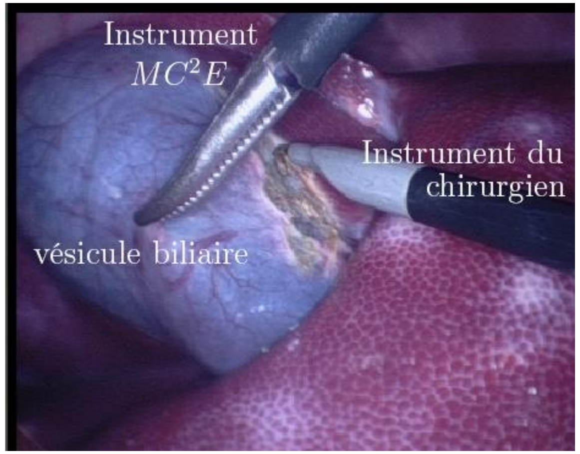


Example: force control for laparoscopic surgery

Motivatons

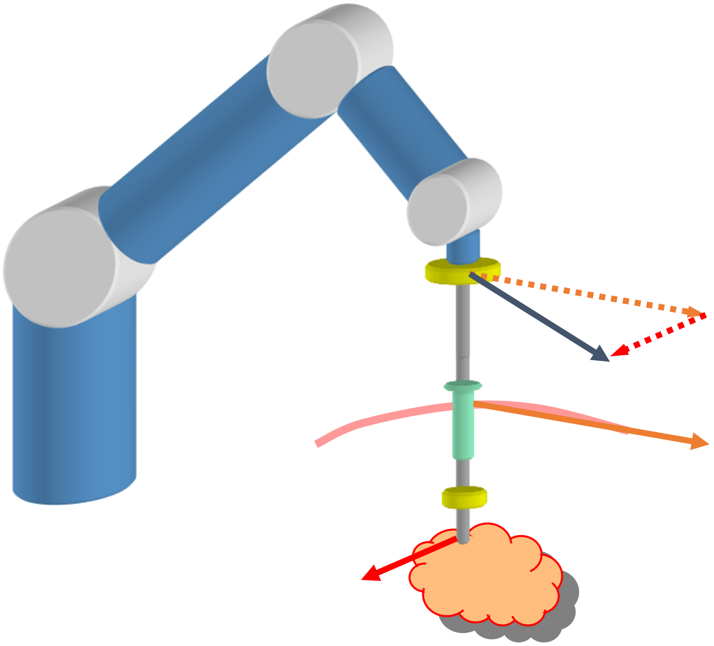
- Force feedback teleoperation.
- Physiological motion compensation.
- Autonomous assistance :
  - Tension control when suturing.
  - Holding organs during their ablation.

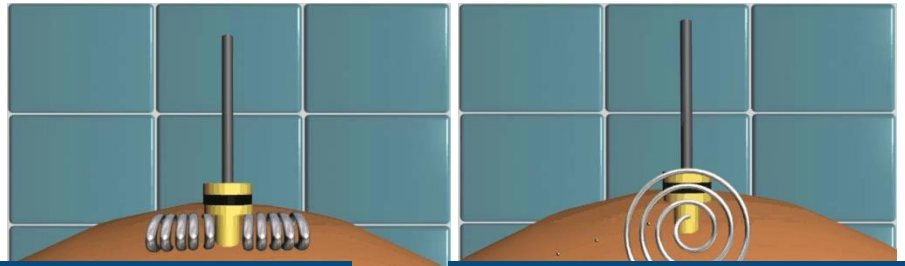




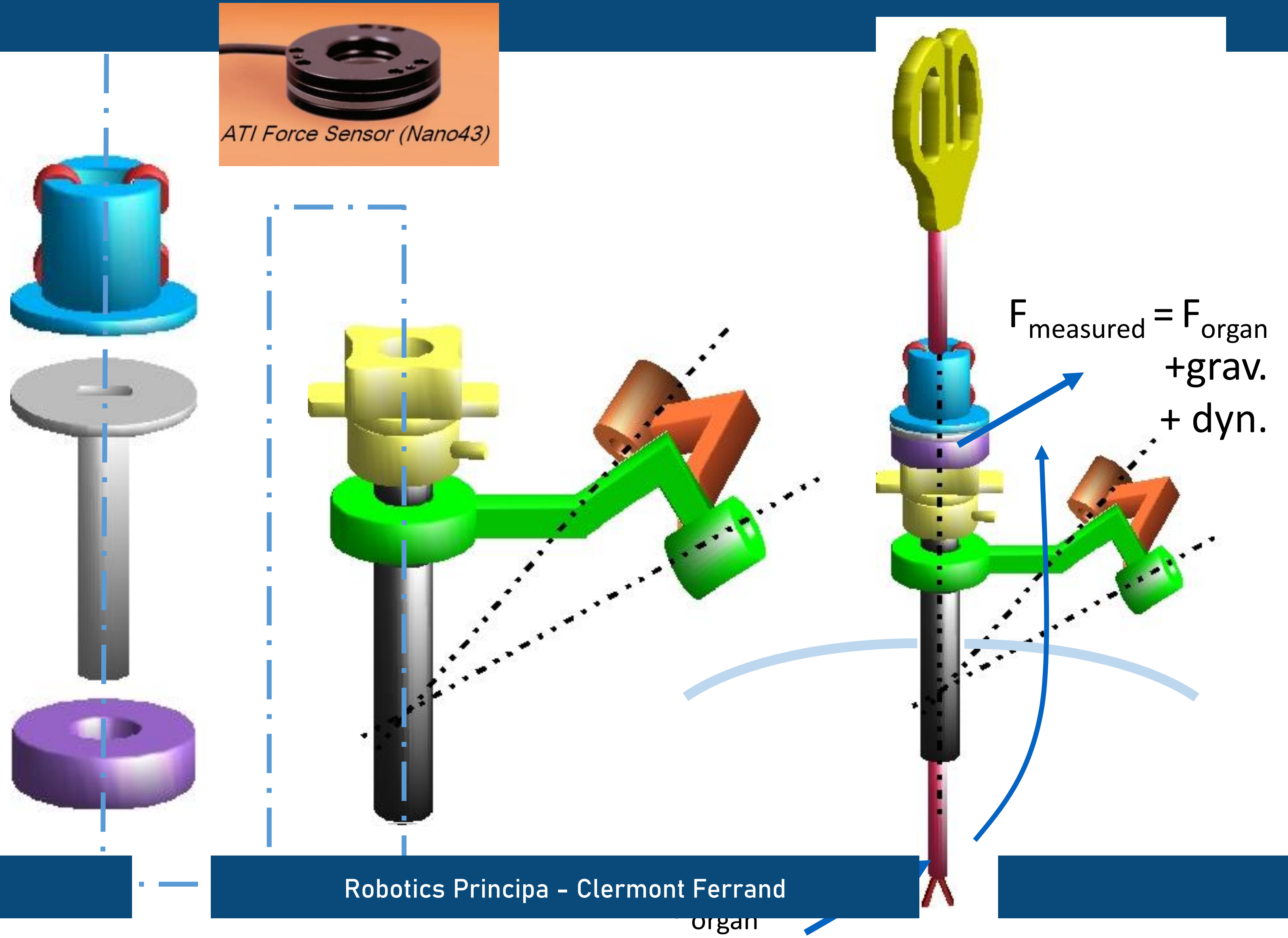
Measuring forces in laparoscopic surgery

- Sensing forces between instrument and organs is required.
- Precise-and-small enough sensors do not exist
- Wrist sensing leads to a corruption of the measure due to **trocar disturbances**



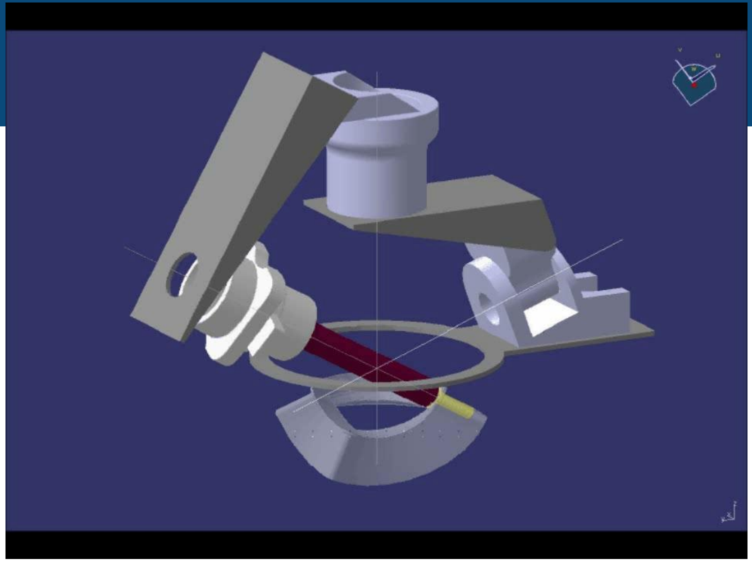


A MECHATRONIC SOLUTION

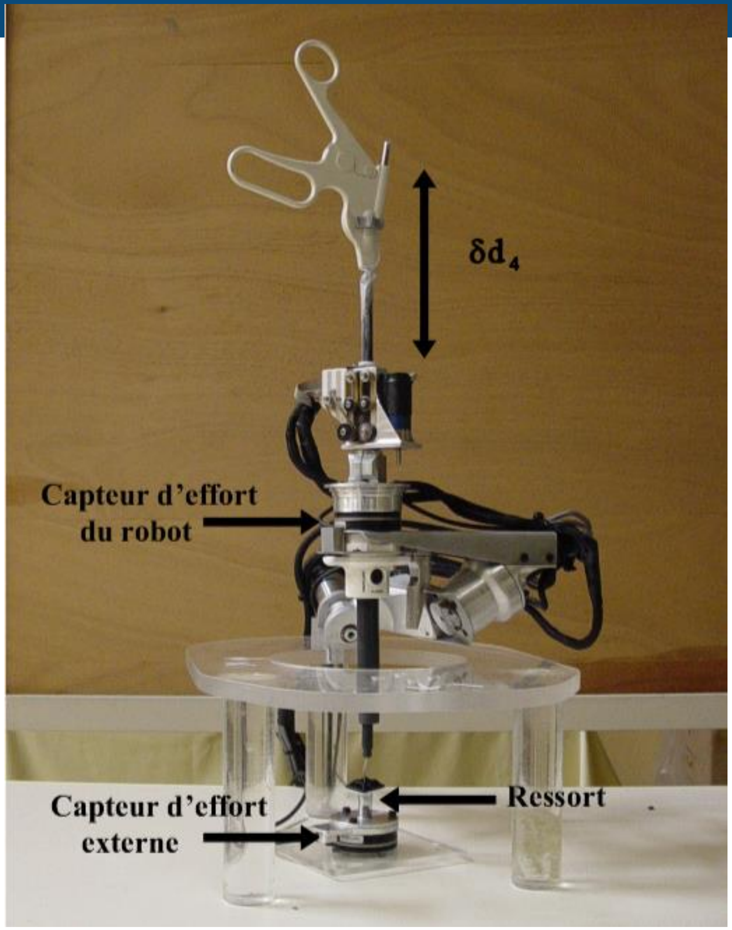


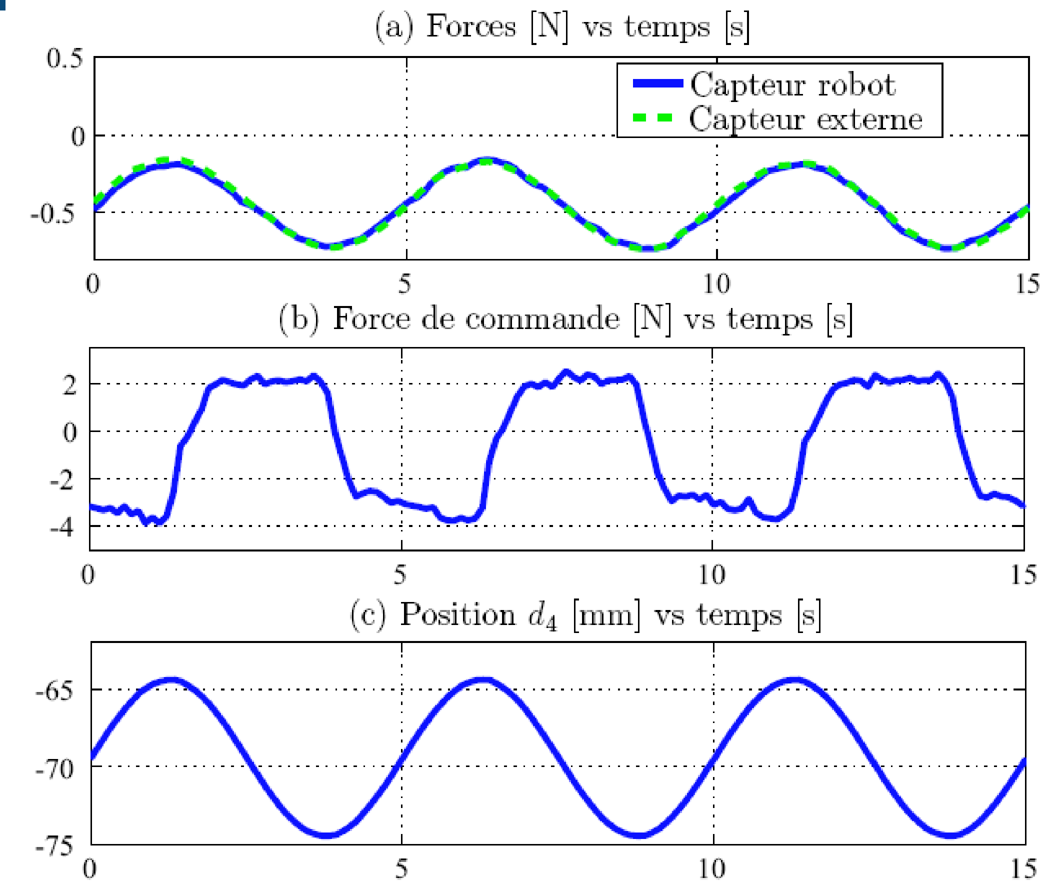
Prototyping

- 4 DOF spherical kinematics.
- 2 first joints for orienting the trocar around the fulcrum point.
- 2 last joints for relative motion of the instrument w.r.t. the trocar.
- ATI nano43 force sensor in the middle of the kinematic chain.



Force measurement verification

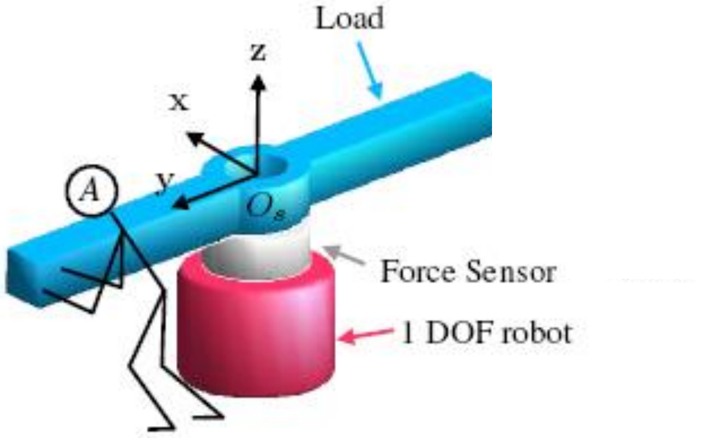


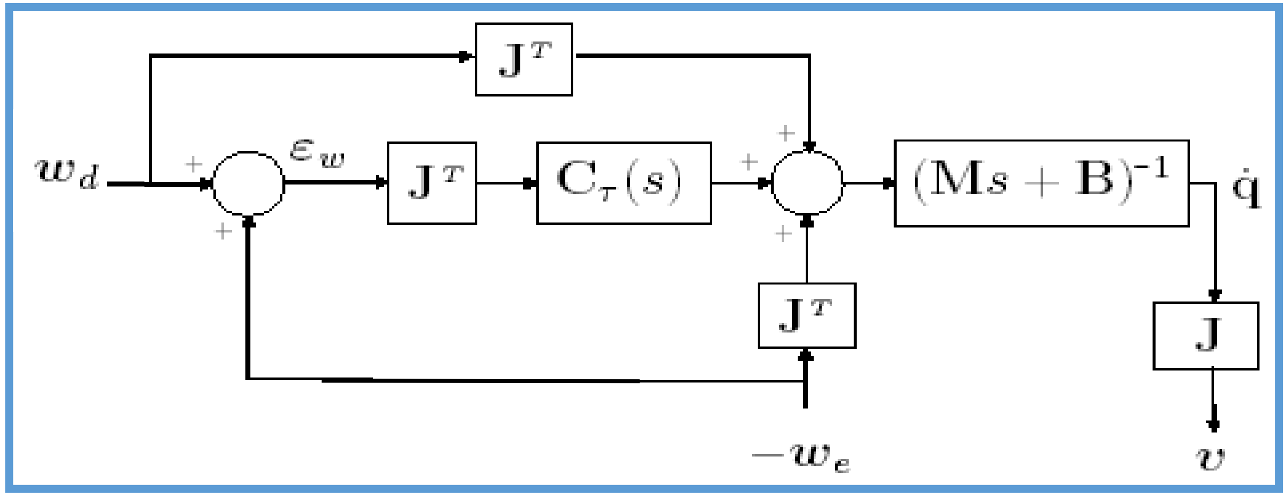


Control problem: selecting the force components

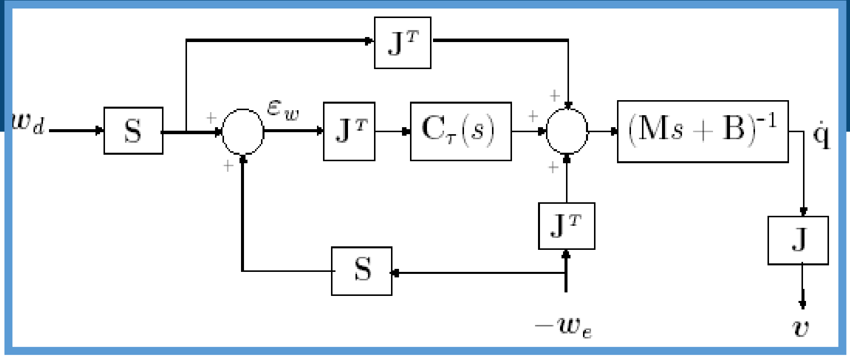
- 4 DOF robot; 6 component wrench measurement *w*.
- No idea where the contact is.
- What error should be servoed ?
  - Component selection: $\varepsilon_R = R_{4\times 6}(w_d - w)$
  - Joint space projection:

$$\varepsilon_\tau = J^T (w_d - w)$$





Formal analysis



$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_d + \underbrace{\left(\mathbf{K}_p + \frac{\mathbf{K}_i}{s}\right)}_{:=\mathbf{C}_{\boldsymbol{\tau}}(s)} (\boldsymbol{\tau}_d - \boldsymbol{\tau}_e)$$

- Interaction port admittance:

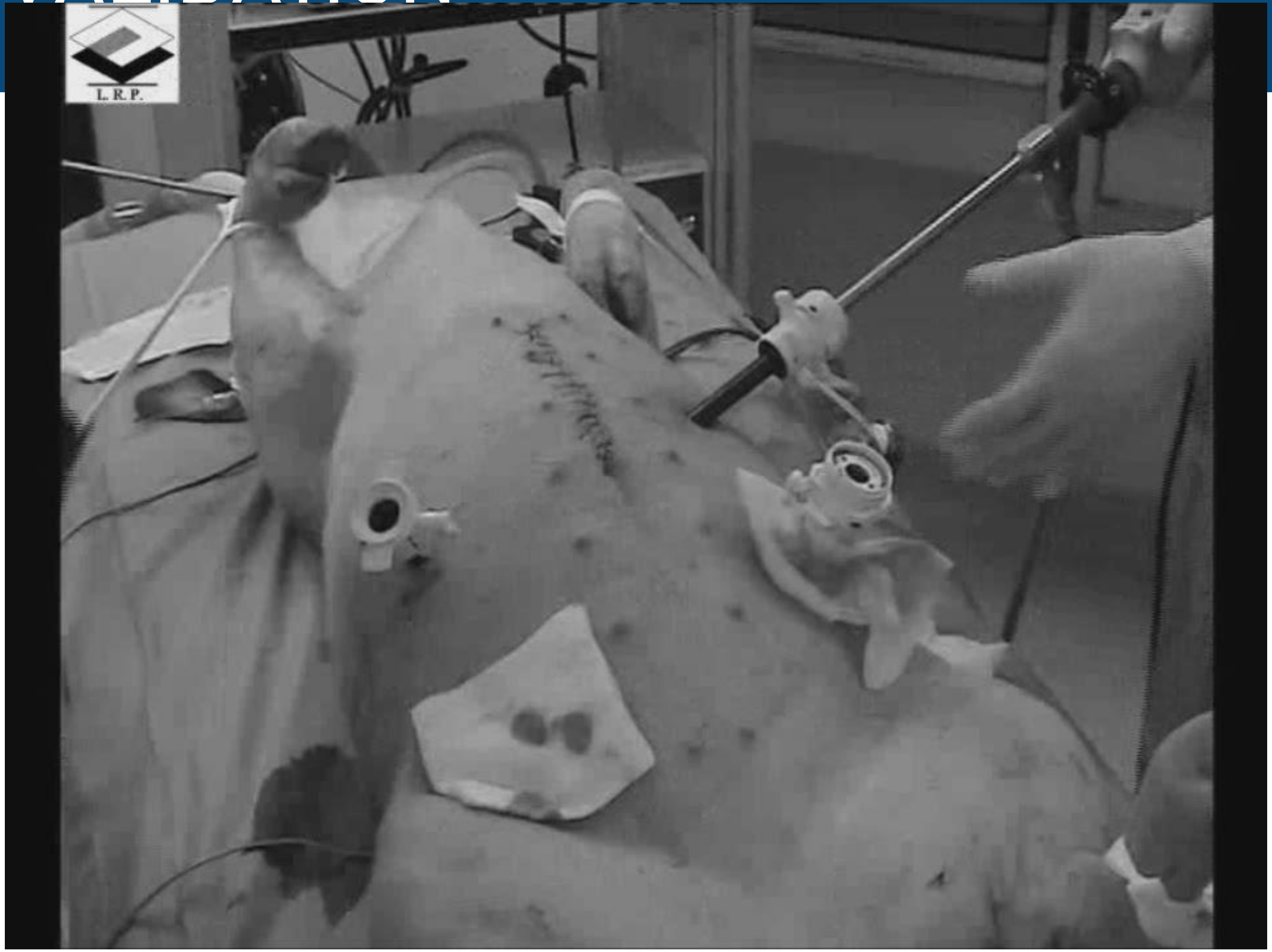
$$\mathbf{Y}_\mathbf{w}(s) = \frac{\mathbf{v}}{-\mathbf{w_e}} = \mathbf{J}\mathbf{Y}_r(s)\left[\mathbf{J}^T + \mathbf{C}_{\boldsymbol{\tau}}(s)\mathbf{J}^T\mathbf{S}\right]$$

- Passivity conditions :

a) $B^{-1}K_i$ is PSD.
b) $M = K_p\ M\ K_p^{-1}$.
c) $(I_n + K_p)B - K_iM$ is PSD.
d) $BK_i = K_iB$.

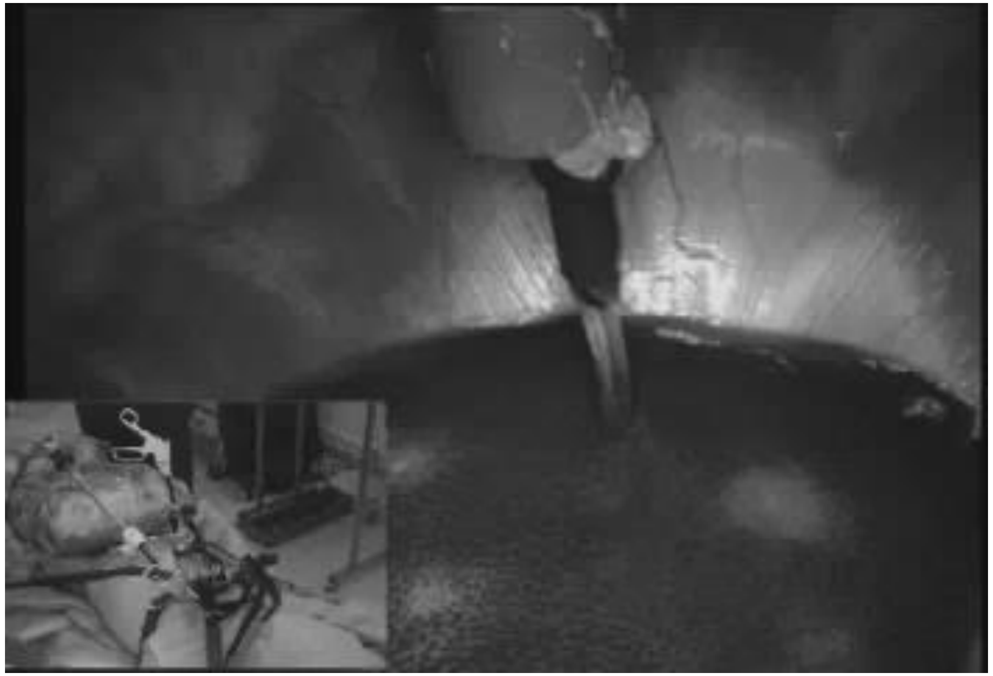
$$\mathbf{SJ} = \mathbf{J}$$

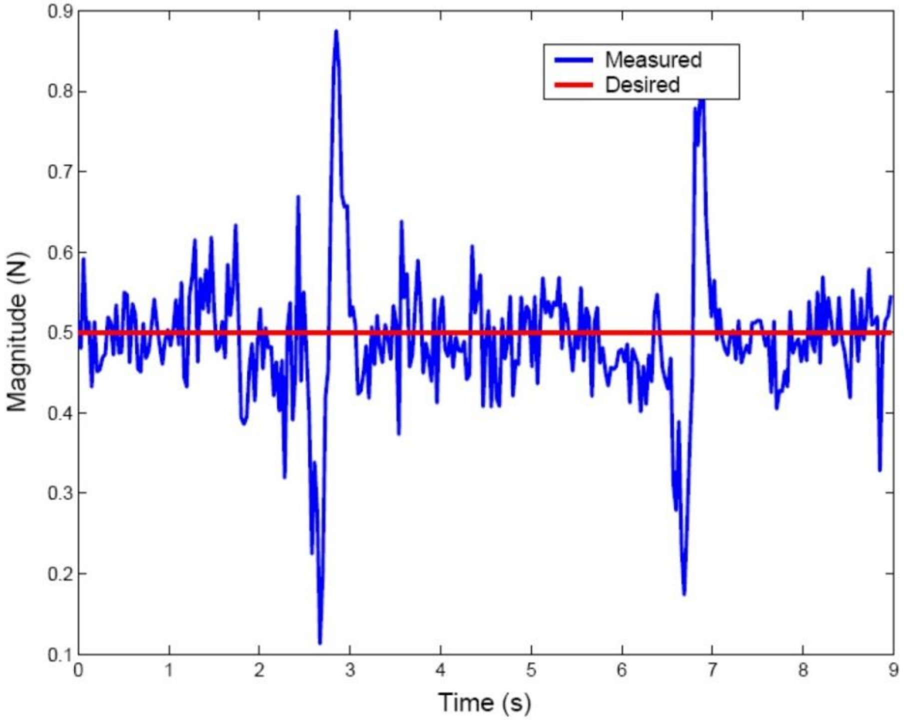
In vivo validation



Ability to maintain contact despite breathing

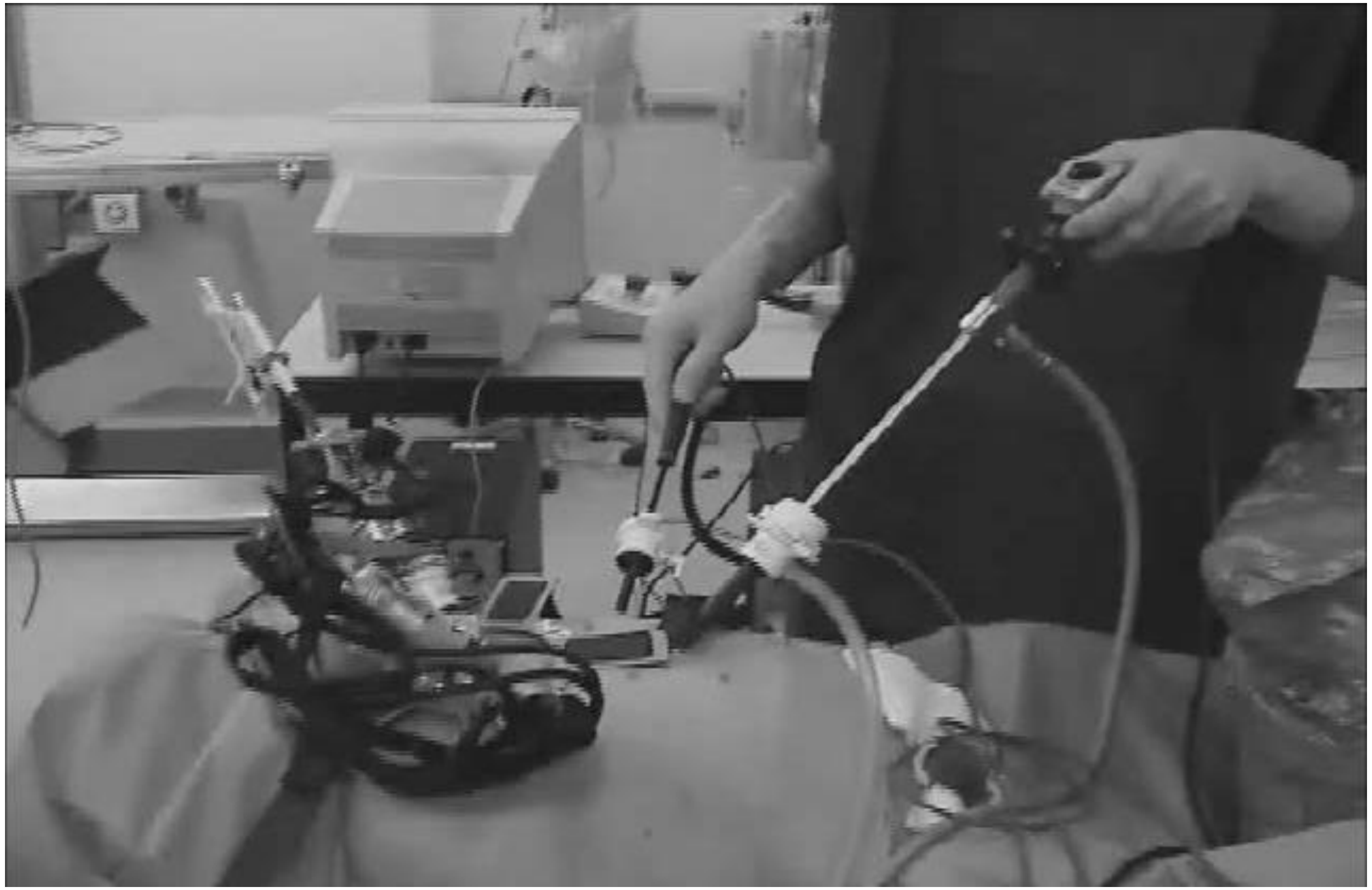
- Natural disturbance rejection properties



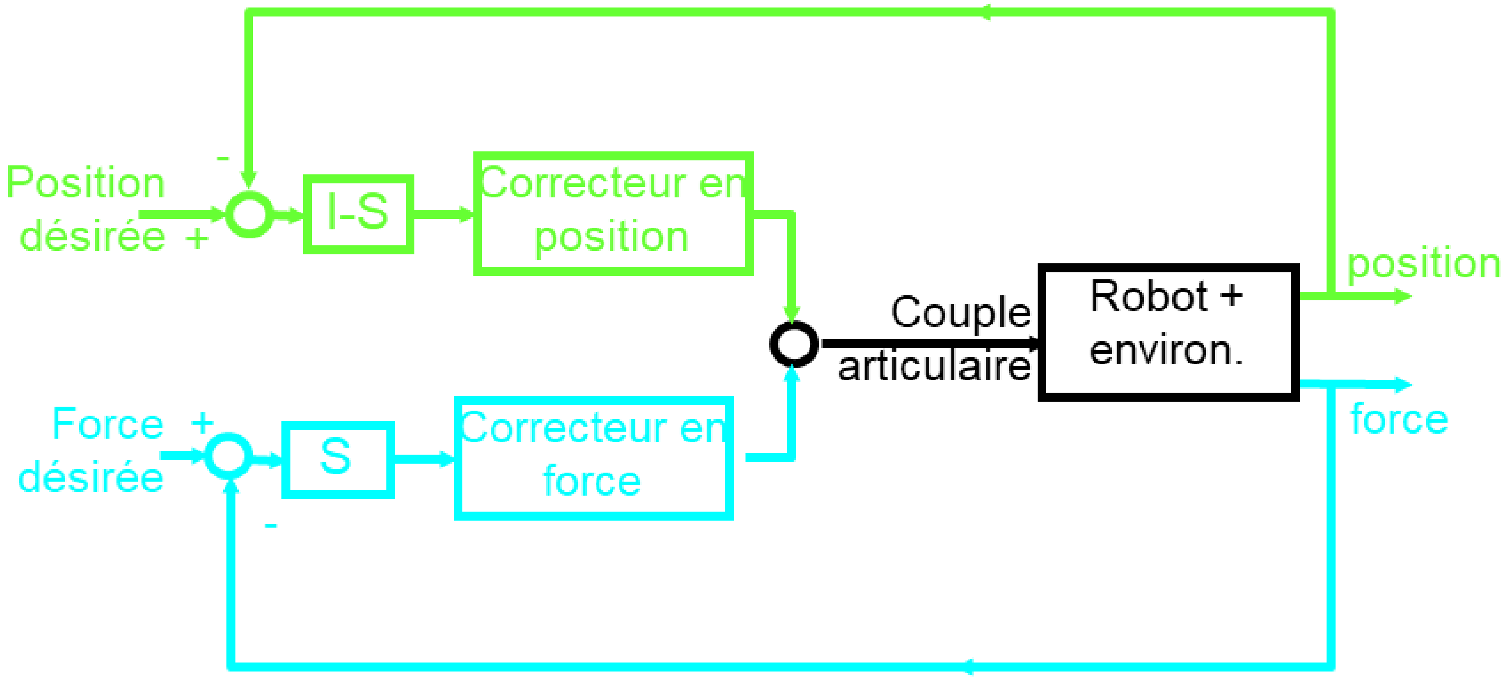


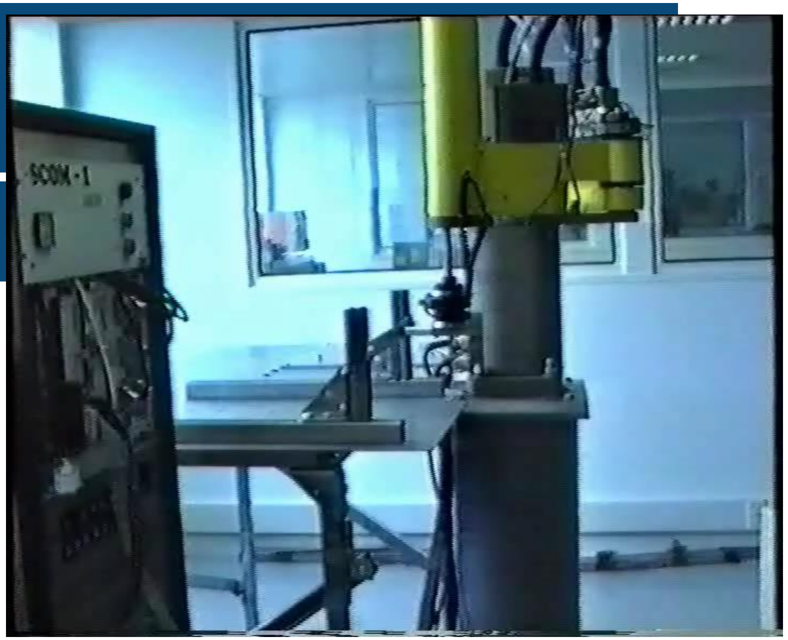
Assistance to surgery

Two successful cholecystectomies realized with pigs by Dr. N. Bonnet at the Surgery School of Paris (APHP).



Hybrid position force control





General conclusions

- We've discussed about modelling and control of robots in terms of :
  - Positioning
  - Moving at a given velocity
  - Applying forces to the environment.
- Major features brought by robot control:
  - Precision : it is rather simple to have a null error in robot control. Simply use an integrator.
  - Robustness to uncertainties in a large extent
  - All this at a price : low bandwidth.