Control of industrial robots

Review of the independent joint control method

Prof. Paolo Rocco (paolo.rocco@polimi.it)

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria

Introduction

- Once the desired motion of the robot has been computed, a real time controller ensures that this motion is tracked in the best possible way

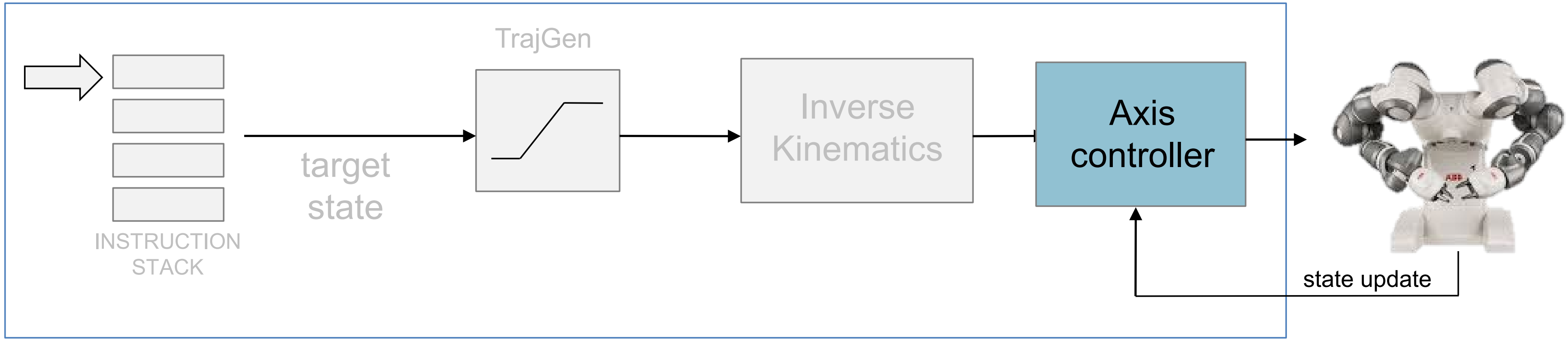
- All industrial robots are in fact endowed with such a real time motion controller

- The industrial robot control follows the **independent joint control** approach, i.e. it is a purely decentralized, non model based, control system

- With these slides, we will review the basics of the approach.

Motion control interfaces

The motion controller is one of the functional units of a robot controller:



The axis controller is the lowest level part of the robot control system, directly interacting with the robot internal sensors and actuators.

Evaluation of control performance

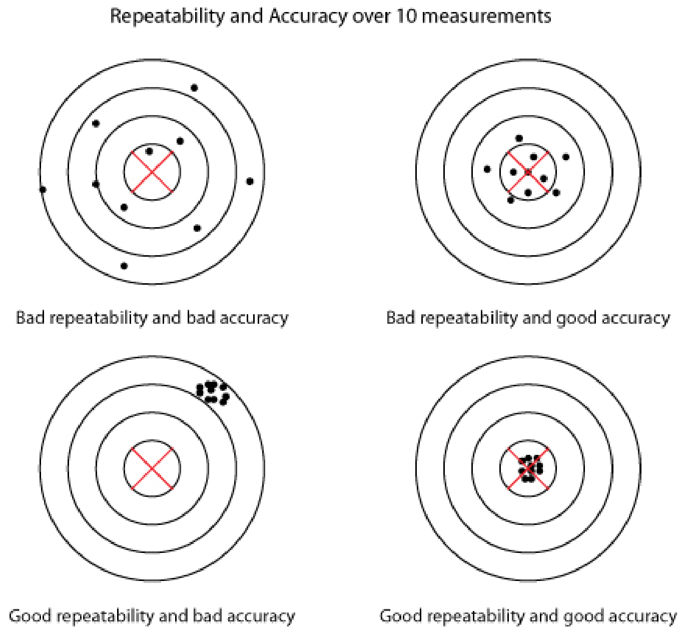
What are possible criteria to evaluate the performance of a motion control system?

- **Quality** of motion in nominal conditions
  - accuracy/repeatability
  - speed of task execution
  - energy saving

- **Robustness** of motion in perturbed conditions
  - adaptation to the environment
  - high repeatability in spite of uncertainties in modelling errors

Accuracy and repeatability

- **Accuracy**: statistical precision with which the robot end-effector reaches a desired target position
- **Repeatability**: statistical precision with which the robot end-effector comes back to the same commanded position, repeating several times the same positioning in the same conditions

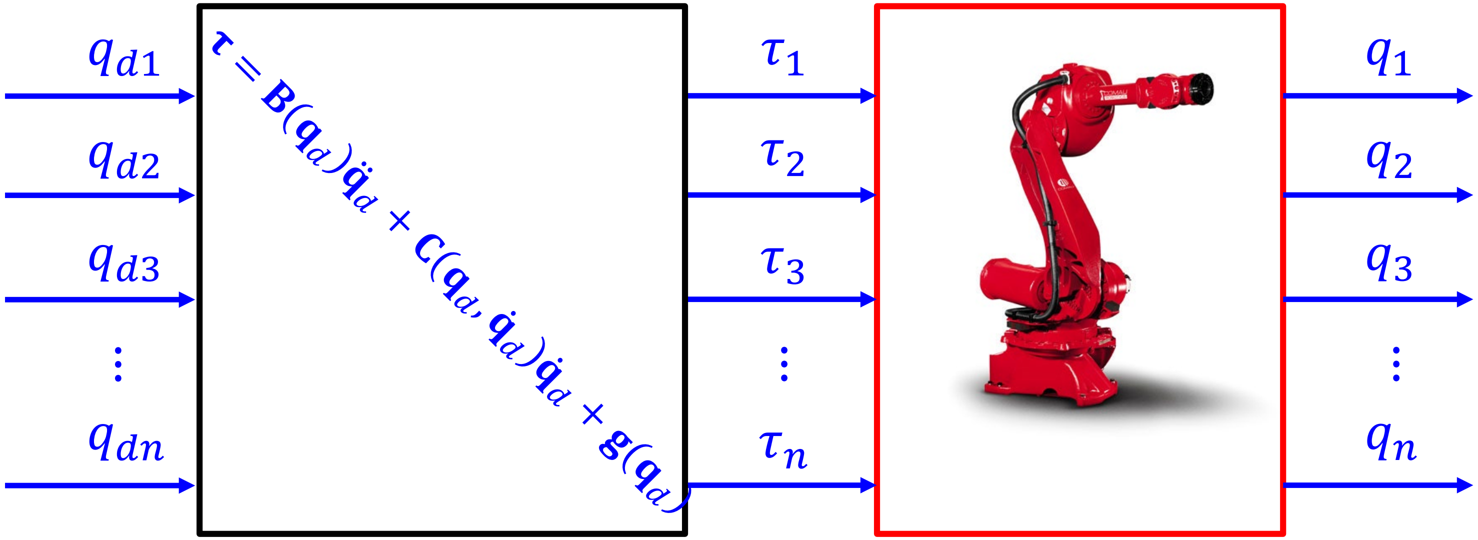


Precise and high-speed motion control

https://www.youtube.com/watch?v=SOESSCXGhFo

An open loop controller

The motion control problem might be in principle approached with an **open loop controller**:

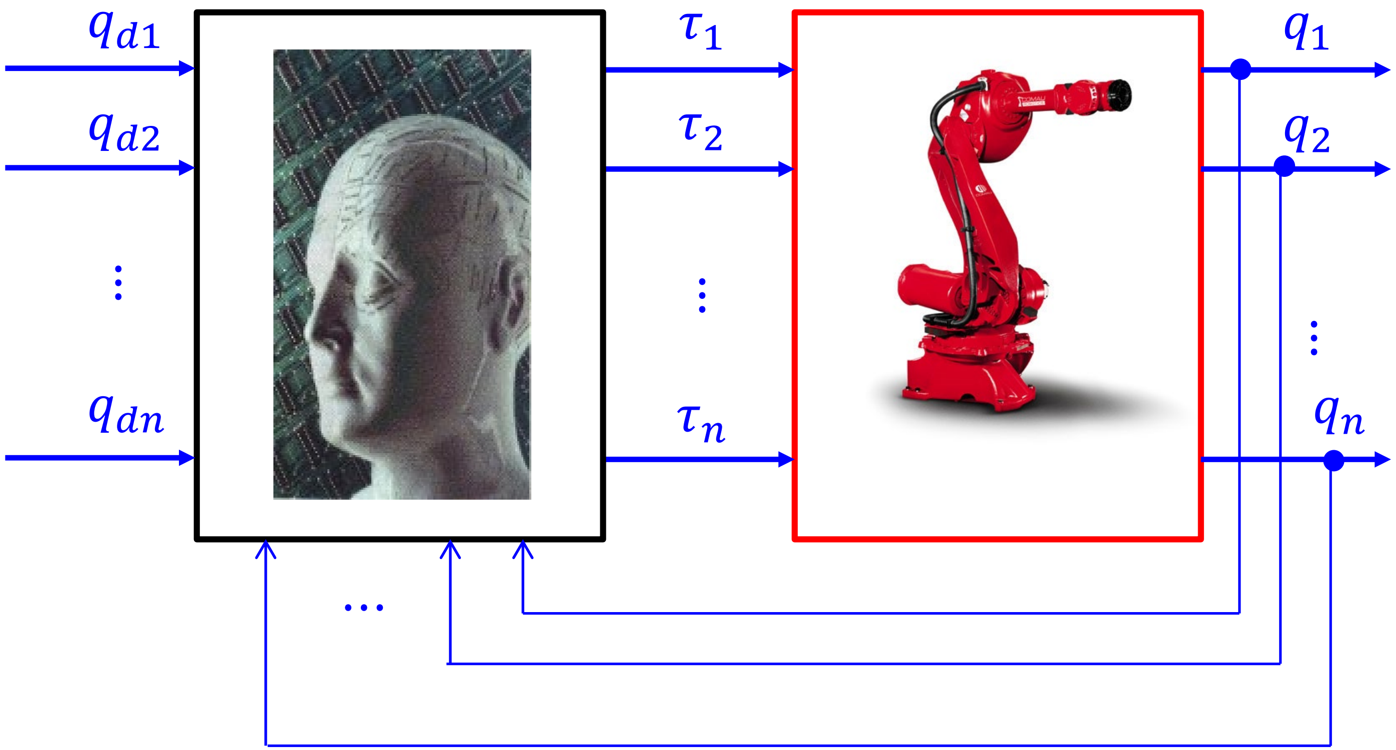


However:

- The model of the robot is not known with enough precision
- The payload of the robot makes the dynamic model change
- Any uncertainty about the model leads to unacceptable errors in the positioning of the robot and also risks

Closed loop control

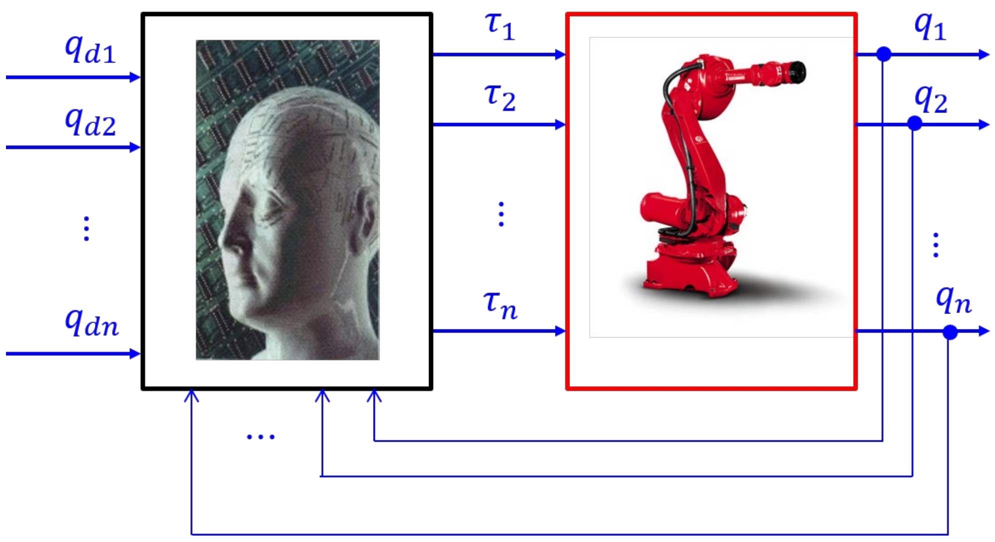
It is therefore necessary to design the control system in closed loop, acquiring the measurements of the positions of the joints, through the position sensors:



Centralized closed loop controller

The motion control problem is a **multivariable one**: we have n controlled variables and n control variables.

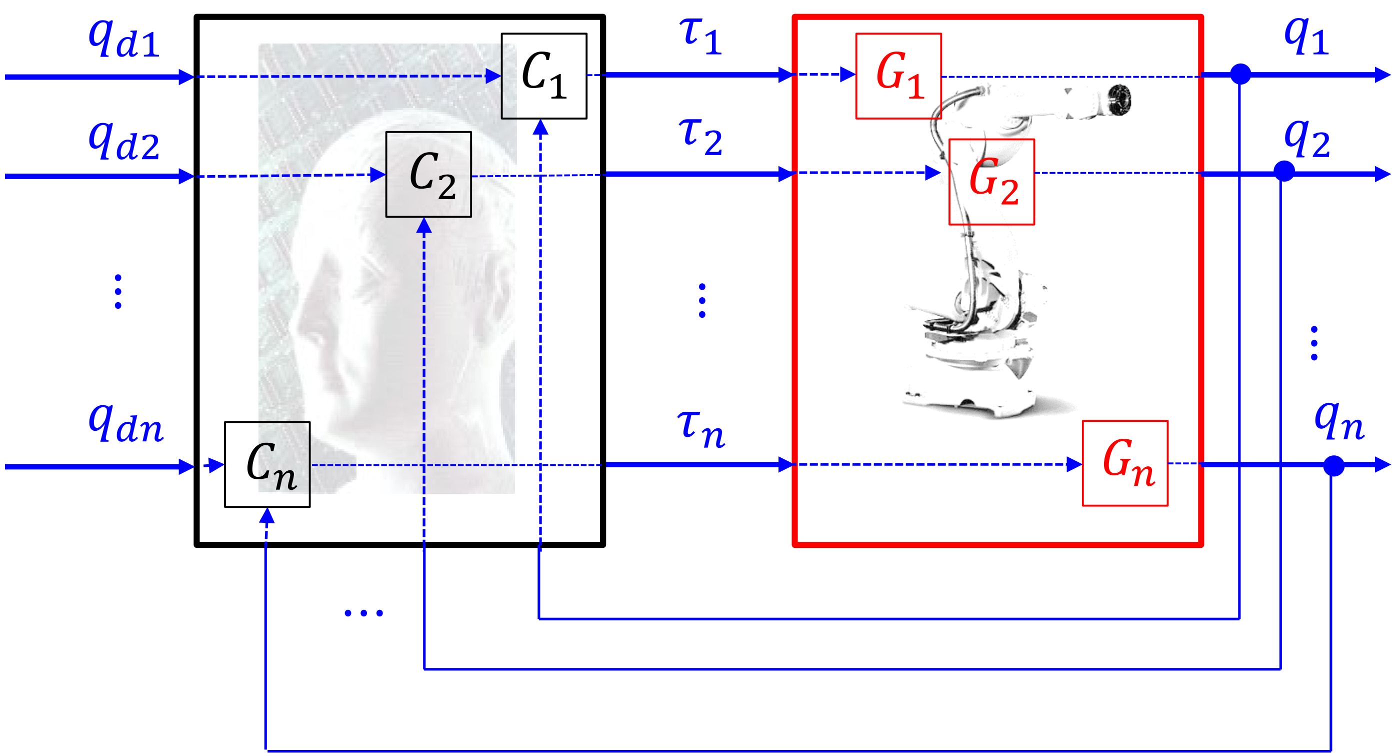
A **centralized control system** might be designed, where each torque τ_i is determined based on the knowledge of the model and on the desired and actual joint positions q_i



We will study these solutions later in the course.

Independent joint control

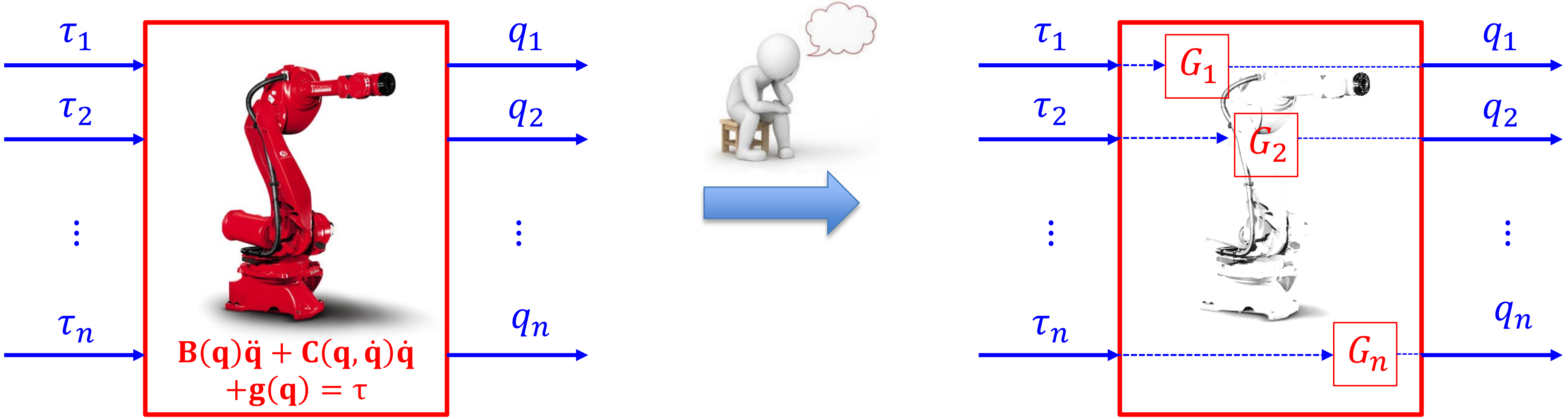
In industry, **decentralized control solutions** (**independent joint control**) are preferred.



- The control system is articulated in n **SISO control loops**
- The controller is structured as n controllers, each devoted to the control of a single joint variable
- The controller makes reference to a completely decoupled model of the system, where each torque τ_i has an influence only on the position q_i

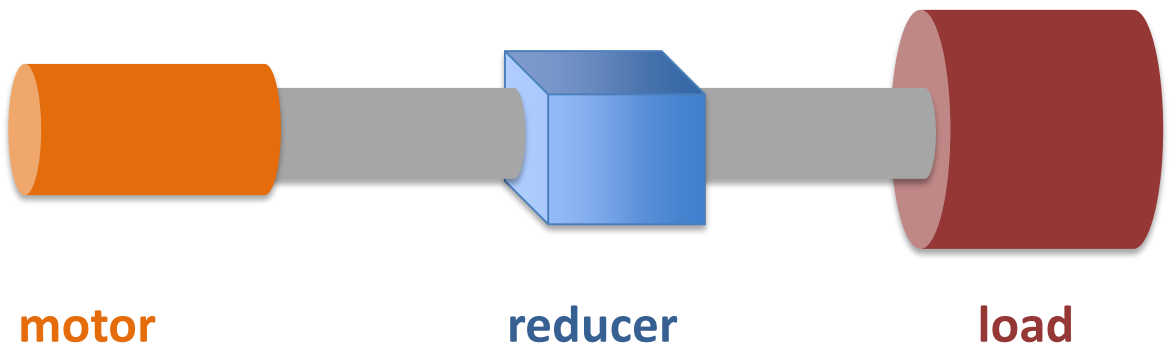
From the centralized model to the decentralized one

How to switch from the complete, non-linear and coupled model of the robot to the decoupled one needed for the independent joint control?



Model of the motors

We know that on each joint of the manipulator a **motor** with its reducer acts.



A simplified way to take into account the dynamics of such motors is to consider only the effect related to the rotation of the motor around its own axis.

$$J_{mi}\ddot{q}_{mi} + D_{mi}\dot{q}_{mi} = \tau_{mi} - \tau_{lmi} \quad i = 1, \cdots, n$$

where J_{mi} and D_{mi} are the moment of inertia and the coefficient of viscous friction of the motor, respectively, while τ_{lmi} is the load torque at the axis of motor i, equal to:

The load torque at the joint

How to represent τ_{lmi}, **load torque** at the axis of motor i ?

It is:

$$\tau_{lmi} = \frac{\tau_i}{n_i}$$

where n_i is the reduction ratio, while τ_i is given by the i_{th} equation of the dynamic model:

$$\tau_i = \mathbf{B}_i^T(\mathbf{q})\ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{C}_i(\mathbf{q})\dot{\mathbf{q}} + g_i(\mathbf{q})$$

We can now rewrite τ_i as:

$$\tau_i = J_{li}\ddot{q}_i + \tau_{di}$$

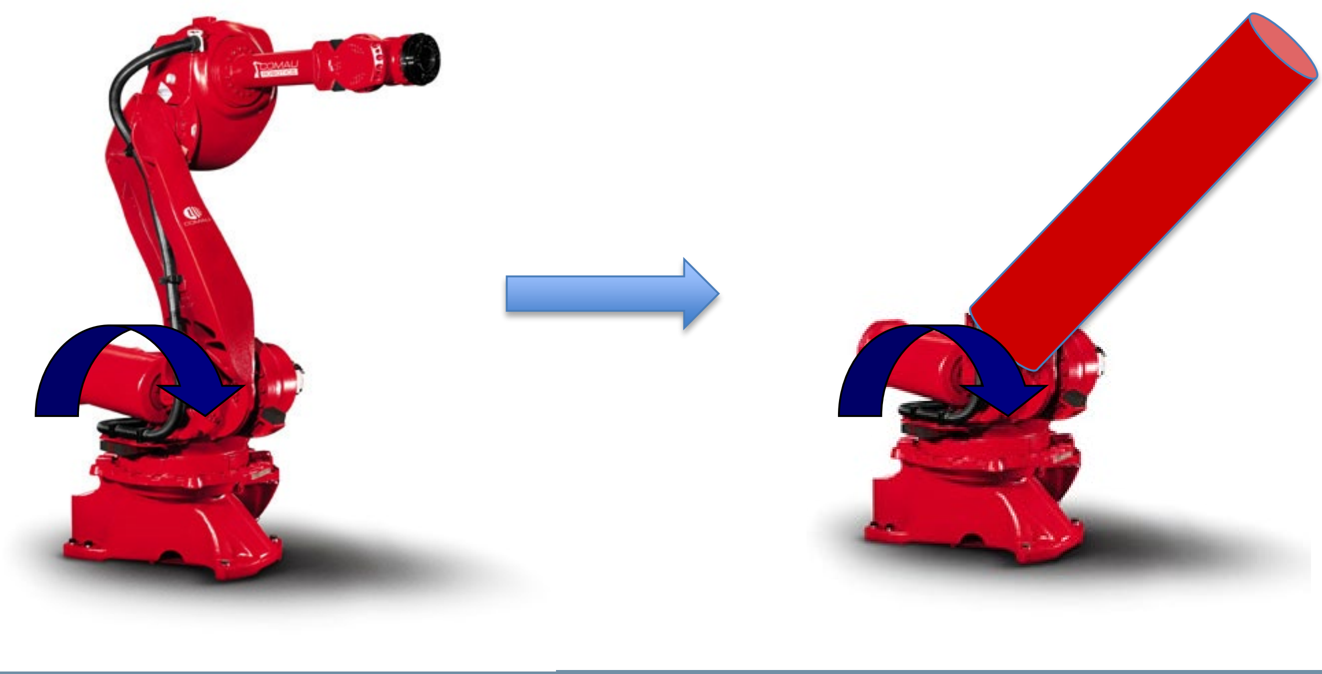
where J_{li} is the **moment of inertia of all the part of the robot moved by joint i** while τ_{di} is a residual term.

The equivalent moment of inertia

$$\tau_i = J_{li}\ddot{q}_i + \tau_{di}$$

J_{li} is the **moment of inertia of all the part of the robot moved by joint i**

Conceptually it is as if we replaced the structure downstream of the joint with a rigid body of inertia J_{li}



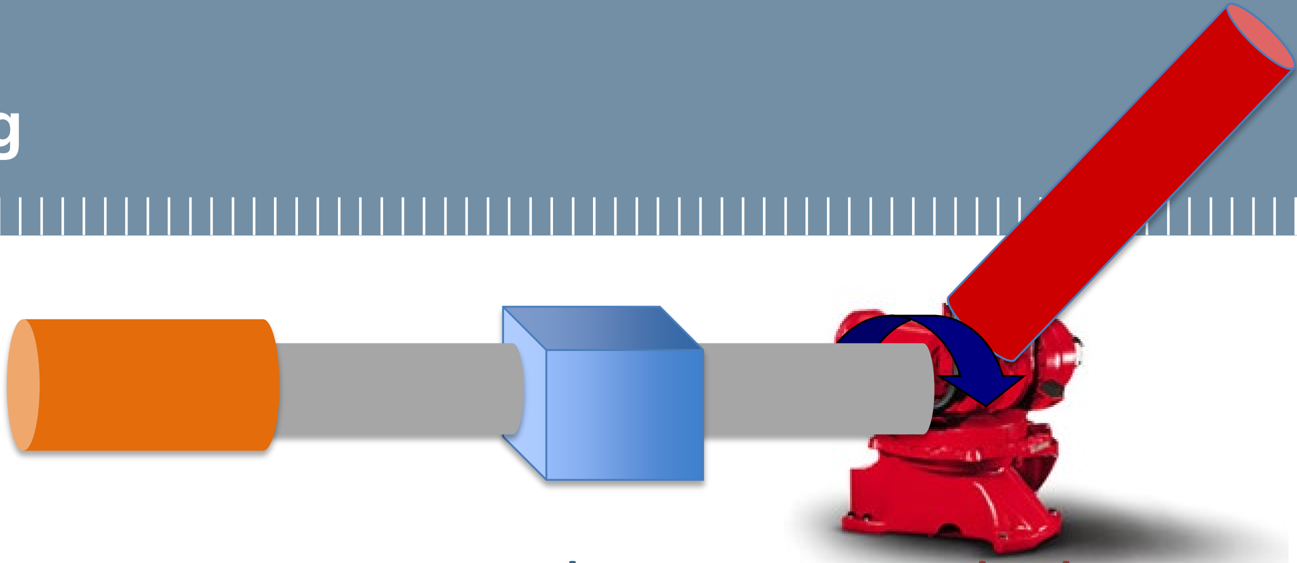
In the image, the operation is performed on joint 2. Note that the moment of inertia varies with the position of the joints from the third onwards.

J_{li} can then be taken as the **average value** of the corresponding **diagonal element in the inertia matrix**:

$$J_{li} = \overline{B_{ii}}$$

or just the value taken by B_{ii} in an arbitrary configuration of the robot.

The rigid coupling



We can then rewrite:

$$J_{mi}\ddot{q}_{mi} + D_{mi}\dot{q}_{mi} = \tau_{mi} - \frac{J_{li}\ddot{q}_i + \tau_{di}}{n_i} \qquad i = 1, \cdots, n$$

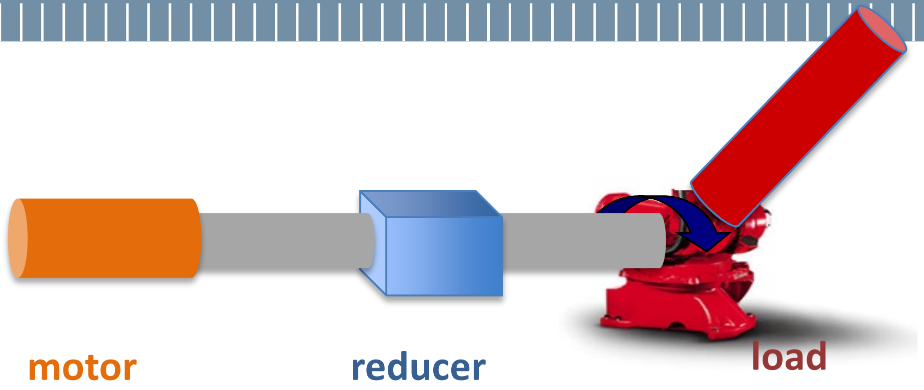
If the motor-reducer-load coupling is rigid, it is $q_{mi} = n_i q_i$ and then:

$$\left(J_{mi} + \frac{J_{li}}{n_i^2}\right)\ddot{q}_{mi} + D_{mi}\dot{q}_{mi} = \tau_{mi} - \frac{\tau_{di}}{n_i} \qquad i = 1, \cdots, n$$

This is the **decentralized model of the robot**

Role of the reduction ratios

$$\left(J_{mi} + \frac{J_{li}}{n_i^2}\right)\ddot{q}_{mi} + D_{mi}\dot{q}_{mi} = \tau_{mi} - \frac{\tau_{di}}{n_i} \qquad i = 1, \cdots, n$$



Torque τ_{di} acts as a disturbance on the decoupled model.

In fact, it accounts for all the nonlinear and coupling effects of the dynamic model

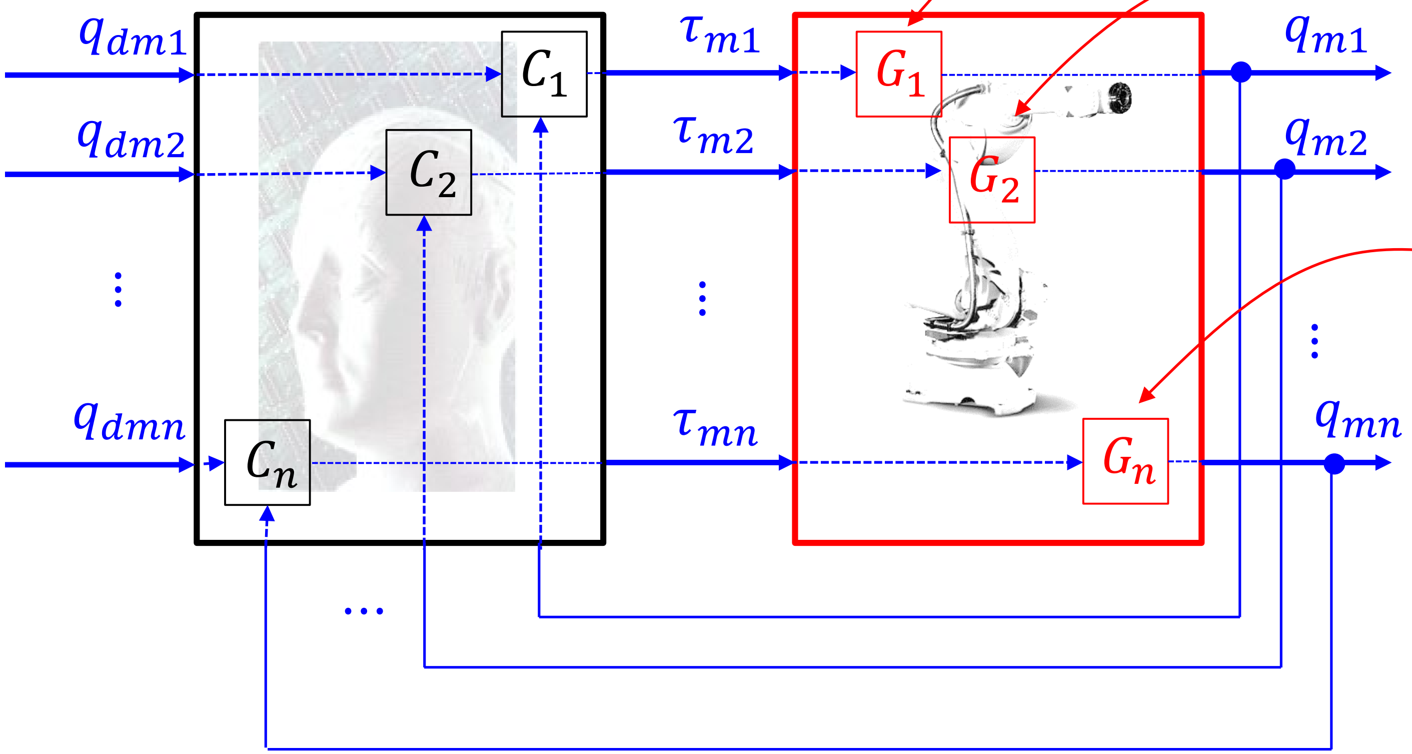
Note that in the equation this torque is **scaled** by the reduction ratio n_i

The high **reduction ratios** used in industrial robotics therefore have a **decoupling effect** and favor the adoption of independent joint control.

Without this effect, neglecting the variability of the inertia of the load and the effects of mechanical coupling with the other joints would not be justified.

Design of the decentralized controller

The controller is then designed joint by joint repeating n times the same procedure.



$$\left(J_{m1}+\frac{J_{l1}}{n_1^2}\right)\ddot{q}_{m1}+D_{m1}\dot{q}_{m1}=\tau_{m1}-\frac{\tau_{d1}}{n_1}$$

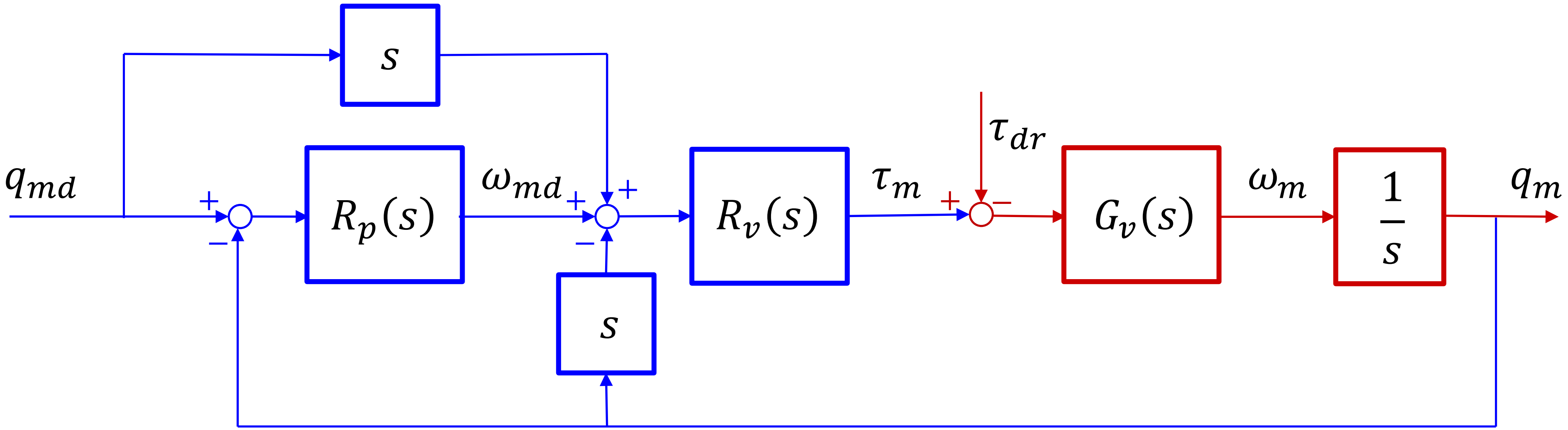
$$\left(J_{m2}+\frac{J_{l2}}{n_2^2}\right)\ddot{q}_{m2}+D_{m2}\dot{q}_{m2}=\tau_{m2}-\frac{\tau_{d2}}{n_2}$$

$$\vdots$$

$$\left(J_{mn}+\frac{J_{ln}}{n_n^2}\right)\ddot{q}_{mn}+D_{mn}\dot{q}_{mn}=\tau_{mn}-\frac{\tau_{dn}}{n_n}$$

Position/speed control

For each joint, the controller is structured in an inner speed control loop and an outer position control loop:

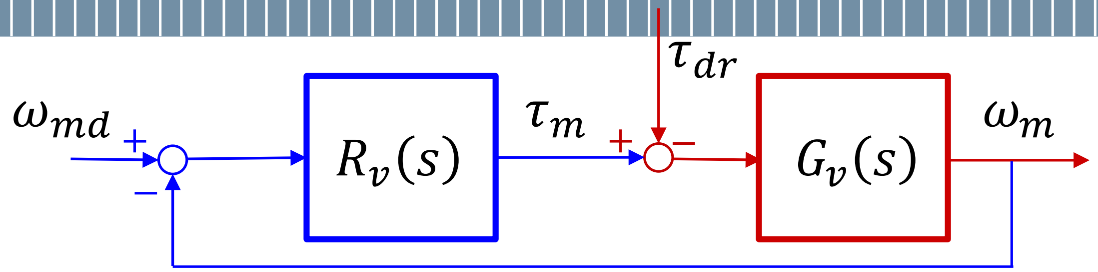


This is a **cascaded loop** arrangement

- Only a position sensor is used, while an estimate of the speed is obtained through **numerical differentiation** of the position
- A **speed feed-forward** can be added to make the control system more reactive to setpoint variations
- The innermost control loop in fact is a **motor current loop** (not shown in the block diagram) that allows to assign the motor torque τ_m assigning a corresponding current setpoint
- Neglecting the viscous friction term, the **transfer function** from motor torque τ_m to motor speed ω_m is:
  $G_v(s) = \frac{\mu}{s}$, with $\mu = \frac{1}{J_m + J_l/n^2}$

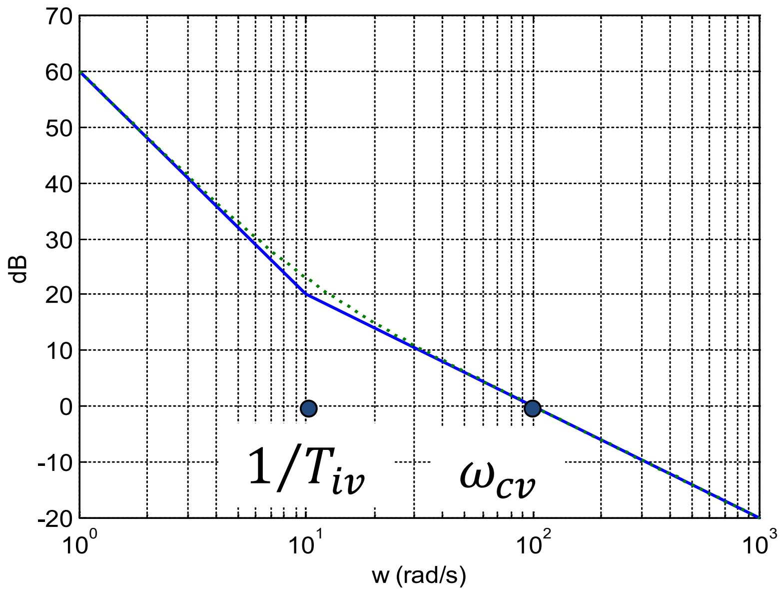
Design of the PI speed controller

A **PI controller** is adopted



$$R_{PI}(s) = K_{pv}\left(1 + \frac{1}{sT_{iv}}\right) = K_{pv}\frac{1 + sT_{iv}}{sT_{iv}}$$

Loop transfer function:
$$L_v(s) = R_{PI}(s)G_v(s) = \frac{K_{pv}\mu}{s}\frac{1 + sT_{iv}}{sT_{iv}}$$



If T_{iv} is sufficiently large, i.e. if the zero of the PI is in a sufficiently low frequency range, the crossover frequency ω_{cv} is well approximated by the high frequency approximation of L_v:

$$L_v(s) \approx \frac{\omega_{cv}}{s}$$

$$\boxed{\omega_{cv} = K_{pv}\mu}$$

selection of the **PI gain**

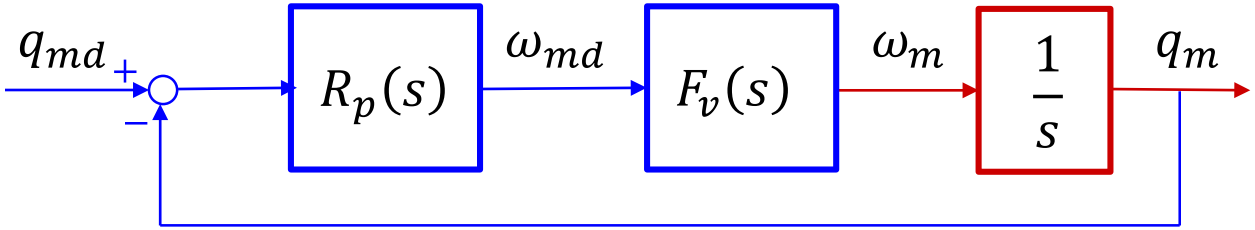
$$\boxed{\frac{1}{T_{iv}} = (0.1 \div 0.3)\omega_{cv}}$$

placement of the **zero of the PI**

Design of the P position controller

A **P controller** is adopted

$$R_p(s) = K_{pp}$$



The position controller "sees" the speed closed loop, whose transfer function can be approximated as:

$$F_v(s) \approx \frac{1}{1 + s/\omega_{cv}}$$

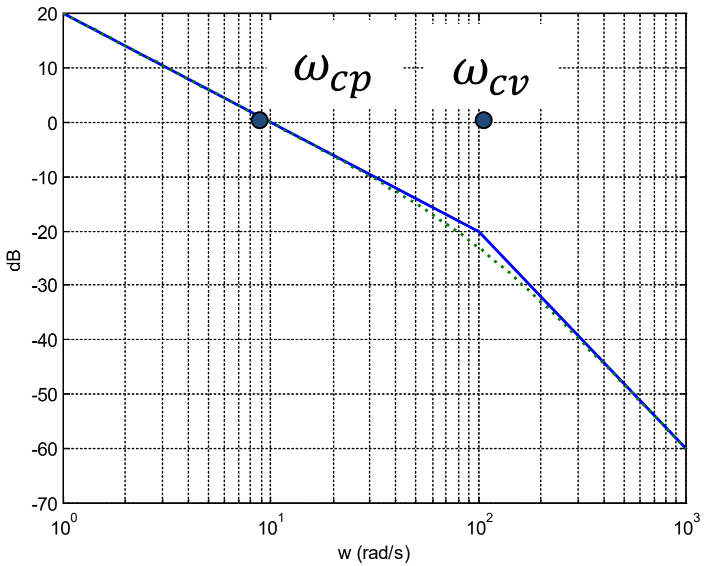
The loop transfer function is thus:

$$L_p(s) = K_{pp} F_v(s) \frac{1}{s} = \frac{K_{pp}}{s(1 + s/\omega_{cv})}$$

It is enough to take $K_{pp} \ll \omega_{cv}$ in order to guarantee a crossover frequency ω_{cp} :

$$\boxed{\omega_{cp} = K_{pp}}$$

Selection of the **P gain**



Limitations of the rigid model

- The rigid model does not imply any significant limitations to the bandwidth of the speed controller: in principle we might obtain an arbitrarily fast closed loop system

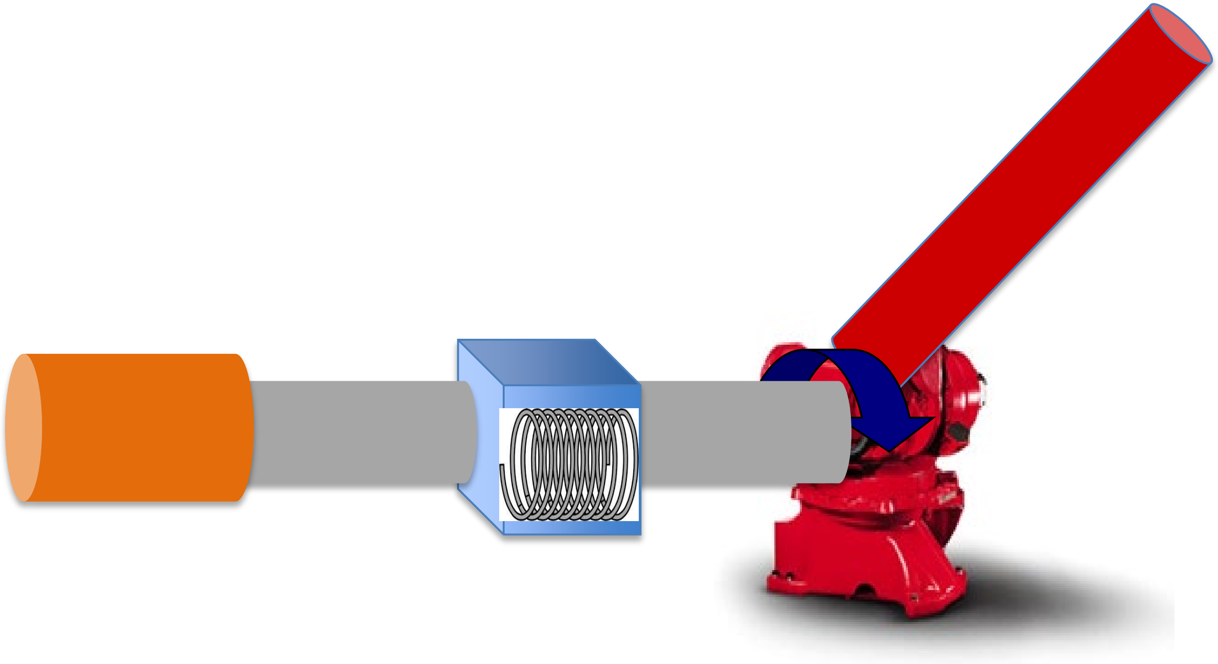
- In practice, limitations clearly emerge, in terms of vibrations, noise, oscillations, etc.

- Obviously the rigid model is not adequate to explain how a servomechanism behaves, in case the required performance are increased

- We need to **complicate the model**

Elastic coupling between motor and load

The simplest way to account for non-rigid behaviour of the system is to include an **elastic coupling** between motor and load, which are still considered rigid systems.



Limitations in the bandwidth

If we revise the design of the speed controller accounting for the flexibility in the joint, we come up with limitations in the bandwidth of the speed controller.

Specifically, if we set:

$$\omega_z = \sqrt{\frac{K_{el}}{J_{lr}}}$$

we have as a **reasonable rule** to tune the controller:

$$\boxed{\omega_{cv} \approx 0.7\omega_z}$$

The bandwidth of the position controller has to be reduced accordingly.