LECTURE 4

Hand guiding

Paolo Rocco

Hand guiding

This mode of collaboration allows the operator to manually drive the robot around. Effort is decreased as power is on and motors provide assistive torques.

When provided, hand guiding equipment shall be located close to the end-effector and shall be equipped with the following:

- an emergency stop (red button)
- an enabling device (dead man button)

It is mostly used during programming of machining operations, welding, etc.

Hand guiding

VS

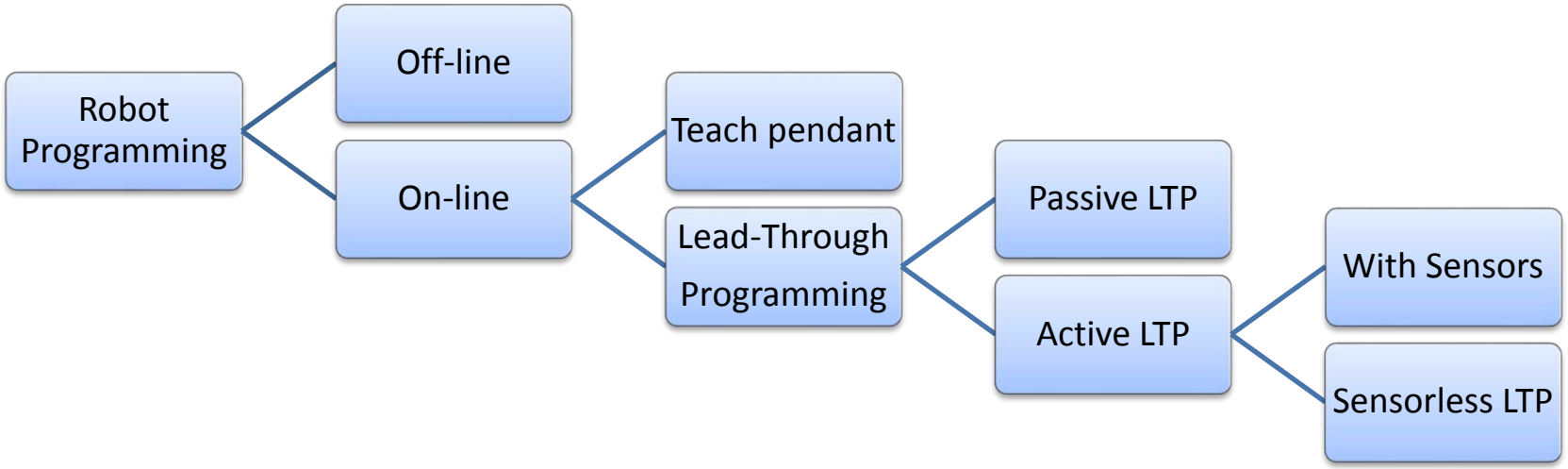
Hand guided robot:

- Potentially unsafe
- Requires expensive hardware (motors, force/torque sensor)
- Provides assistive torques (reduced inertia)
- Mid/low payload

Handling device:

- Inherently safe (passive device)
- Less expensive
- No assistive torque (no inertia reduction)
- High payload

A taxonomy of robot programming



Lead-through programming

How can we obtain this?

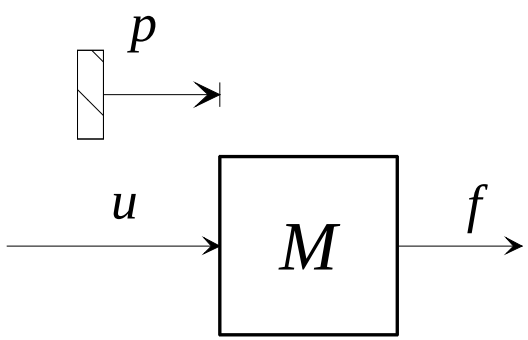
Mechanical impedance

When studying the interaction of the robot with the environment, it is helpful to introduce the concept of mechanical impedance:

- The flow of power among two physical systems can always be defined as the product of two conjugate quantities: a (generalized) effort and a (generalized) flow
- In the electrical framework, the flow is the current, while the effort is the voltage. In the mechanical framework, instead, the flow is the velocity, the effort is the force (or the moment)
- The mechanical impedance is then defined as dynamical relation that is established between force and velocity (or displacement) for a mechanical system. The admittance is just the reciprocal of the impedance
- The impedance control aims at making the manipulator in interaction with the environment assume a desired mechanical impedance, like a generalized mass-spring-damper system

A 1 d.o.f. system

In order to understand how an impedance controller operates, let us first consider the simple example of a body with mass *M* interacting with the environment, which exerts a force *f*:



The model of the system is simply:

$$Ma = u + f$$

Notice that the model of the interaction with the environment is not specified: we just know that there is an external force acting on the system

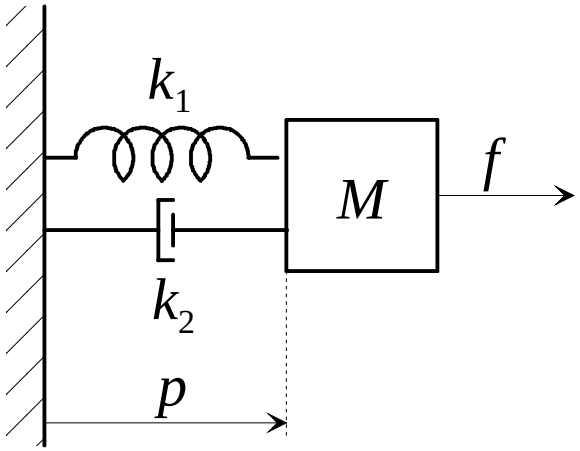
A 1 d.o.f. system

If we assume a control proportional to position and velocity:

$$u = -k_1 p - k_2 v$$

We obtain:

$$Ma + k_2 v + k_1 p = f$$



The controlled system reacts to the external force like a mass-spring-damper

The spring and damper factors are tunable, the mass is the physical one of the system.

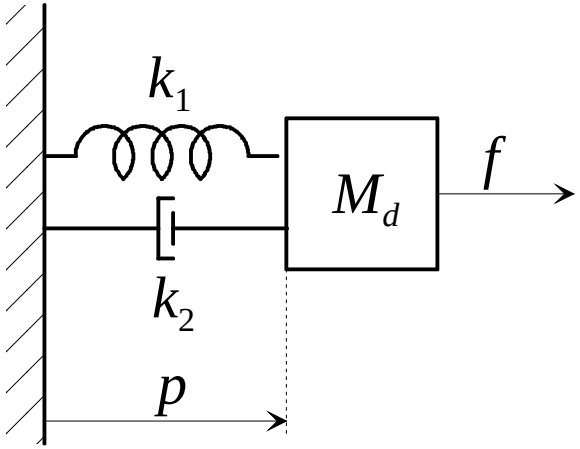
A 1 d.o.f. system

If we want to change the mass, too, we need to measure the force:

$$u = \frac{M}{M_d}\left(-k_1 p - k_2 v + f\right) - f$$

We obtain:

$$M_d a + k_2 v + k_1 p = f$$

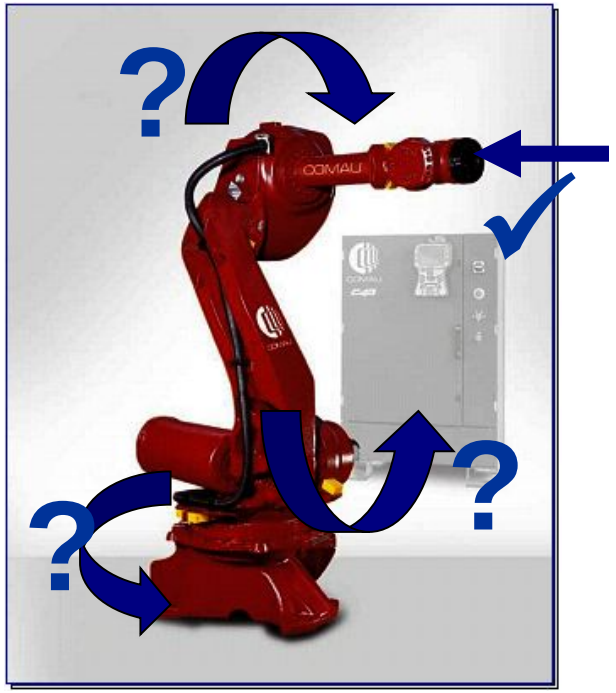


All the parameters of the mass, spring and damper system are now tunable.

Extension to the whole manipulator

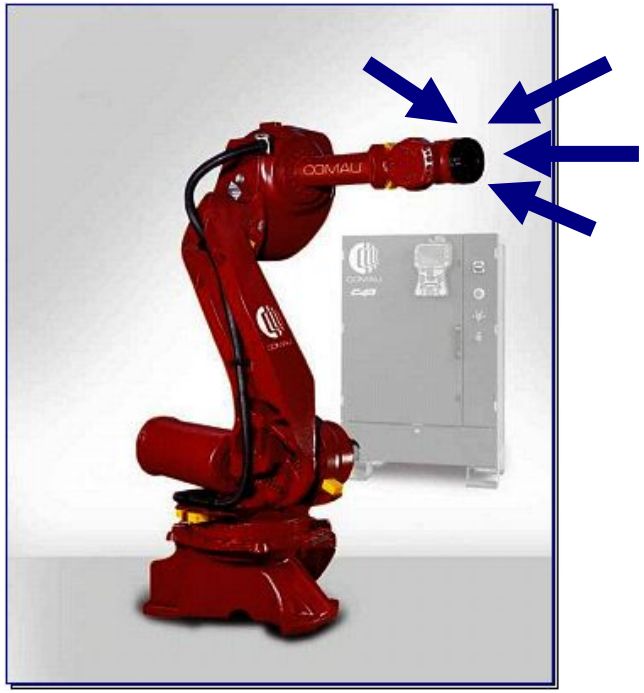
We want to extend the concept of impedance control to the whole manipulator.
We need first to address a basic problem.

How to include the effect of a force applied at the end-effector in the model of the robot?



Suppose then that a force is applied at the end-effector.
What are the joint torques that keep the system in equilibrium?

Systems of forces



As a matter of fact a system of forces may act at the end-effector.

How can this system be reduced?
From classical mechanics…

- One force applied in a point (the resultant)
- One torque

It follows that we will always be able to represent a system of forces with two vectors:

- A force vector applied in a point *O* (the resultant)
- A moment vector (the moment of the original system of forces with respect to the fulcrum *O*)

Statics of the manipulator

Consider then a manipulator subject to a set of forces at the end effector, having a resultant $\boldsymbol{f}$ and moment, w.r.t. the origin of the end effector frame, μ. We then define vector:

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f} \\ \mu \end{bmatrix}$$

Let us call τ the vector of torques (or forces) applied at the joints.

The virtual work of the forces/torques at the joints is:

$$\delta W_\tau = \tau^T \delta \boldsymbol{q}$$

where $\delta \boldsymbol{q}$ is the elementary displacement of the joint coordinates.

The virtual work of the forces at the end-effector is:

$$\delta W_h = \boldsymbol{f}^T \delta \boldsymbol{p} + \mu^T \omega dt = \boldsymbol{f}^T \boldsymbol{J}_P(\boldsymbol{q}) \delta \boldsymbol{q} + \mu^T \boldsymbol{J}_O(\boldsymbol{q}) \delta \boldsymbol{q} = \boldsymbol{h}^T \boldsymbol{J}(\boldsymbol{q}) \delta \boldsymbol{q}$$

where $\delta \boldsymbol{p}$ is the virtual displacement of the origin of the end-effector frame, consistent with the differential kinematics of the robot, and ω is the angular velocity of the end effector frame.

Statics of the manipulator

$$\delta W_\tau = \boldsymbol{\tau}^T \delta \boldsymbol{q}$$

$$\delta W_h = \boldsymbol{f}^T \boldsymbol{J}_P(\boldsymbol{q})\delta \boldsymbol{q} + \boldsymbol{\mu}^T \boldsymbol{J}_O(\boldsymbol{q})\delta \boldsymbol{q} = \boldsymbol{h}^T \boldsymbol{J}(\boldsymbol{q})\delta \boldsymbol{q}, \qquad \boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_O(\boldsymbol{q}) \end{bmatrix}$$

position Jacobian

orientation Jacobian

Based on the principle of virtual work the manipulator is in static equilibrium if and only if:

$$\delta W_\tau = \delta W_h \quad \Rightarrow \quad \boxed{\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}}$$

Notice that the same matrix, the Jacobian matrix, which defines the differential kinematics of the manipulator, is also involved in the statics of the system (kineto-static duality).

Impedance control for the whole manipulator

We are now ready to extend the impedance control to the whole manipulator.

The dynamic model in interaction with the environment is given by the equation:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}$$

where $\boldsymbol{h}$ is the vector of the contact forces (and moments) applied by the end effector on the environment.

Consider now the adoption of an inverse dynamics control law:

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

Substituting we obtain:

nonlinear coupling term

$$\ddot{\boldsymbol{q}} = \boldsymbol{y} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}$$

Impedance control for the whole manipulator

Assume now for $\boldsymbol{y}$ an inverse dynamics control in the operational space:

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)$$

By substituting the expression for $\boldsymbol{y}$ we obtain the equation of the dynamics in closed loop:

$$\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\tilde{\boldsymbol{x}} = \boldsymbol{B}_A^{-1}(\boldsymbol{q})\boldsymbol{h}_A$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}$, while:

$$\boldsymbol{B}_A(\boldsymbol{q}) = \boldsymbol{J}_A^{-T}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}_A^{-1}(\boldsymbol{q})$$

is the inertia matrix in the operational space and:

$$\boldsymbol{h}_A = \boldsymbol{T}_A^T(\boldsymbol{x})\boldsymbol{h}, \qquad \boldsymbol{J} = \boldsymbol{T}_A(\boldsymbol{x})\boldsymbol{J}_A \quad \Rightarrow \quad \boldsymbol{J}^T\boldsymbol{h} = \boldsymbol{J}_A^T\boldsymbol{h}_A$$

generalized forces performing work on $\dot{\boldsymbol{x}}$

Impedance control for the whole manipulator

$$\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D \tilde{\boldsymbol{x}} = \boldsymbol{B}_A^{-1}(\boldsymbol{q})\boldsymbol{h}_A$$

We have obtained an impedance relation which is coupled and only partially assignable: a force applied along a Cartesian direction yields motion in other directions as well.

If force measurements are available, we can set:

$$\tau = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}$$

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\boldsymbol{M}_d^{-1}\left(\boldsymbol{M}_d \ddot{\boldsymbol{x}}_d + \boldsymbol{D}_d \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d \tilde{\boldsymbol{x}} - \boldsymbol{M}_d \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{h}_A\right)$$

where $\boldsymbol{M}_d$, $\boldsymbol{D}_d$, $\boldsymbol{K}_d$ are diagonal, positive definite matrices.

We obtain:

$$\boldsymbol{M}_d \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d \tilde{\boldsymbol{x}} = \boldsymbol{h}_A$$

i.e. a completely decoupled system.

Impedance control for the whole manipulator

The equation:

$$\boldsymbol{M}_d \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d \tilde{\boldsymbol{x}} = \boldsymbol{h}_A$$

defines a mechanical impedance relation between the forces/moments defined at the right hand side of the equations and the position/orientation errors in the operational space.
This impedance has the form of a mechanical system with mass matrix $\boldsymbol{M}_d$, damping $\boldsymbol{D}_d$, and stiffness $\boldsymbol{K}_d$.

The dynamic behaviour along each direction of the operational space can thus be specified: a force or a moment along a certain Cartesian direction yields a motion in such direction, which is related to the force and moment by a fully controllable dynamics.

In Lead-Through programming the stiffness $\boldsymbol{K}_d$ is usually set to zero.

A note on rotational impedance (1/2)

The last three degrees of freedom in the operational space consist in a minimal representation (Euler angles) of the end-effector orientation.
In this case the definition of the impedance is less intuitive:

$\boldsymbol{\phi}_d$: desired orientation

$\boldsymbol{\phi}_e$: actual orientation

$$\Delta\boldsymbol{\phi} = \boldsymbol{\phi}_d - \boldsymbol{\phi}_e$$

The impedance relation is:

$$\boldsymbol{M}_\phi \Delta\ddot{\boldsymbol{\phi}} + \boldsymbol{D}_\phi \Delta\dot{\boldsymbol{\phi}} + \boldsymbol{K}_\phi \Delta\boldsymbol{\phi} = \boldsymbol{T}^T(\boldsymbol{\phi}_e)\boldsymbol{\mu}$$

Due to the presence of matrix ***T***, this relation is subject to representation singularities and is geometrically inconsistent.

Geometrical inconsistency:

- The relation between the moments applied by the environment at the end effector frame and the angular velocity of the same frame depends on the orientation of the frame
- The eigenvectors of matrix $\boldsymbol{K}_\phi$ do not represent the principal axes of inertia of the rotational stiffness

A note on rotational impedance (2/2)

An alternative definition which does not suffer from representation singularities and geometrical inconsistency problems is the following one:

ϕ_{ed} set of XYZ Euler angles extracted from the rotation matrix $\boldsymbol{R}_e^d$

μ^d Vector of moments applied by the environment on the end-effector, expressed in the desired frame

$$\boldsymbol{M}_\phi \ddot{\phi}_{ed} + \boldsymbol{D}_\phi \dot{\phi}_{ed} + \boldsymbol{K}_\phi \phi_{ed} = \boldsymbol{T}^T(\phi_{ed})\mu^d$$

Matrix $\boldsymbol{T}(\phi_{ed})$, for small values of ϕ_{ed}, is almost an identity matrix, and thus does not yield neither representation singularities nor geometrical inconsistency problems .

Issues with model-based impedance control

In view of the application of model-based impedance control to industrial robotics manipulator, so as to achieve lead-through capabilities, there are several issues with the method:

- The method requires complete knowledge of the dynamic model and perfect cancellation of the nonlinearities and couplings
- The method assumes a completely open control architecture, where access to the joint torque commands is available
- The system become inherently compliant to external disturbances, which is conflicting with the typical stiffness required to the industrial robots

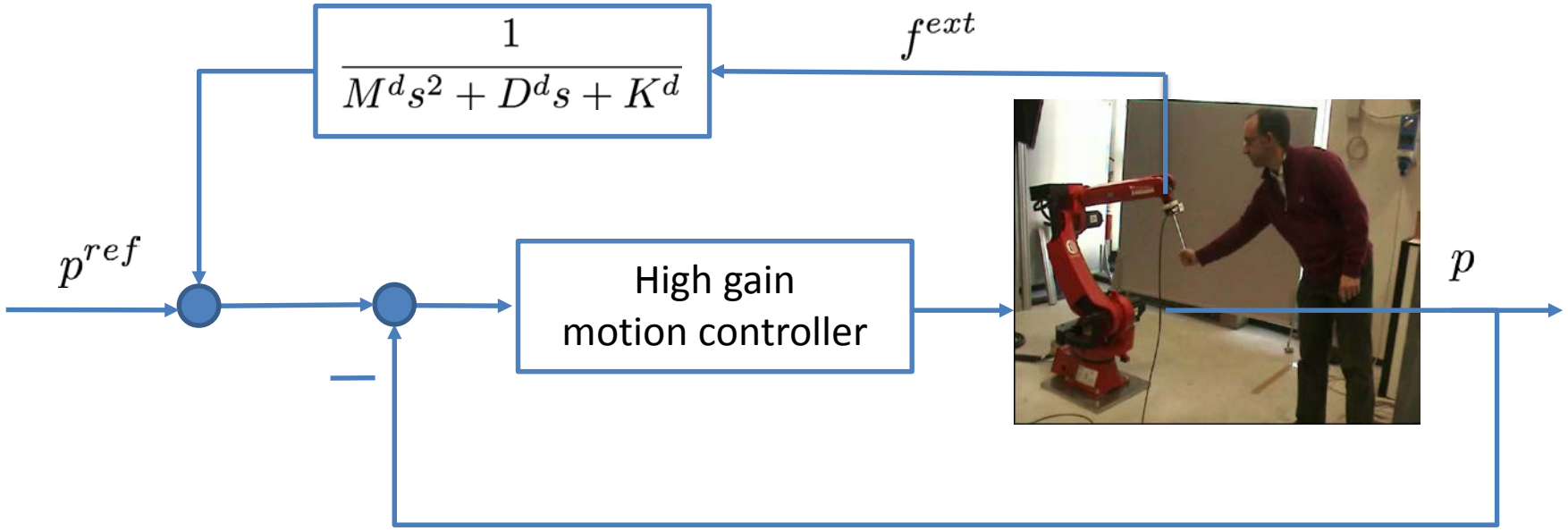
How can we devise a solution that can be implemented with a reasonable effort on an industrial robot?

Admittance control

We can **rely on the stiff position control** and modify the reference position:

Admittance filter

converts a force into desired displacements

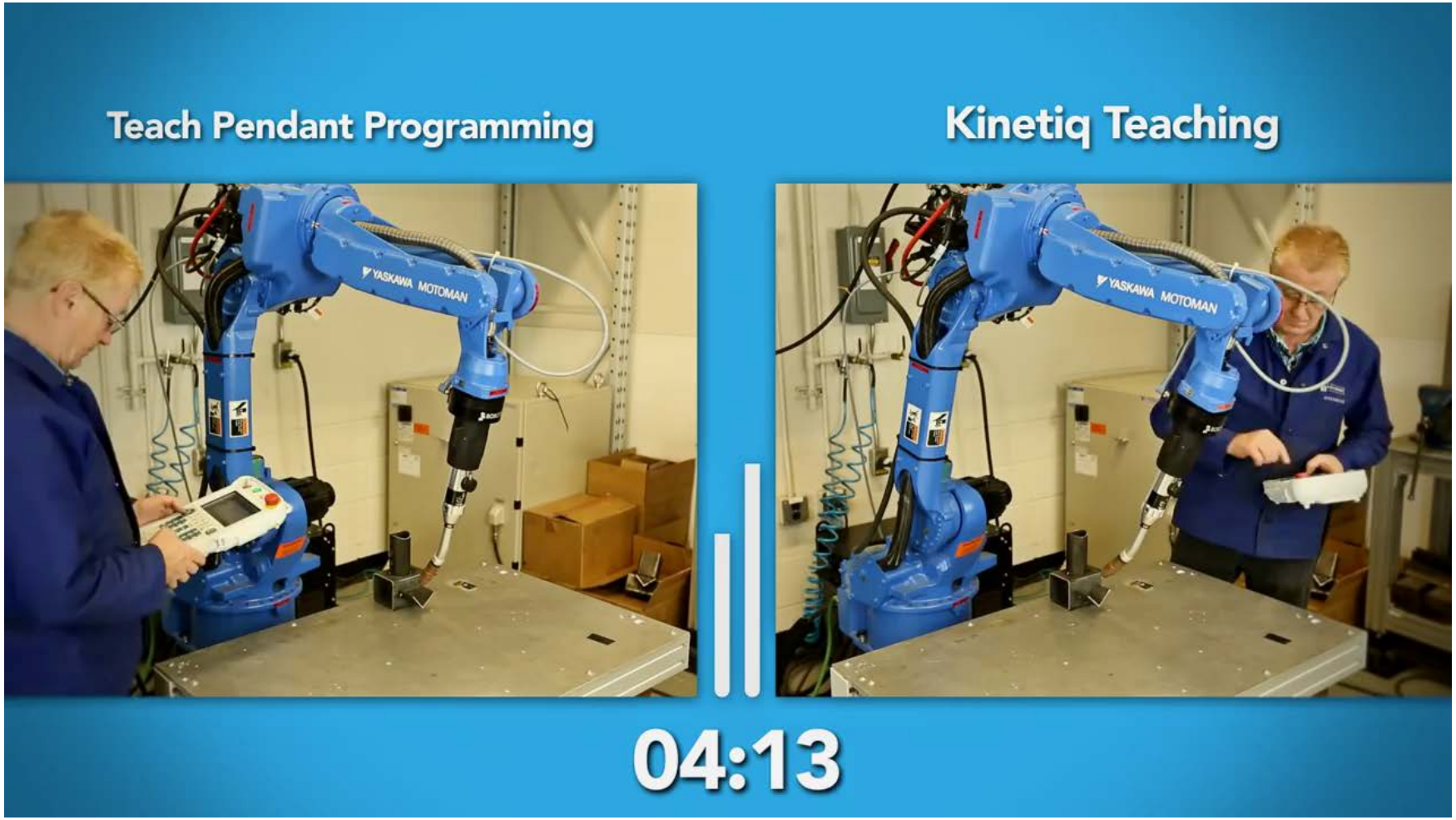


$$p \approx p^{ref} - \frac{1}{M^d s^2 + D^d s + K^d} f^{ext}$$

$$\boxed{p^{ref} - p \approx \frac{1}{M^d s^2 + D^d s + K^d} f^{ext}}$$

desired impedance achieved

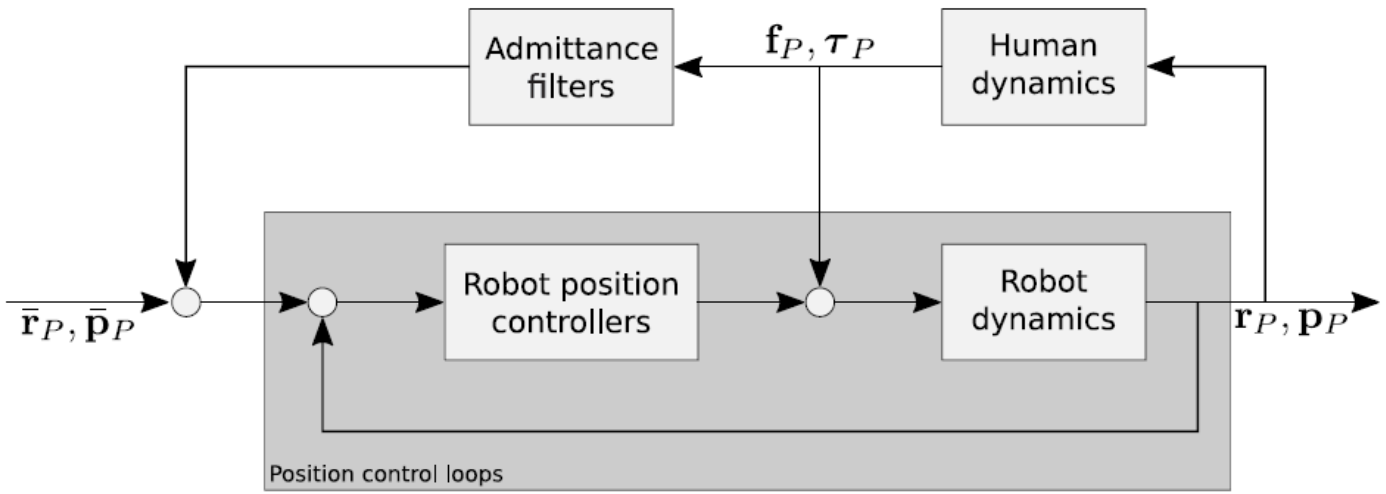
Hand guiding for programming by demonstration



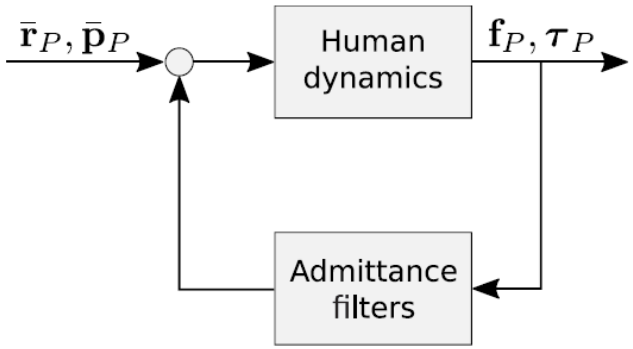
Stability with human in the loop

The forces applied by the human are not exogenous inputs. Instead they are the outputs of a model of human dynamics.

A stability analysis of the overall system can then be performed.

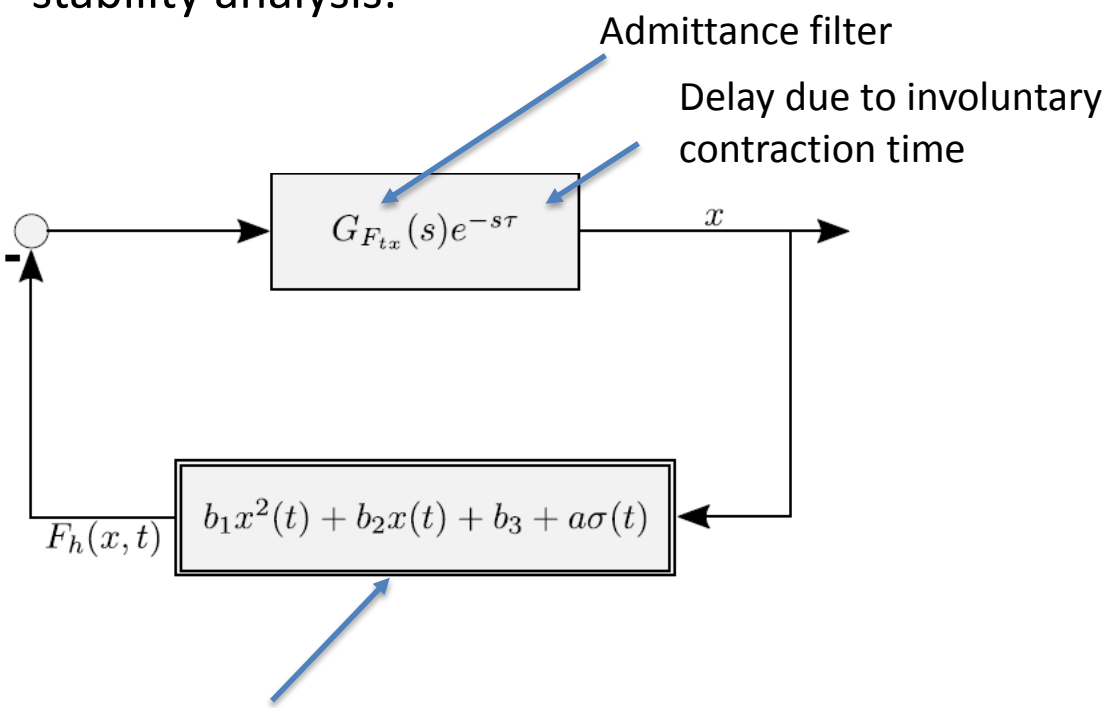


A simplified block diagram:

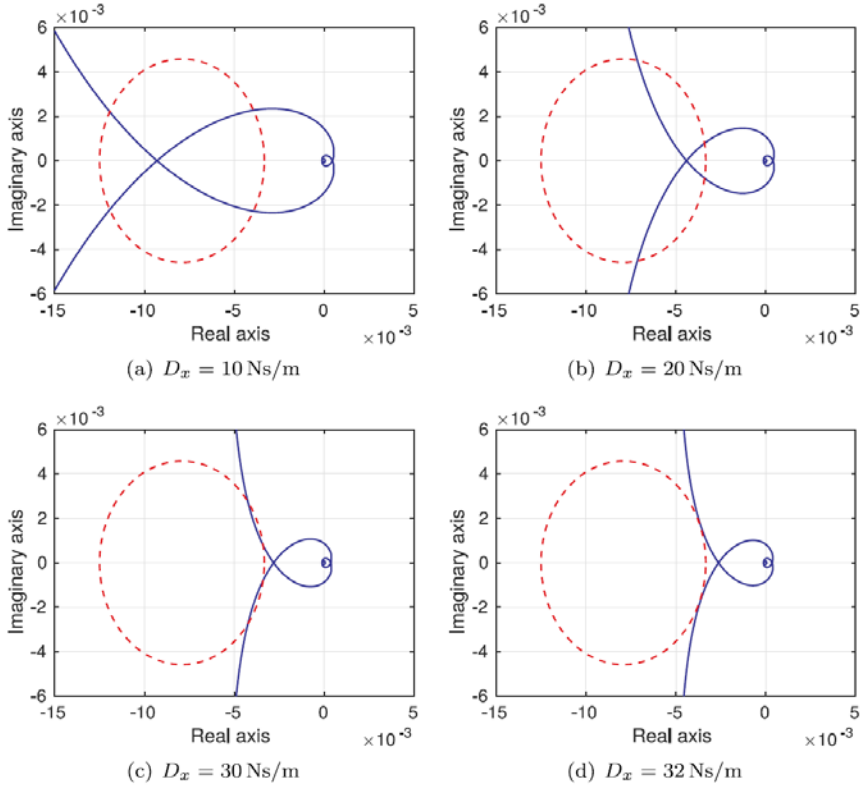


Stability with human in the loop

Closed loop system for stability analysis:



Bascetta and Ferretti, RCIM 2019



(a) $D_x = 10\,\mathrm{Ns/m}$ (b) $D_x = 20\,\mathrm{Ns/m}$ (c) $D_x = 30\,\mathrm{Ns/m}$ (d) $D_x = 32\,\mathrm{Ns/m}$

Absolute stability analysis yields guidelines on the parameters of the admittance filters

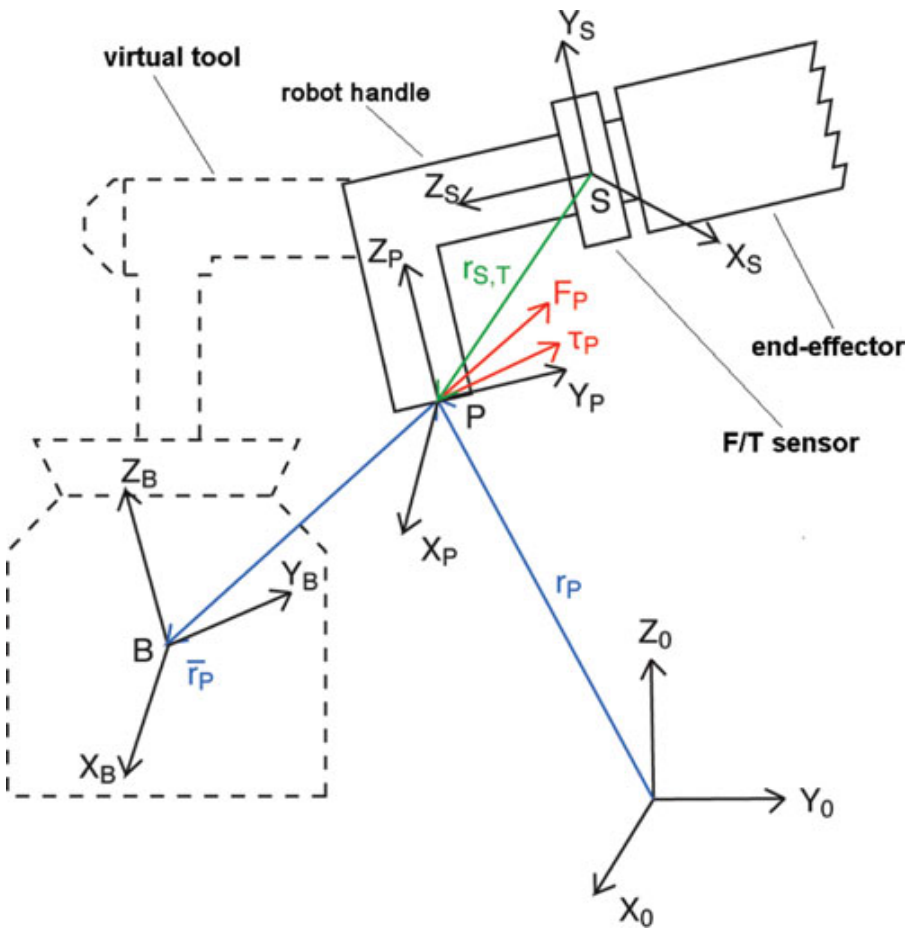
A more general virtual tool

Apart from the simple linear mass-damper system, more elaborated virtual tools can be considered.

By writing the equations of the virtual tool in a local frame:

$$\begin{cases} \boldsymbol{f}_p = m_p\left(\boldsymbol{a}_p - \boldsymbol{g} + \boldsymbol{\varepsilon}_p \times \boldsymbol{r}_p + \boldsymbol{\omega}_p \times \left(\boldsymbol{\omega}_p \times \boldsymbol{r}_p\right)\right) + \boldsymbol{D}_{1p}\boldsymbol{v}_p \\ \boldsymbol{\tau}_p = \boldsymbol{I}_p\boldsymbol{\varepsilon}_p + \boldsymbol{\omega}_p \times \left(\boldsymbol{I}_p\boldsymbol{\omega}_p\right) + \boldsymbol{r}_p \times \boldsymbol{f}_p + \boldsymbol{D}_{2p}\boldsymbol{\omega}_p \end{cases}$$

and integrating such equations based on the sensed forces and moments, the corresponding position and orientation displacements can be obtained.



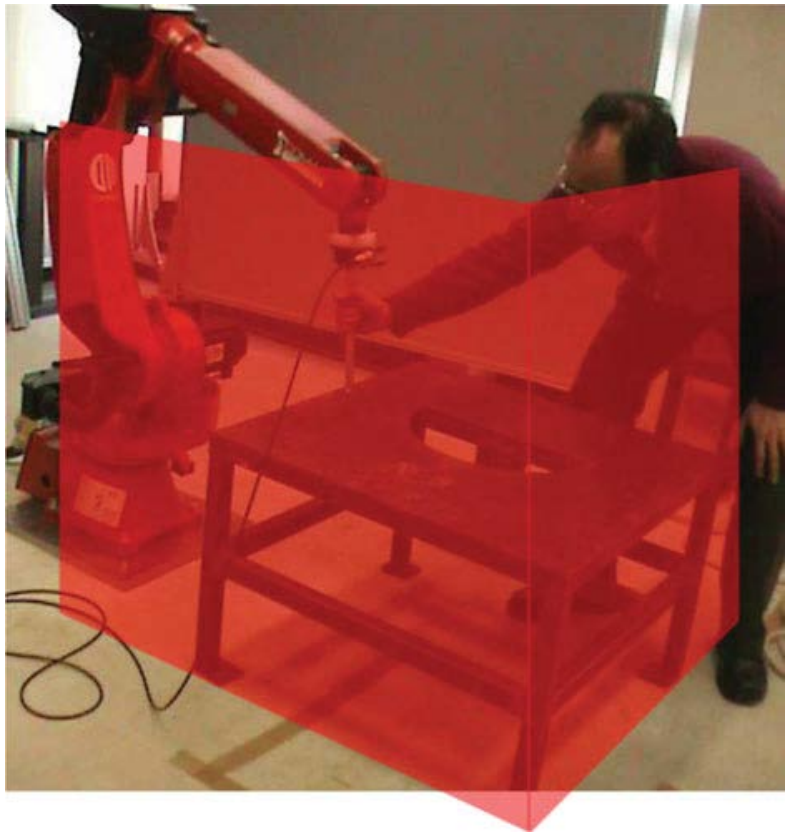
Bascetta, Ferretti, Magnani and Rocco, Robotica 2013

Virtual safety surfaces

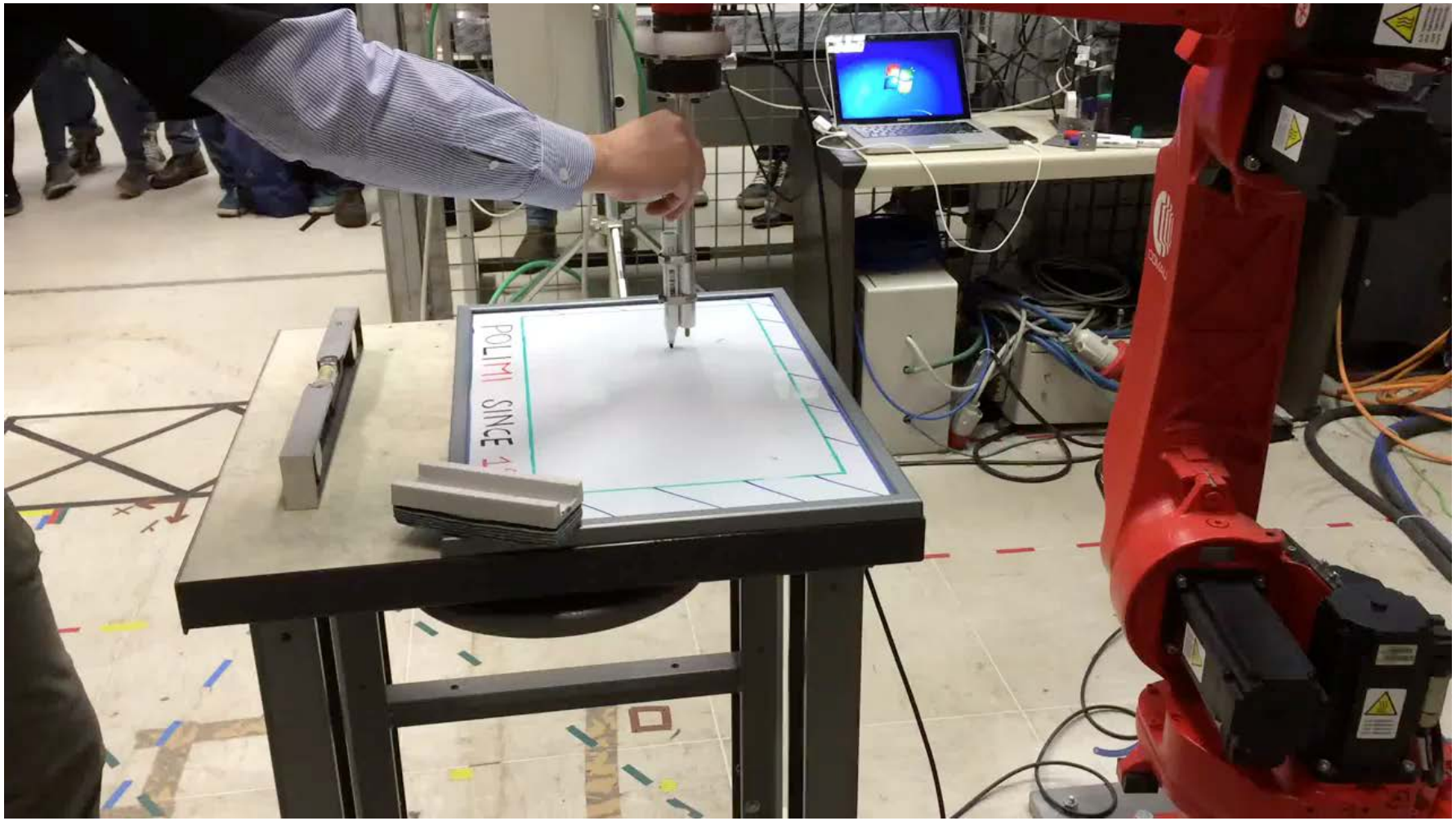
For safety purposes, a virtual separation between the robot end-effector and the human operator through a virtual safety surface.
The simplest example of such surface is a virtual wall, a plane that reacts to penetration generating a viscoelastic virtual force.

Adding this virtual force to the real forces measured by the sensor allows to prevent, as much as possible, the robot tool center point from entering the space behind the virtual wall.

The virtual and real forces are then fed together into the admittance filter to generate the robot motion.

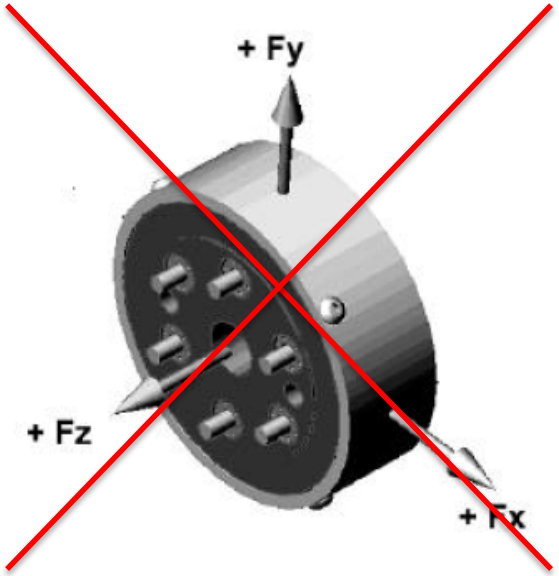


An admittance controller with contact force control



Sensorless hand guiding

What if the robot is not equipped with a force sensor?



We need to "compute" the forces from the dynamic model.

Very basic idea:

unknown human force

$$F + F_H = Ma$$

$$\hat{F}_H = Ma - F$$

requires acceleration!

Virtual force/torque sensor

Consider again the robot dynamic model where friction terms and external torques are included:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{F}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}^{ext}$$

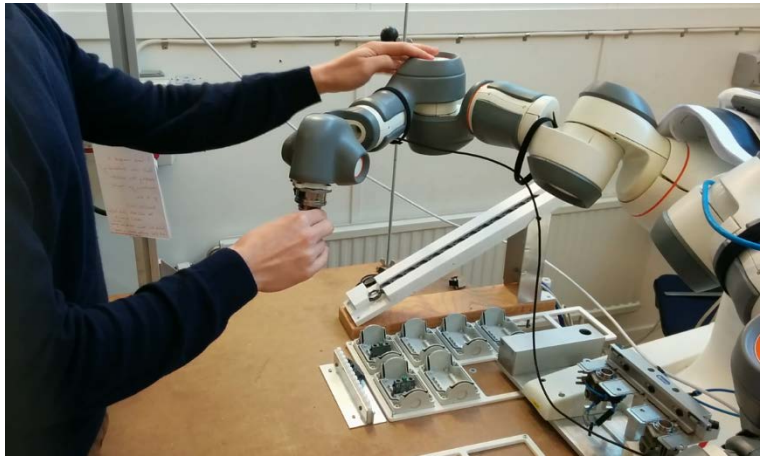
Define the generalized momentum:

$$\boldsymbol{p} = \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

Take the time derivative:

$$\dot{\boldsymbol{p}} = \left[\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})^T\right]\dot{\boldsymbol{q}} + \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} =$$

$$= \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})^T\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{\tau} + \boldsymbol{\tau}^{ext}$$



Virtual force/torque sensor

A residual vector can then be defined as:

Does not require accelerations!

$$\boldsymbol{r}(t) = \boldsymbol{K}\left[\boldsymbol{p}(t) - \int_0^t \left(\tau + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})^T \dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{r}(t)\right)dt - \boldsymbol{p}(0)\right]$$

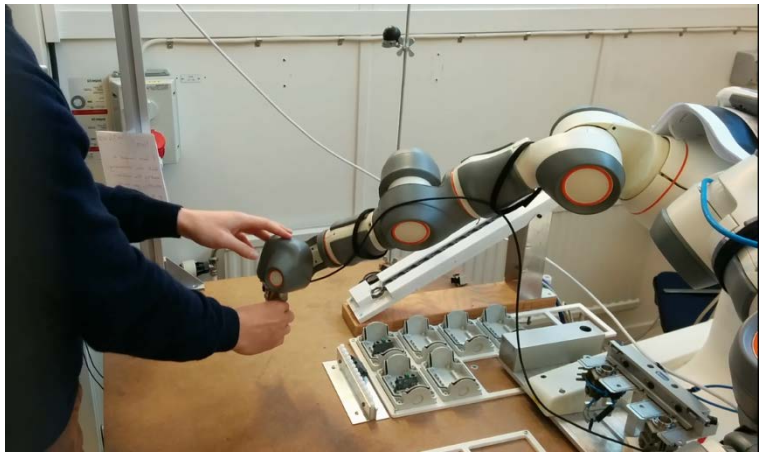
where $\boldsymbol{K}$ is a positive definite diagonal matrix.
Then:

$$\dot{\boldsymbol{r}}(t) = \boldsymbol{K}\left[\tau^{ext} - \boldsymbol{r}(t)\right]$$

meaning that the residual will converge to the external torques.
Forces/moments at the end effector can then be obtained through (pseudo)inversion of the Jacobian:

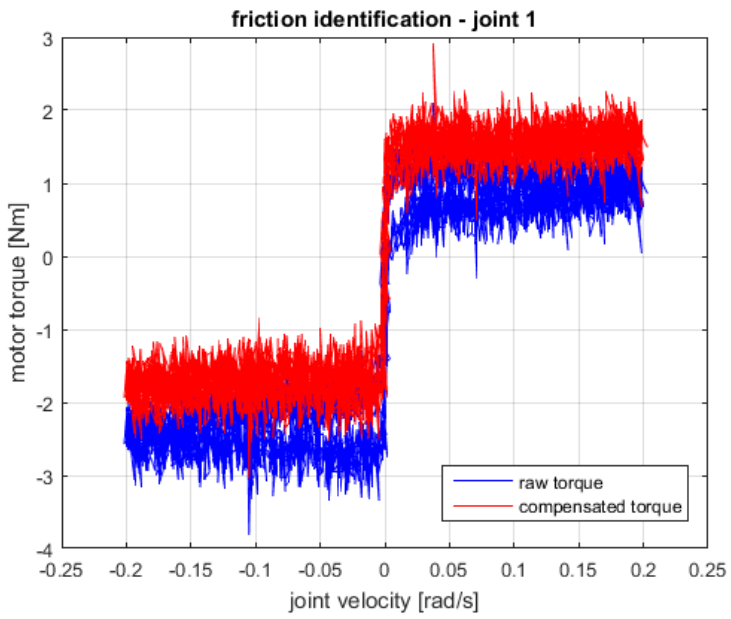
$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f} \\ \mu \end{bmatrix} = \left(\boldsymbol{J}(q)^T\right)^{\#} \boldsymbol{r}$$

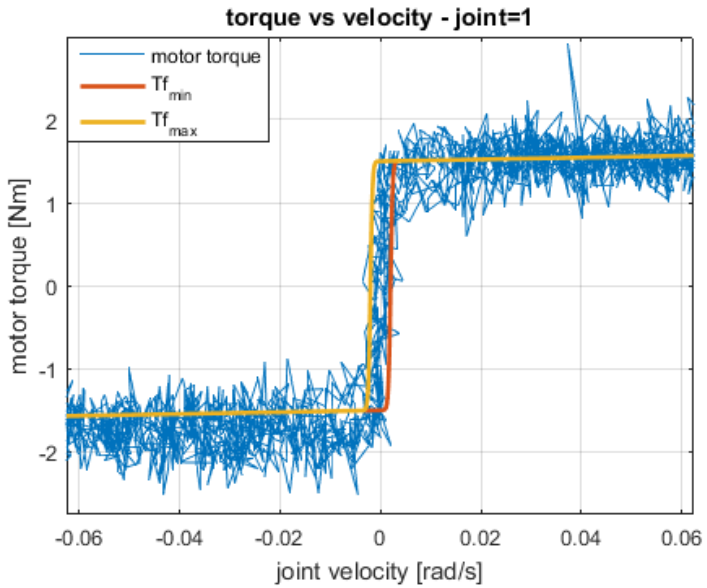
De Luca and Mattone, ICRA 2005

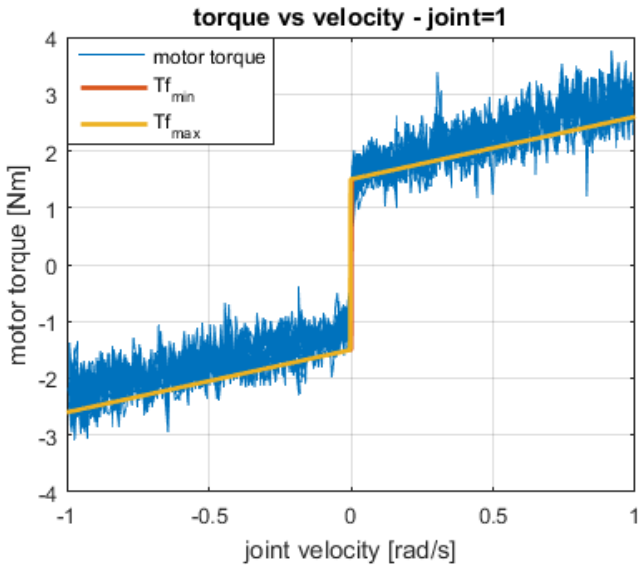


Friction in the joints

A major issue in the estimation of the external torques is friction in the joints
Friction needs to be estimated with ad-hoc experiments.



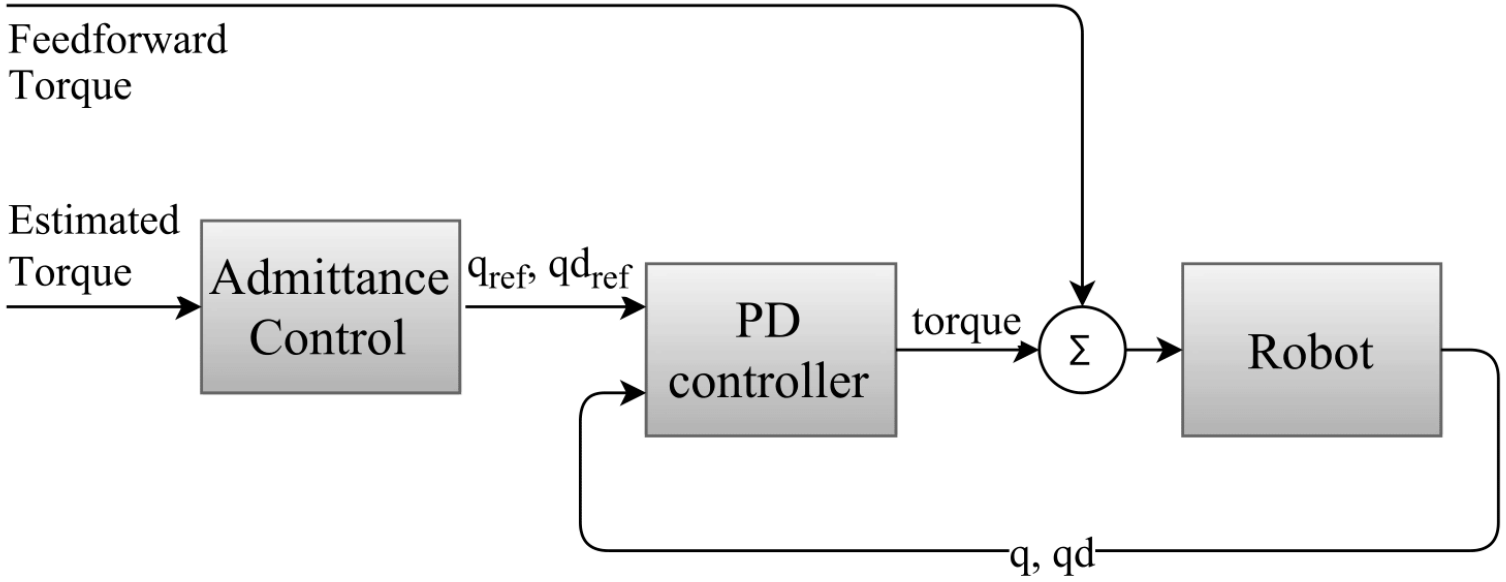




Torque due to joint friction
has to be isolated.

A curve (sigmoid) ha to be fitted to data.
A dithering torque can be used to mitigate the effects of friction

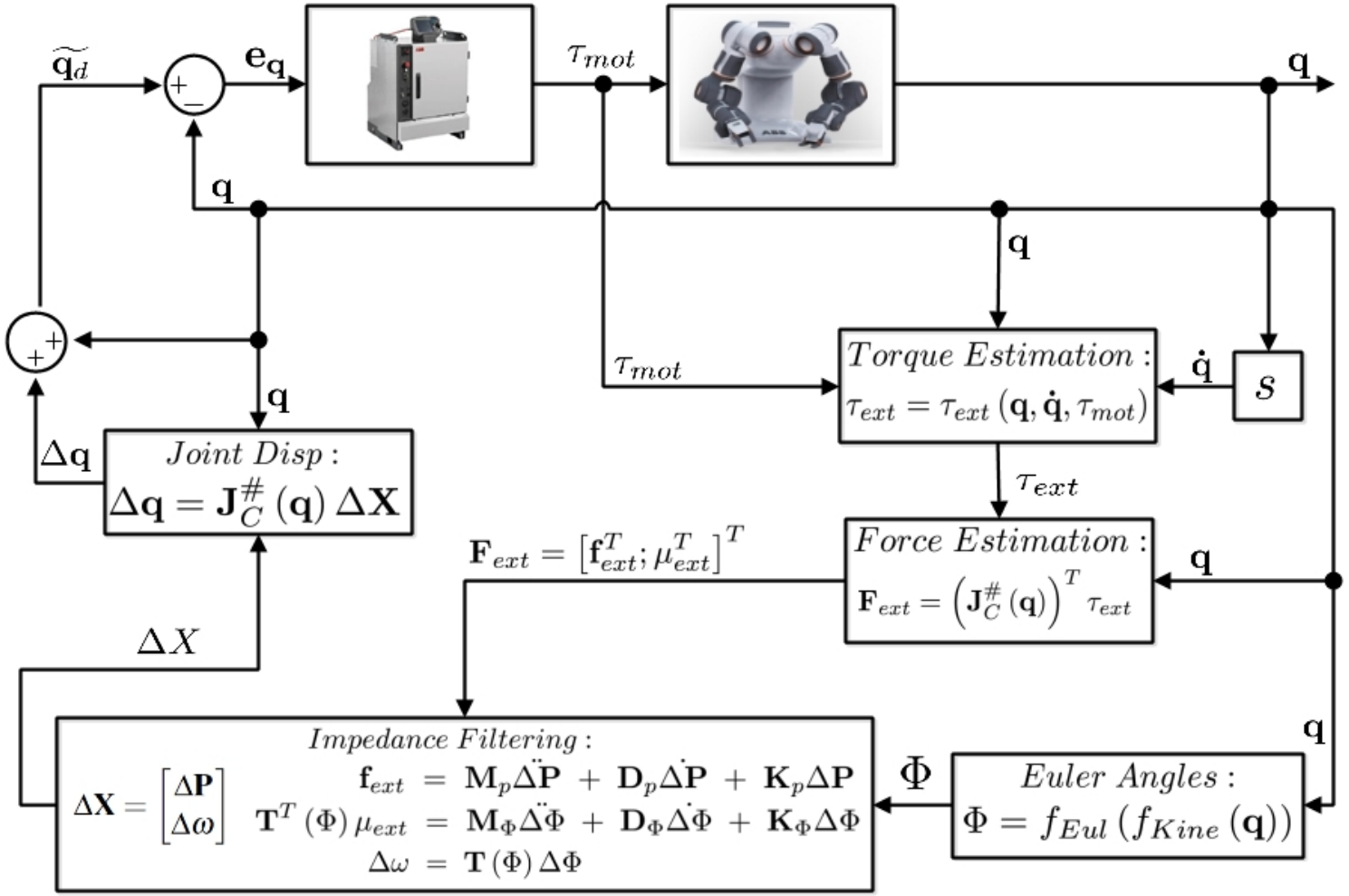
An implementation of the method (LUND University)



Capurso, Ghazaei Ardakani, Johansson, Robertsson and Rocco, ICRA 2017

An implementation of the method (LUND University)

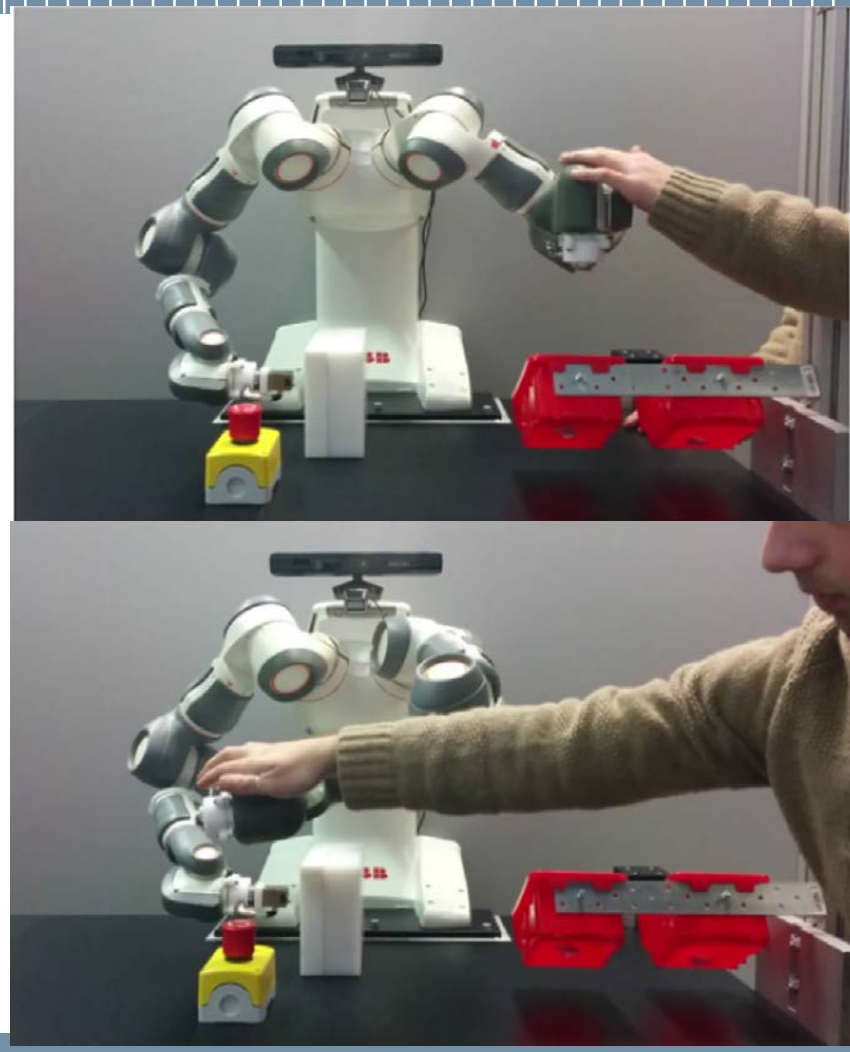
An implementation of the method (POLIMI)



Motion along a specific Cartesian direction

Often the user wants to manually guide the robot along a specific Cartesian direction. Applying a force in that direction only might be problematic.

We identify the largest Cartesian component of the applied force/momentum and we force the manipulator end-effector to move/rotate exclusively along/around the corresponding Cartesian axis.

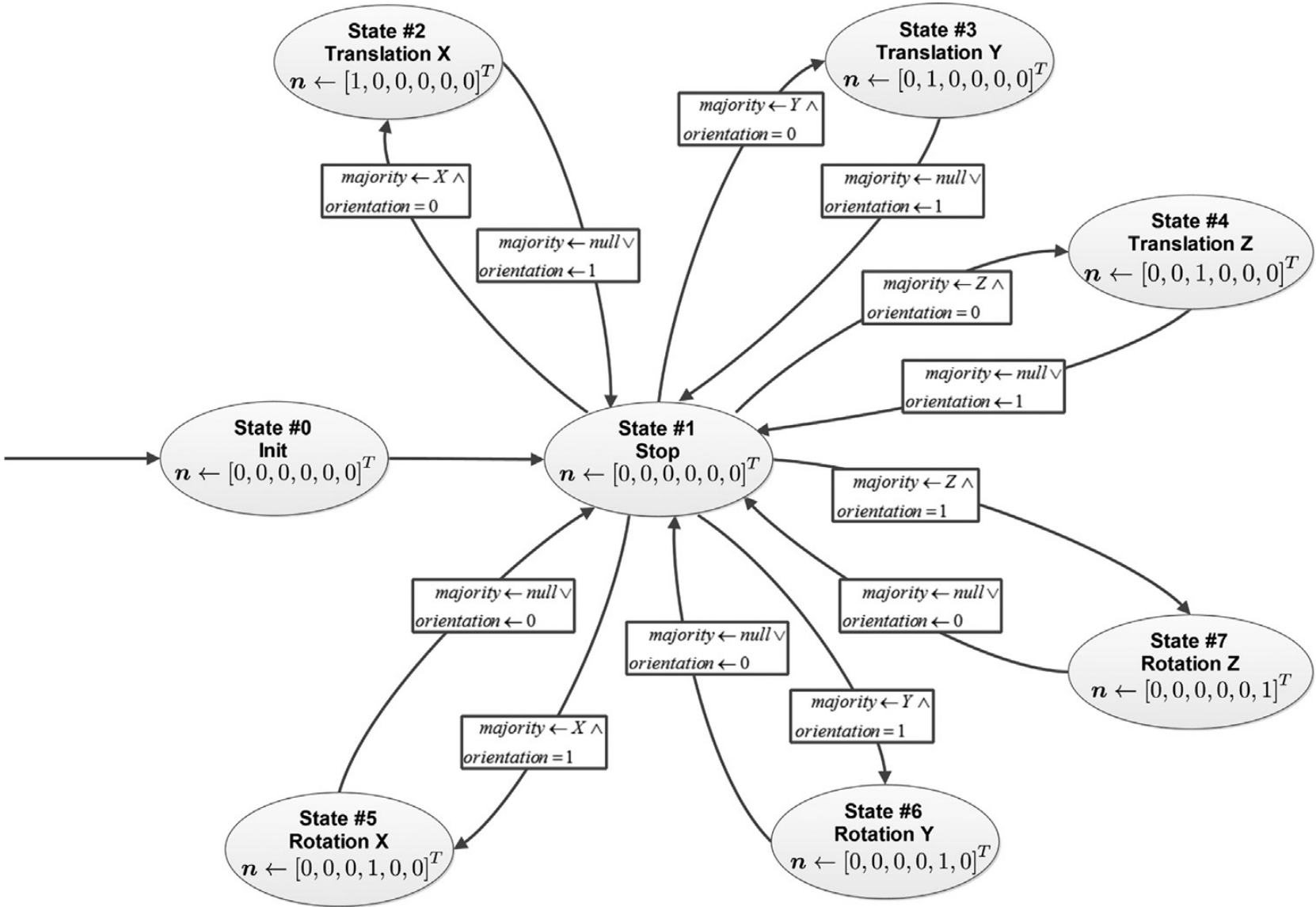


Ragaglia, Zanchettin, Bascetta and Rocco, RCIM 2016

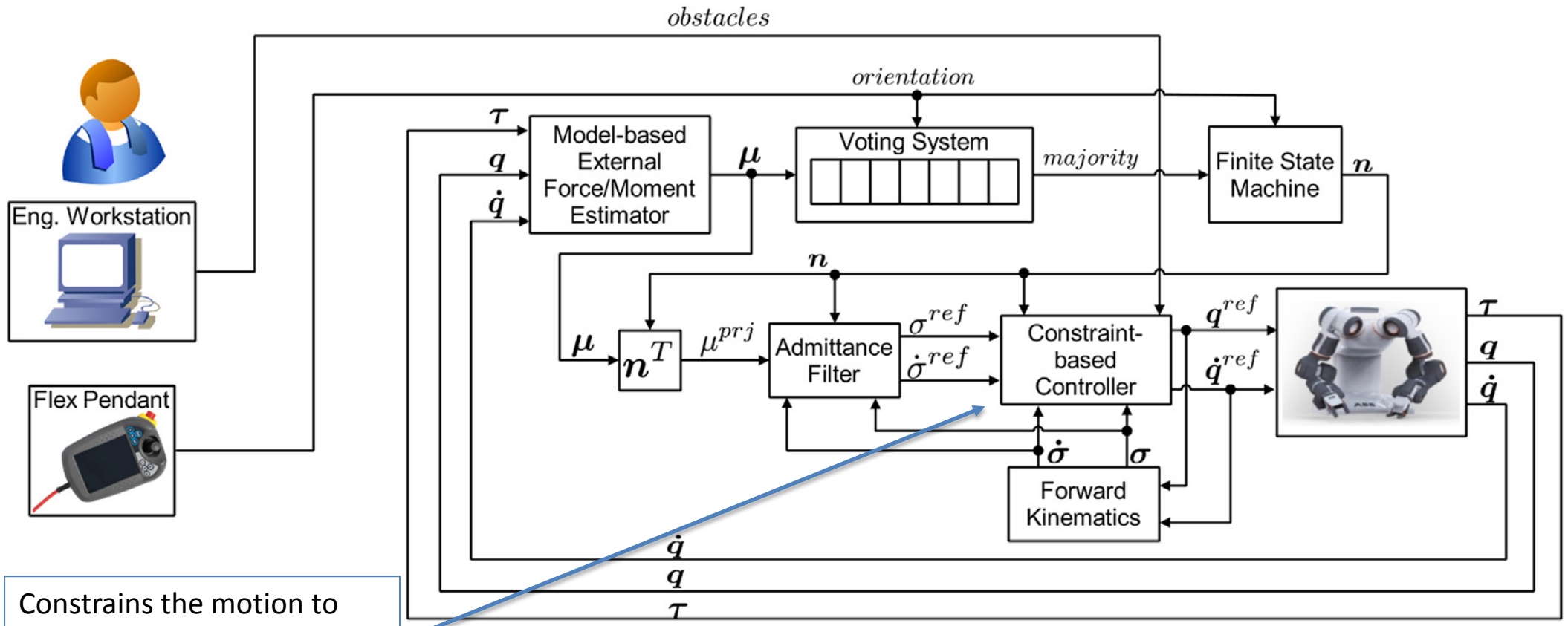
Motion along a specific Cartesian direction

Voting system

- It is a circular buffer that collects votes.
- These votes are assigned to the three Cartesian axes X, Y and Z on the basis of two distinct inputs: the estimated applied forces/moments and a boolean flag, “orientation”, set by the operator.
- When a significantly large number of votes is assigned to the same Cartesian axis, we assume that the operator wants to enforce a translational/rotational motion along/around that direction.

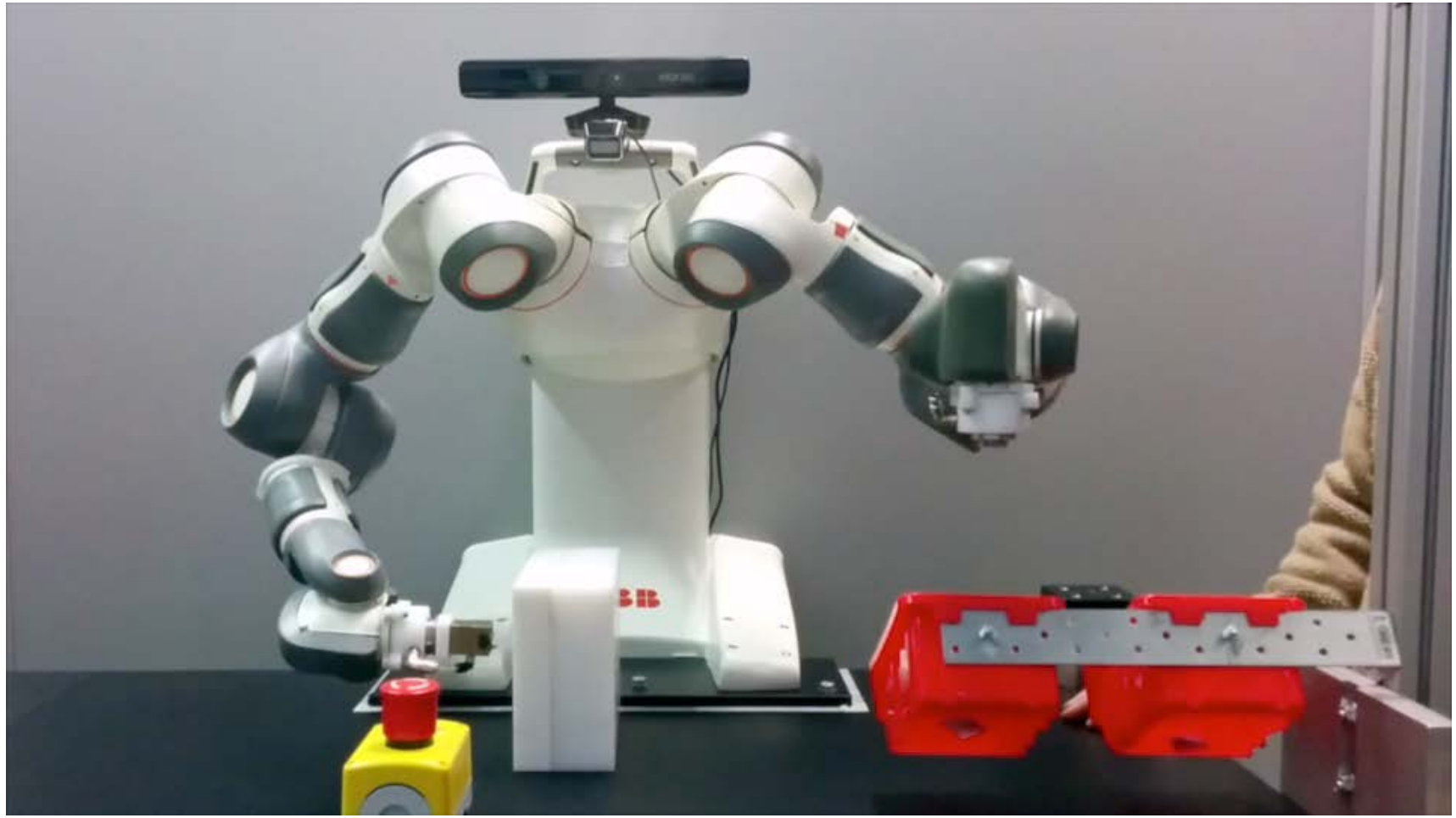


Motion along a specific Cartesian direction



Constrains the motion to evolve along a specific Cartesian direction

Motion along a specific Cartesian direction



POLITECNICO
MILANO 1863