H0-Task Space Control-Cartesian space

Cartesian Space Control

Inverse Kinematics and Joint Motion Control

Direct Cartesian Space Control

CARTESIAN SPACE CONTROL

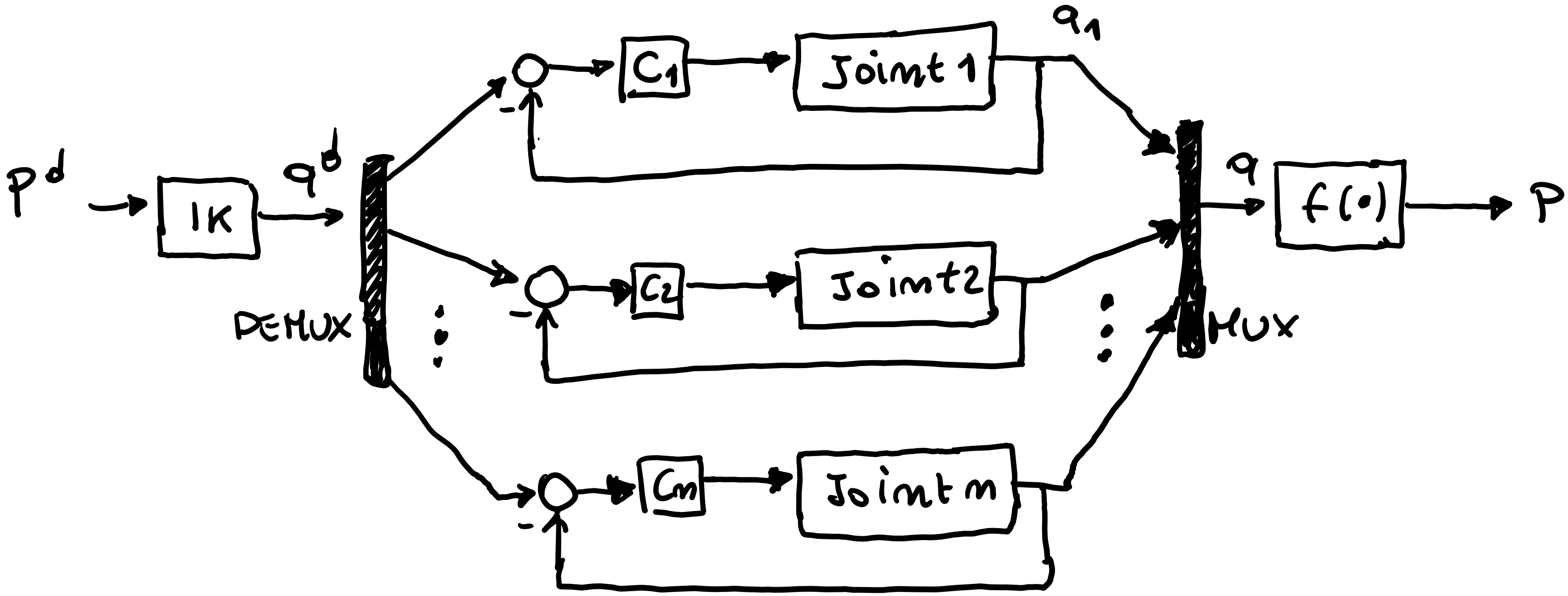
- given a reference trajectory for the end-effector $p^d(t)$, compute $u(t)$ such that $p(t) \approx p^d(t)$ where $p \in \mathbb{R}^3$ is the position of the end-effector

(A) IDEA #1: INVERSE KINEMATICS FOR END-EFFECTOR CONTROL

Compute $q^d(t)$ such that $x^d(t) = f(q^d(t))$

where $f(\cdot)$ is the forward kinematics function.

Then apply joint motion control



if $q \to q^d$ then $p \to p^d$

⊕ simple

⊕ joint feedback rejects perturbations that do not affect end-effector

⊖ constant joint stiffness results in time-varying stiffness at E-E if config. changes

⊖ Need IK (may be slow in the numeric case)

B DIRECT CARTESIAN SPACE CONTROL

- The error driving the feed-back action is defined at the end-effector level on the Cartesian space coordinates

- The desired trajectory is defined directly in the cartesian space (no kinematic inversion)

- measurements in the cartesian space are obtained by forward kinematics computations of joint variables

- Idea: instead of defining a motion for the joints, directly control the forces that act on the end-effector

kinematics : $p = f(q) \quad \dot{p} = J(q)\dot{q}$

PANORAMIC VIEW

assume control commands are always Joint Torques

definition of error / type of Task		JOINT SPACE (ref. desired configuration)	TASK SPACE (reference desired pose)
free motion	Regulator (initial/final)	P, PD, PID gravity compensation	PD + gravity compensation
	Traj. Tracking	feed back linearization (JSID)	feedback linearization (TSID)
motion in contact			Impedance / admittance control (with variants)

CARTESIAN PD + GRAVITY COMPENSATION

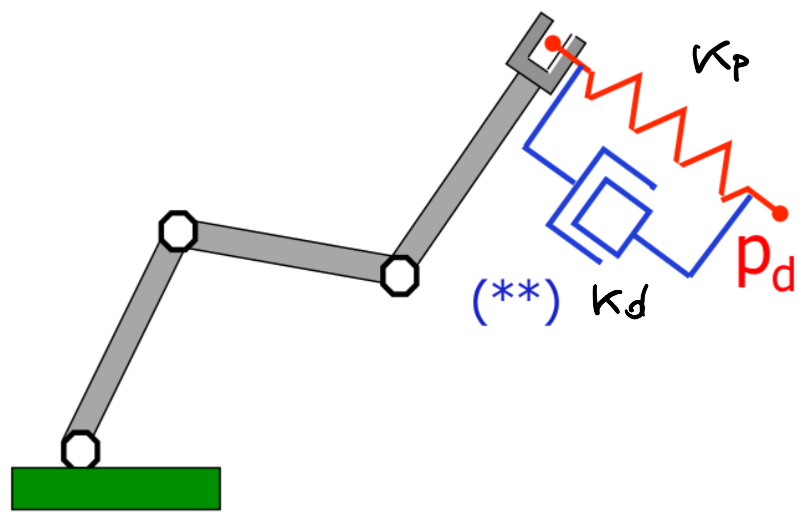
idea: create a virtual elastic / viscous attractor at the end-effector and convert their force / moments into joint torques

- assumption $n = m$ (no redundancy)

$$\boxed{u = J^T(q)\left[K_p(p^d - p) + K_d(\dot{p}^d - \dot{p})\right] + g(q)} \qquad K_p \succ 0,\ K_d \succ 0, \quad K_p, K_d \in \mathbb{R}^{m \times m}$$

Recall

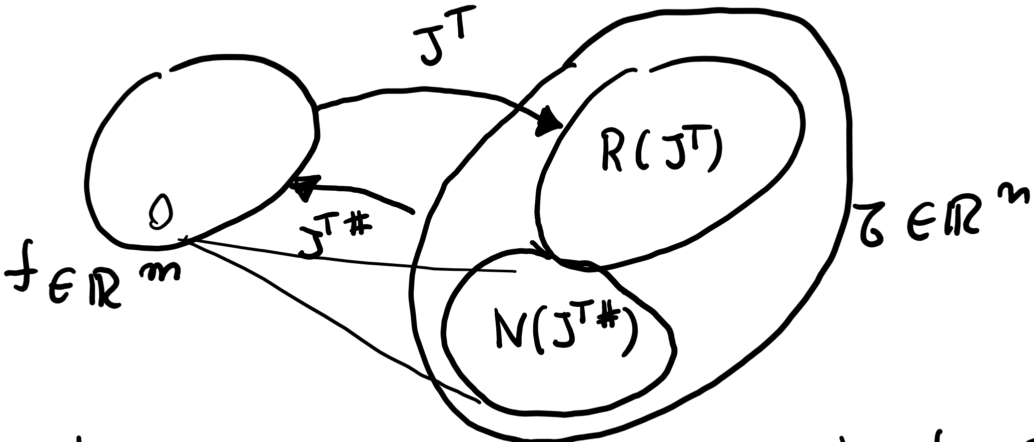
$\tau = J^T F$



- J^T maps force / moments from end-effector to joint space, because $n = m \Rightarrow J^T$ square

REDUNDANT MANIPULATORS

$n > m$ ➡ $n-m$ joint coordinates are undefined



$$A^{\#} = (A^T A)^{-1} A^T$$

$J^T = [\ ]$ "skinny"

employ a null-space method to add a postural task that tries to keep the robot in a default configuration

$$\boxed{u = J^T f^d + [I - J^T J^{T\#}]\, \tau_0}$$

Null space projector of $J^{T\#}$

where:

$$f^d = K_p (p^d - p) + K_d (\dot{p}^d - \dot{p})$$ cartesian Task

$$\tau_0 = K_q (q_0 - q) - K_{\dot{q}} (\dot{q})$$ postural Task

PANORAMIC VIEW

assume control commands are always joint torques

type of task \ definition of error		JOINT SPACE (ref. desired configuration)	TASK SPACE (reference desired pose)
free motion	Regulator (initial/final)	P, PD, PID gravity compensation	PD + gravity compensation
free motion	Traj. Tracking	feedback linearization (JSID)	feedback linearization (TSID)
motion in contact			Impedance / admittance control (with variants)

(c) COMPUTED TORQUE IN CARTESIAN SPACE

Joint space dynamics: $\boxed{M\ddot{q} + R = u}$ (1)

assumption $m = n$ → non-redundant manipulator

- To achieve the "inversion" we need first to consider the dynamic behaviour of the manipulator at the end-effector (cartesian dynamics)

$$\dot{p} = J\dot{q} \Rightarrow \ddot{p} = J\ddot{q} + \dot{J}\dot{q} \Rightarrow \boxed{\ddot{q} = J^{-1}(\ddot{p} - \dot{J}\dot{q})} \quad (2)$$

replacing (2) into (1) we obtain:

$$M J^{-1}(\ddot{p} - \dot{J}\dot{q}) + R = u$$

if we left-multiply by J^{-T} (J not singular)

$$\underbrace{(J^{-T} M J^{-1})}_{\Lambda} \ddot{p} + \underbrace{J^{-T} R - (J^{-T} M J^{-1}) \dot{J}\dot{q}}_{\mu} = \underbrace{J^{-T} u}_{F}$$

$$\boxed{\Lambda(q)\ddot{p} + \mu(q,\dot{q}) = F} \quad (1)$$ dynamics reflected at the end-effector

$\Lambda(q)$: $J^{-T} M J^{-1}$ cartesian inertia = joint space inertia as "seen" by the end-effector

Note 1: Λ is singular if J becomes singular ($\nexists\, J^{-1}$)

$\mu(q,\dot{q})$: $J^{-T} R - \Lambda \dot{J} \dot{q}$ cartesian bias terms

F : cartesian force

Now we can apply feedback linearization technique to linearize (1):

$$\boxed{F = \hat{\Lambda}(q) v + \hat{\mu}(q,\dot{q})}$$

assuming perfect knowledge of Λ, μ :

$$\Lambda \ddot{p} + \cancel{\mu} = \Lambda v + \cancel{\mu}$$

$$\boxed{I_{m \times m} \ddot{p} = v} \quad (2) \qquad \text{unit mass !}$$

- m independent double integrators

Decoupled system!

Finally we can design a stabilizing controller for the tracking error in the m double integrators of (2):

$$\boxed{v = \ddot{p}^d + K_p (p^d - p) + K_d (\dot{p}^d - \dot{p})}$$

$K_p > 0 \quad K_d > 0$ and diagonal

resulting in the closed loop dynamics:

$$I_{m \times m} \underbrace{(\ddot{p}^d - \ddot{p})}_{\ddot{e}_x} + K_p \underbrace{(p^d - p)}_{e_x} + K_d \underbrace{(\dot{p}^d - \dot{p})}_{\dot{e}_x} = 0$$

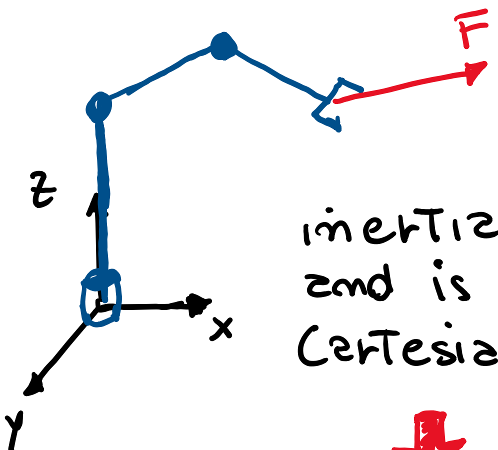
$e_x \in \mathbb{R}^m$: cartesian tracking error

$$\ddot{e}_x + K_d \dot{e}_x + K_p e_x = 0 \quad \Rightarrow \text{ Stable } \forall K_p, K_d \geq 0$$

The final control law is:

$$\boxed{u = J^T \left[\hat{\Lambda}(q) \left(\ddot{p}^d + K_p e_x + K_d \dot{e}_x \right) + \hat{\mu}(q, q) \right]}$$

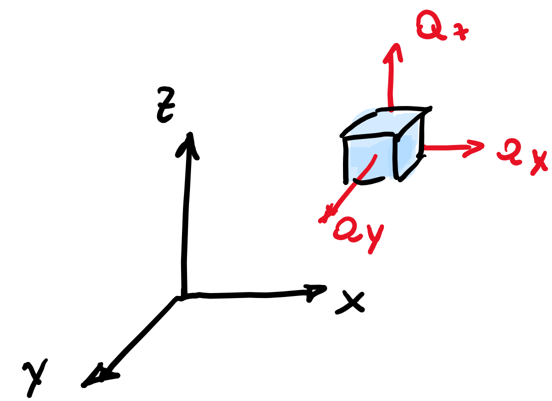
PHYSICAL INTERPRETATION



ID $\ddot{p} = a$

inertia depends on q and is variable in different Cartesian directions

when you apply a force F the E-E accelerates in a different direction



unitary mass

accelerates in the same direction of the applied force

REDUNDANT MANIPULATORS

$m < n \Rightarrow J$ is rectangular (fat) $\Rightarrow J^{-1}$ doesn't exist!

- we can obtain only input-output decoupling / linearization (but not of the whole state dynamics) by replacing J^{-T} with $J^{T\#}$

Alternative derivation:

$M\ddot{q} + h = u \rightarrow$ left-multiply by JM^{-1}

$J\ddot{q} + JM^{-1}h = JM^{-1}u$

$\ddot{p} - \dot{J}\dot{q} + JM^{-1}h = JM^{-1}u$

let's consider an alternative expression of Λ that exists for redundant manipulators:

$$\boxed{\Lambda = (JM^{-1}J^T)^{-1}}$$

$\rightarrow$ defined if J is full row rank!

left-multiplying by Λ:

$$\Lambda\ddot{p} - \Lambda\dot{J}\dot{q} + \Lambda JM^{-1}h = \Lambda JM^{-1}u$$

$$\Lambda \ddot{p} + \underbrace{\Lambda\left(J M^{-1} h - \dot{J}\dot{q}\right)}_{\mu} = \underbrace{\Lambda J M^{-1}}_{(J M^{-1} J^T)^{-1} J M^{-1} \Rightarrow J^{T\#}} u$$

proof: $A = J^T = [\ ]$ skinny

$$(A^T M^{-1} A)^{-1} A^T M^{-1} \Rightarrow A^{\#}$$

for a redundant robot : $F = J^{T\#} u$

$$\boxed{\Lambda \ddot{p} + \mu = F}$$ cartesian dynamics

- due to the redundancy...

it will remain an internal dynamics of dimension $n-m$ that should be stabilyzed by a postural task in the null-space of $J^{T\#}$

exploit torques that do not generate any end-effector force (an so motion) to control the redundant DoFs:

$$u \in N(J^{T\#}) = \{ u \in \mathbb{R}^n : J^{T\#} u = 0 \}$$

any u_0 such that $J^{T\#} u_0 = 0$ does not affect $\ddot{p}$

$$\Lambda \ddot{p} + \mu = J^{T\#}(u + u_0) = J^{T\#} u$$

I can compute u_0 as : $u_0 = [I - J^T J^{T\#}] u_p$

verify $J^{T\#} u_0 = 0$

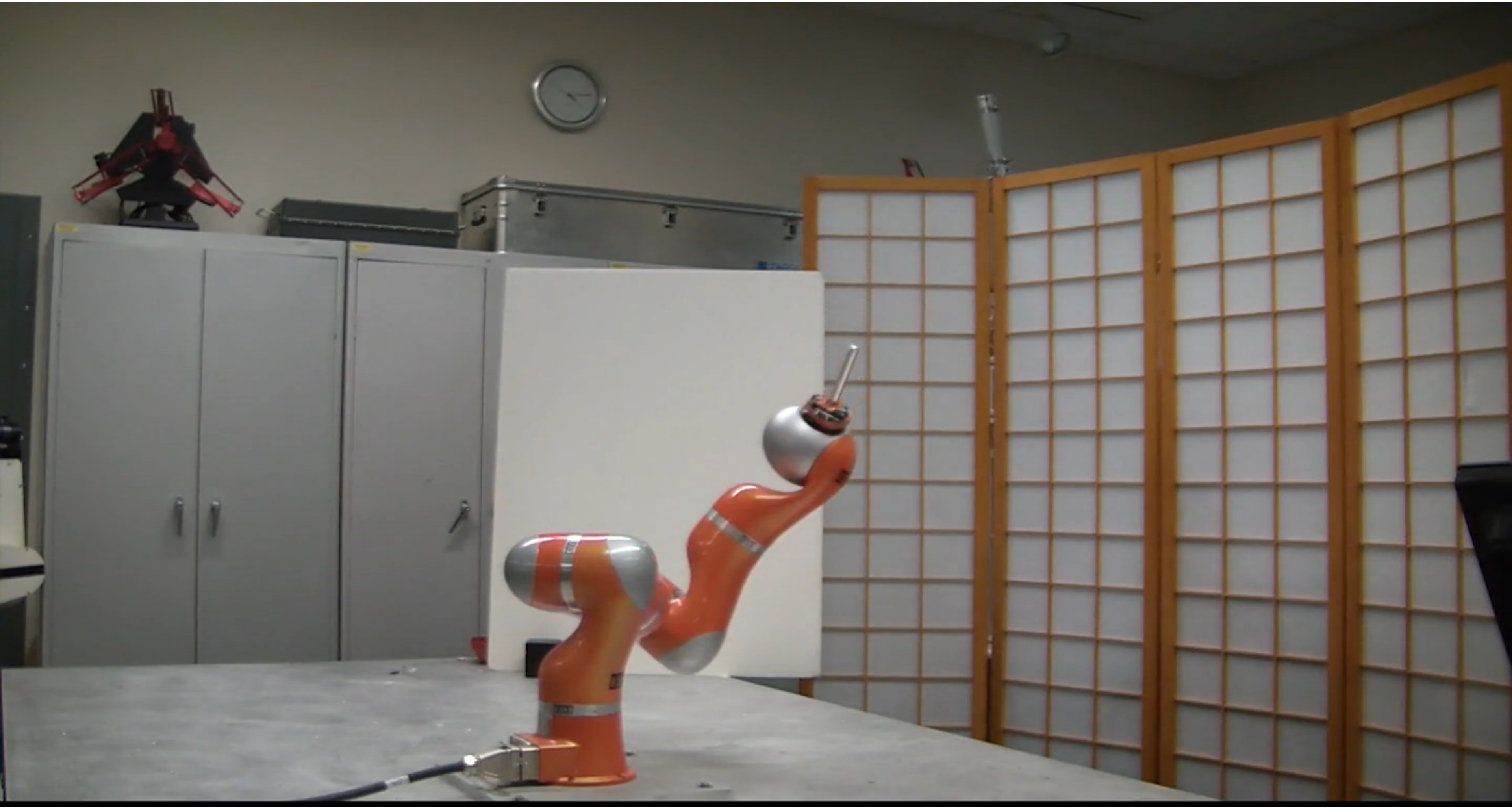
$$= J^{T\#}[I - J^T J^{T\#}] u_p = (J^{T\#} - \cancel{J^{T\#}} \cancel{J^T} J^{T\#}) u_p = 0$$

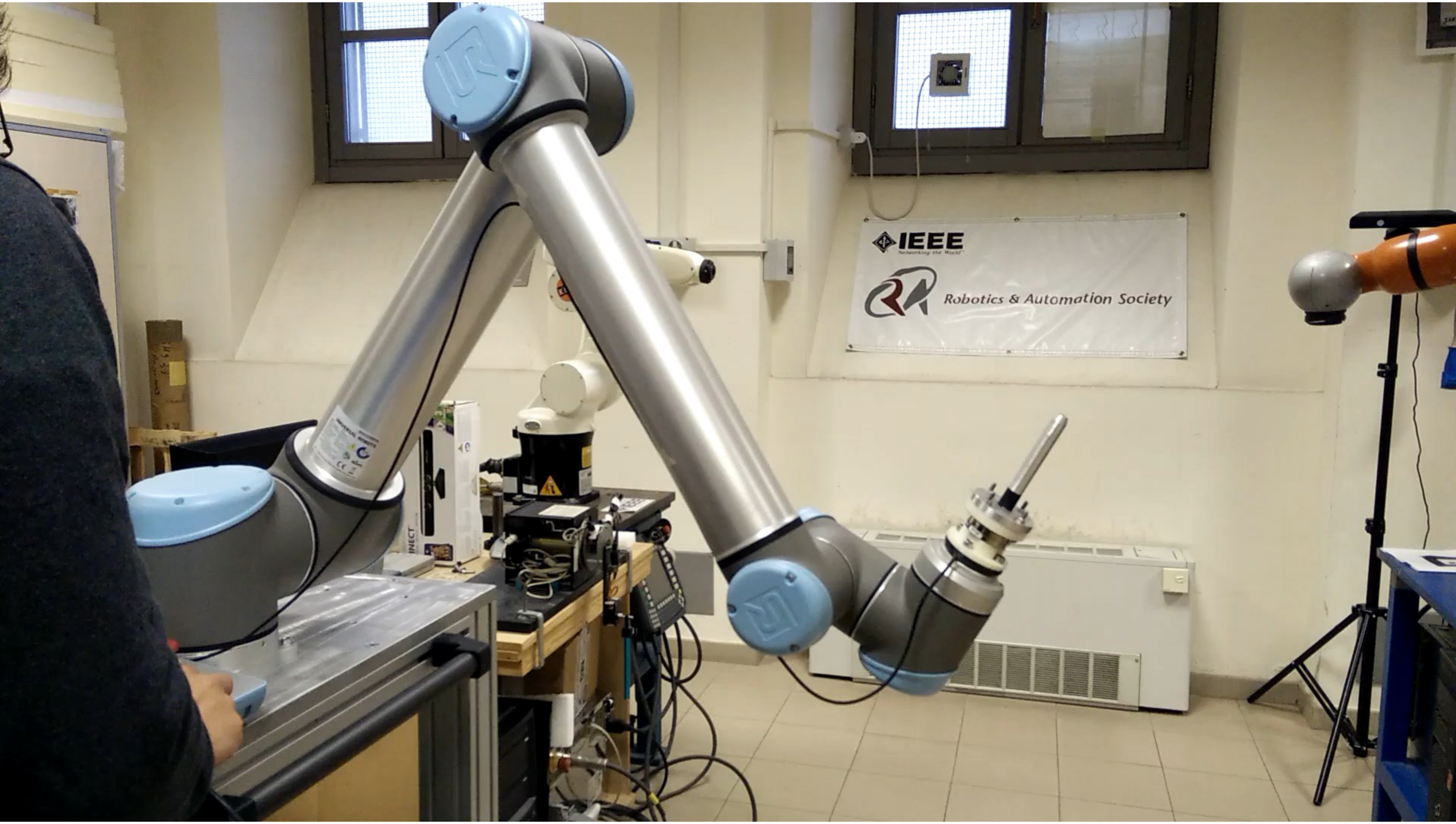
The final control law will be:

$$\boxed{u = J^T [\Lambda v + \hat{\mu}] + [I - J^T J^{T\#}] u_p}$$

where u_p is the postural task that stabilizes the internal dynamics:

$$u_p = K_q(q^d - q) - k_{\dot{q}} \dot{q}$$




IEEE
Networking the World
Robotics & Automation Society

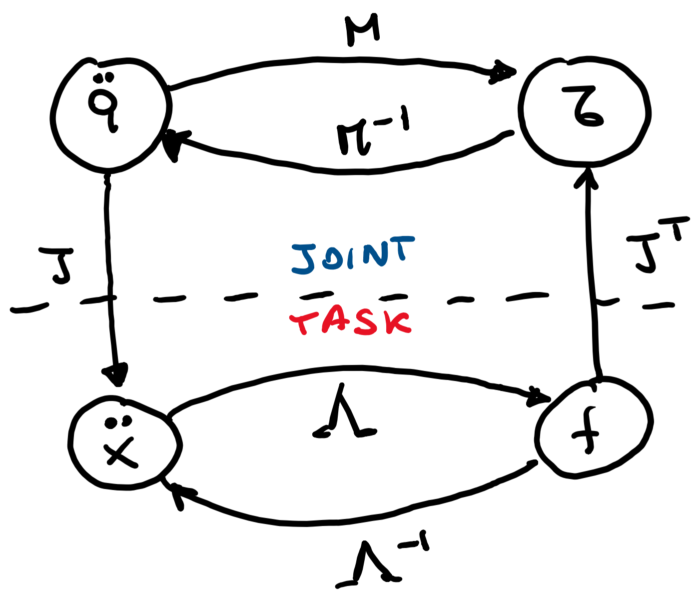
REMARKS ON CARTESIAN SPACE INVERSE DYNAMICS

⊕ can achieve decoupling in cartesian space

⊖ more complex of the joint space version ⇒ requires the computation of J that is influenced by the presence of

- redundancy → add postural task in NS
- singularities → implement singularity robustness as for IK

RECAP MAPPINGS



M: "Joint accel." $\longrightarrow$ "Joint Torques"

M^{-1}: "Joint Torques" $\longrightarrow$ "Joint accel"

J: "Joint velocities" $\longrightarrow$ "Task velocities"

"Joint accel." $\longrightarrow$ "Task accel."

J^T: "Task forces" $\longrightarrow$ "Joint Torques"

$\Lambda^{-1} = (J M^{-1} J^T)$: "Task forces" $\longrightarrow$ "Task accel."

$\Lambda = (J M^{-1} J^T)^{-1}$: "task accel." $\longrightarrow$ "Task forces"

① INVERSE KINEMATICS + JOINT COMPUTED TORQUE

Idea: solve IK at the acceleration level to get $\ddot{q}^d$
Then use joint space inverse dynamics to get u

assume redundant manipulator (general case)

$$\ddot{p} = J\ddot{q} + \dot{J}\dot{q} \Rightarrow \ddot{q}^d = J^{\#}(v - \dot{J}\dot{q}) + (I - J^{\#}J)q_0$$

plug it into the JSID to obtain joint torques

$$\boxed{u = M\ddot{q}^d + h = M\left[J^{\#}(v - \dot{J}\dot{q}) + (I - J^{\#}J)q_0\right] + h}$$

assumption: J full row rank otherwise $\nexists J^{\#}$

⊕ easier to implement

⊕ faster to compute (no need to compute Λ)

verify closed-loop behaviour:

$$\cancel{M}\ddot{q} + \cancel{h} = \cancel{M}\ddot{q}^d + \cancel{h}$$

$$\ddot{q} = J^{\#}(v - \dot{J}\dot{q}) + (I - J^{\#}J)q_0$$

$$J\ddot{q} = \cancel{J}\cancel{J^{\#}}(v - \dot{J}\dot{q}) + (\cancel{J} - J\cancel{J^{\#}}J)q_0$$

$$J\ddot{q} + \dot{J}\dot{q} = v$$

$$\boxed{\ddot{p} = v}$$ same as in (c)

References:

- D. E. Whitney, Resolved *motion rate* control of manipulators and human, 1969.
- O. Kathib, A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation, 1987.
- P. Wensig, AME 50551 – Introduction To Robotics (L36): http://sites.nd.edu/pwensing/ame-50551-introduction-to-robotics/