# G1-Robot Control

## Inverse dynamics

# Outline

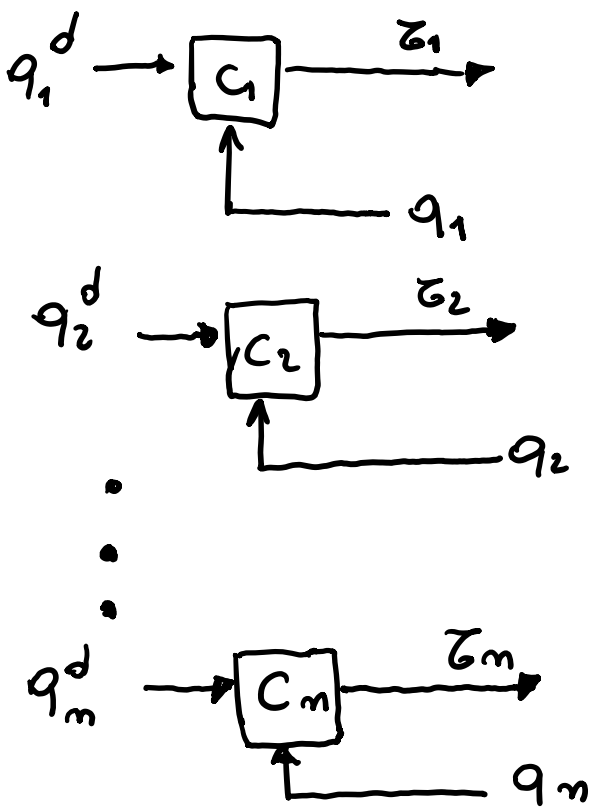
Decentralized Control

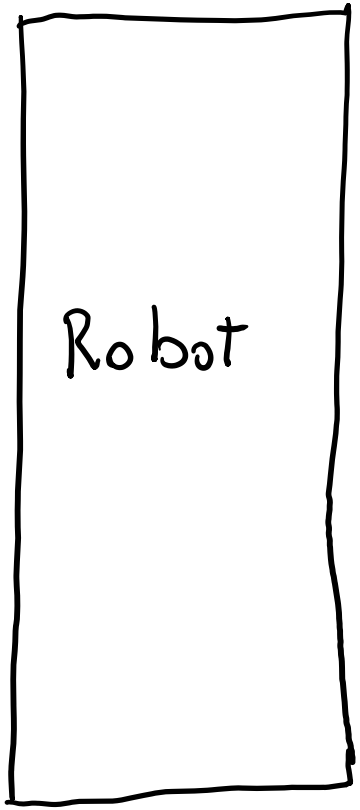
Inverse Dynamics Compensation

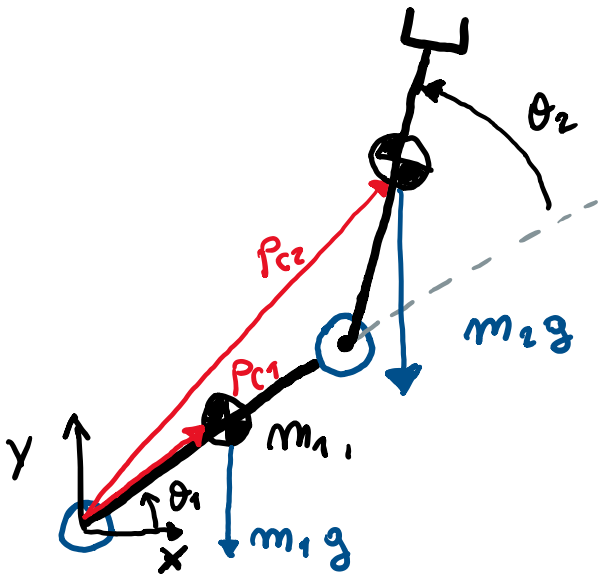
Joint-Space Computed Torque Control

# DECENTRALIZED CONTROL

- independent controllers for each joint
- non model based
- control system is composed of n SISO (Single Input/Single Output) loops, ignoring the dynamic coupling effects that are dealt as disturbances



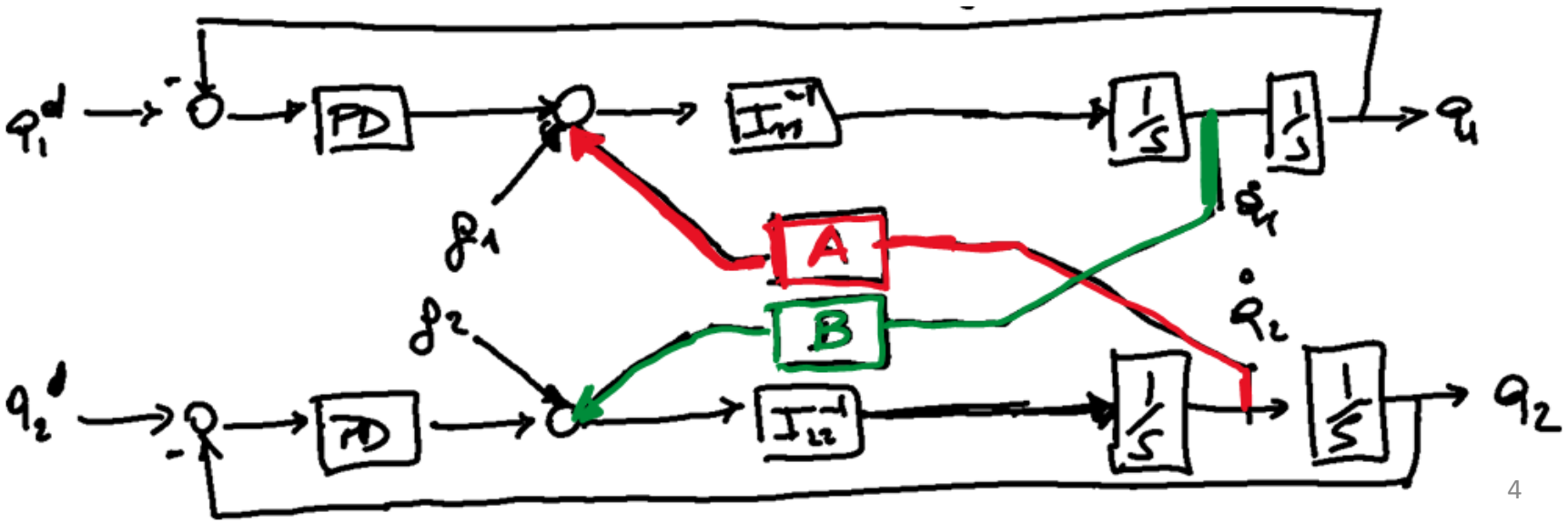



$$I_{11}\ddot{q}_1 + \underbrace{I_{12}\ddot{q}_2 + I_{112}\dot{q}_1\dot{q}_2 + I_{112}\dot{q}_2^2}_{A} + g_1 = \tau_1$$

$$I_{22}\ddot{q}_2 + \underbrace{I_{21}\ddot{q}_1 - \frac{I_{112}}{2}\dot{q}_1^2}_{B} + g_2 = \tau_2$$

A = influence of JOINT 2 on JOINT 1

B = influence of JOINT 1 on JOINT 2



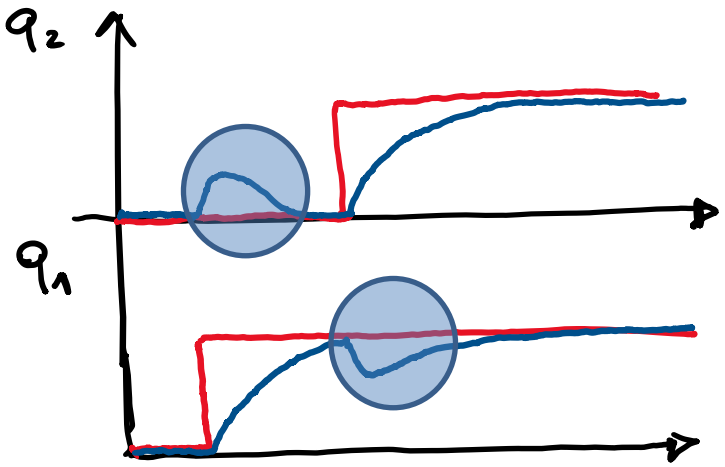
- # Decentralized control drawbacks

For a multi-dof articulated robot, the dynamics of each link is subject also to forces/torques due to:

- motion couplings with other links (inertial, centrifugal)
- its own motion simultaneous with that of other links (Coriolis)
- static loads (gravity, contact forces)

These “dynamic couplings” are completely neglected by a decentralized approach

Link 1 and 2 influence each other




The effects of these nonlinear couplings and loads can be partially “masked” in the dynamic behavior of a joint axis/motor load if transmissions with high reduction ratios (n ≥ 100) are used

What To do?

... I just crank up The gains...

- ⊖ unmodeled flexibiliTies can make The robot unsTable
- ⊖ The robot becomes very rigid

consider joint interzctions and compensate for them

# CENTRALIZED CONTROL

- The controller when computing input Torque for a joint takes into account the behaviour of the other joints
- requires a mathematical model of the manipulator
- Idea: compensate the non-linearities to cancel the coupled dynamics and achieve a linear system with low gains (Feed back linearization idea)
- If, to compute the compensation, we employ:

  desired variables ⇒ inverse dynamics

  actual state variables ⇒ computed Torque

# 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 |
| | Traj. Tracking | feed back linearization (JSID) | feedback linearization (TSID) |
| motion in contact | | | Impedance / admittance control (with variants) |

# FEED-BACK LINEARIZATION

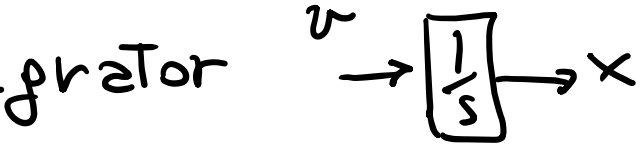
- FL is a methodology to control non-linear systems

$\dot{x} = \alpha(x)\, u + \beta(x)$

defining a control action as: $u = \alpha(x)^{-1}(v - \beta(x))$

I obtain a linearized system in the new control input $v$

$\dot{x} = v$ single integrator



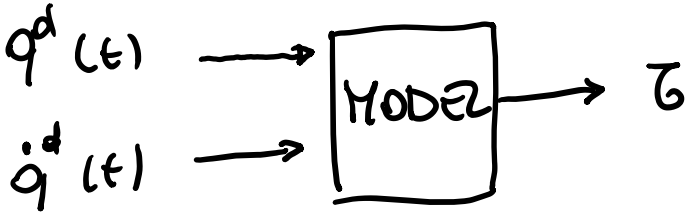

- If we have a model of the system we can compensate for non linearities
- The system becomes LINEAR and DECOUPLED w.r.t. any input of the system (e.g. $v_i$ influences only $x_i$)

⊕ we can set ANY stabylizing controller in $v$

⊖ perfect cancellation is not possible due to model inaccuracies

# INVERSE DYNAMICS COMPENSATION

reference Trajectory is used to compute the Torques That compensate coupling effects




given a Twice differentiable Trajectory for $t \in [0,T]$

$q^d(t)$ , $\dot{q}^d(t)$, $\ddot{q}^d(t)$

assuming perfect knowledge of the dynamics applying The feed-forward action:

$$u^d = M(q^d)\,\ddot{q}^d + h(q^d, \dot{q}^d)$$

if $q(0)^d = q(0)$ , $\dot{q}^d(0) = \dot{q}(0)$

leads To the exact reproduction of The desided motion but:

- Initial state not matched
- disturbances
- inaccurate model
- Un modelled dynamics

} divergence from desired Trajectory

The feed-back term to make the control scheme more robust

$$u = u^d + K_p(q^d - q) + K_p(\dot{q}^d - \dot{q})$$

↓ feed-forward compensates non-linearities

$$M(q)\ddot{q} + B(q,\dot{q}) = M(q^d)\ddot{q}^d + h(q^d,\dot{q}^d) + PD$$

The compensation is computed on $q^d, \dot{q}^d$ not on $q, \dot{q}$ ⇒ any Tracking error causes inexact cancellation

# COMPUTED TORQUE CONTROL

(A) Add a state feed-back that cancels non-linearities and achieves a closed loop exact linearization of the dynamics

step1: evaluate $M, R$ at the current state $q, \dot{q}$

step2: add an additional feed back

(1) $$u = \hat{M}(q)\, v + \hat{R}(q, \dot{q})$$

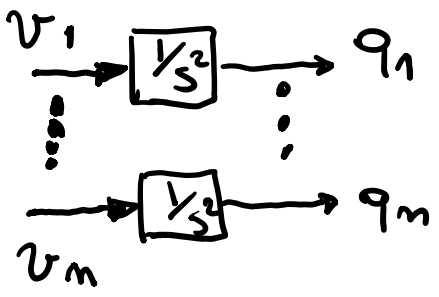
non-linear state feed-back

replacing in the nonlinear dynamics and assuming $\hat{M} = M$, $\hat{R} = R$:

$$M(q)\ddot{q} + \cancel{R} = M v + \cancel{R}$$

leads to:

(2) $$\ddot{q} = v$$




- pure linear system
- n decoupled double integrators with unitary mass

③ Syntesis of a linear control law To stabilize The Tracking error To zero

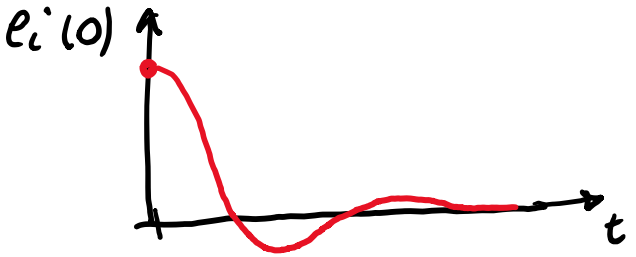
$$v = \ddot{q}^d + K_p (q^d - q) + K_d (\dot{q}^d - \dot{q})$$

replacing in $\ddot{q} = v$ we get The error dynamics:

$$(\ddot{q}^d - \ddot{q}) + K_p (q^d - q) + K_d (\dot{q}^d - \dot{q}) = 0$$

$$\ddot{e} + K_d \dot{e} + K_p e = 0$$

$\Rightarrow$ Tracking error is governed by a 2° order dynamics That can be arbitrarily assigned (on each joint) by suitably selecting gains of $K_p, K_d$




since The dynamics is:

- decoupled: each joint coordinate evolves independently from others, forced by $v_i$

$\Rightarrow$ $K_p, K_d$ are diagonal matrices

- linear: we have global asymptotic ($e_i \to 0$) stability for any $K_{Pi} > 0$, $K_{di} > 0$ (necessary / sufficient condition).

- The Time evolution is governed by the eigenvalues That are the roots of The polynomial:

$$s^2 + K_d s + K_p = 0 \quad \to \quad s_{1,2} = \frac{-K_d \pm \sqrt{K_d^2 - 4K_p}}{2}$$

if we have a PID:

$$\left(s^2 + K_d s + K_i \frac{1}{s}\right) e(s) = 0$$

$$\downarrow$$

$$\underbrace{\left(s^3 + K_d s^2 + K_p s + K_i\right)}_{P} e(s) = 0$$

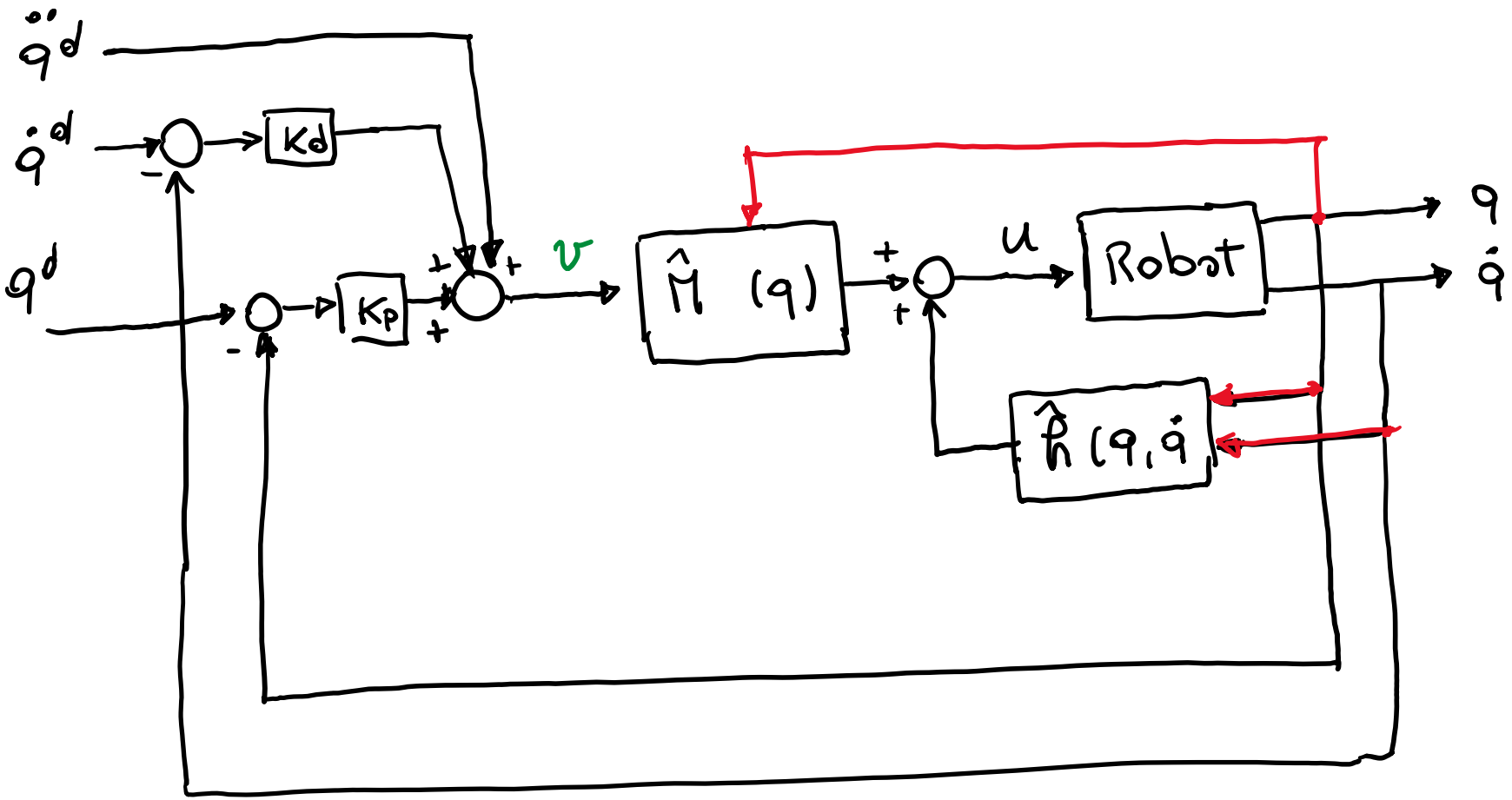
$\Rightarrow$ as. stable if The roots of characTenstic polynomial P have $Re < 0$

$\Rightarrow$ from RouTh criTerion

$K_{Di} > 0 \quad K_{Ii} > 0 \quad K_{Pi} > \frac{K_{Ii}}{K_{Di}}$

# BLOCK DIAGRAM

Computed torque control is a special case of the more general feedback-linearization methodology



# COMMENTS ON COMPUTED TORQUE

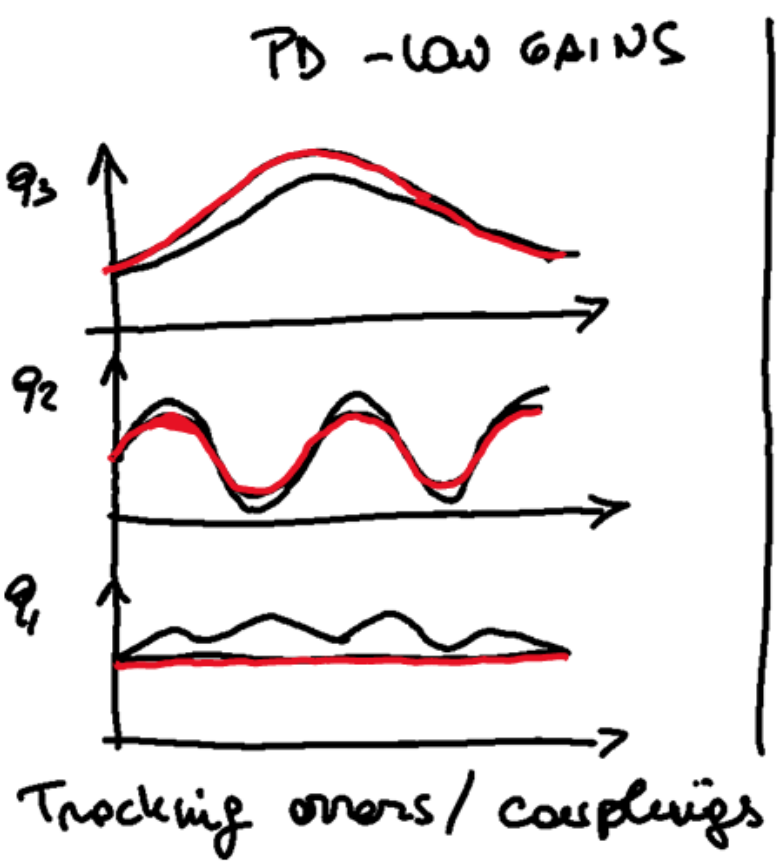
- compensation terms $\hat{M}(q)$, $\hat{h}(q,\dot{q})$ must be computed online with a small sampling time (0.5 - 10ms)
- we can use RNEA to compute them
- a perfect cancellation ($\hat{M} = M$, $\hat{h} = h$) is difficult in practice
- if we start from a cartesian trajectory ?

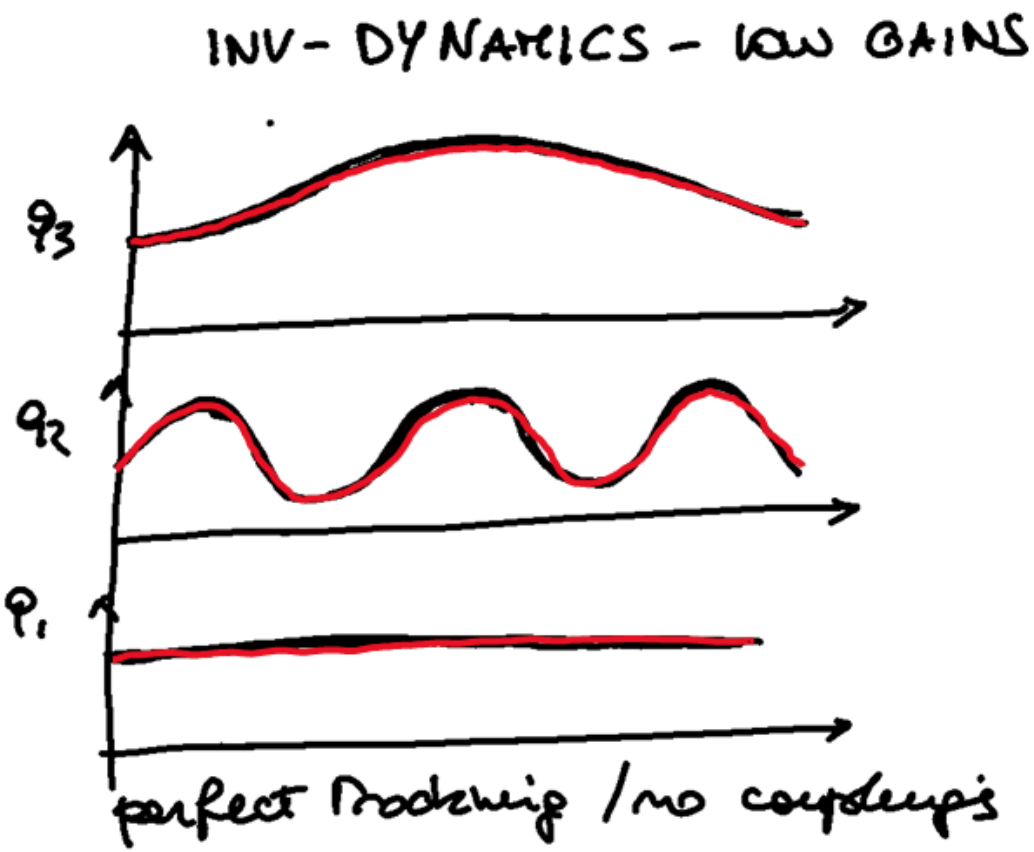
  $p^d(t)$, $\dot{p}^d(t)$, $\ddot{p}^d(t)$

  we can map it to joint space and still have a joint space controller:

$$\begin{cases} q^d(t) = f^{-1}(p^d(t)) \\ \dot{q}^d(t) = J^{-1}(q^d)\, p^d(t) \\ \ddot{q}^d(t) = J^{-1}(q^d)\left[\ddot{p}^d(t) - \dot{J}(q^d)\dot{q}^d\right] \end{cases}$$

- Kp and Kd can be Tuned considering The system DECENTRALIZED



# References


- Robotics Modelling, Planning and Control - Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.