Gravity learning for elastic joint robots

Robotics 2 Final Project

Master in Artificial Intelligence and Robotics

made and presented by:

Dennis Rotondi - 1834864
rotondi.1834864@studenti.uniroma1.it
Fabio Scaparro - 1834913
scaparro.1834913@studenti.uniroma1.it
Academic Year 2021-2022
Sapienza University of Rome

In this presentation . . .

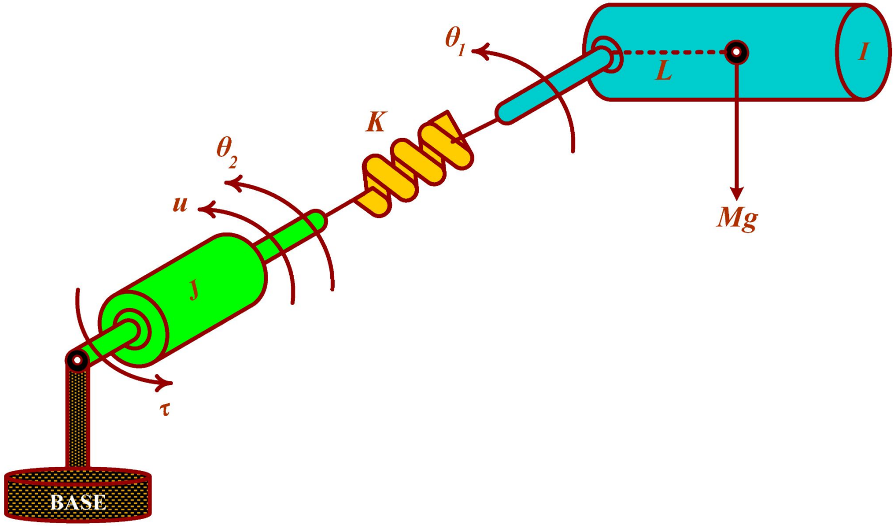
Dennis

1	Understand why regularize elastic robots is important	2	Elastic joint robots dynamic model
3	Why the classic PID fails	4	Iterative method with a sketch of proof
5	Implementation on Simulink	6	Simulation results

Fabio

Are elastic joint robots a thing?

YES!!

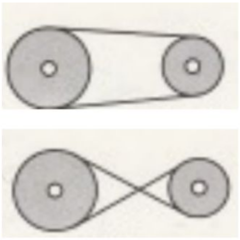
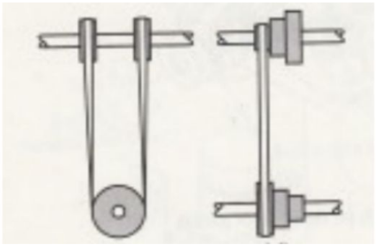


Joint flexibility is common in industrial robots

Motion transmission/reduction elements such as: belts, long shafts, cables, harmonic drives, or cycloidal gears, **intrinsically introduce this property**!

- allow to **relocate the actuators** next to the robot base, thus improve **power/dynamic efficiency**;
- are also **preferred for physical human-robot interaction**, since grant a decoupling between actuators and the lighter links, thus **reducing the kinetic energy involved in undesired collisions with humans**.




Dynamic Modeling

When **reduction gearings** are present, they are **modeled** as being placed **before the joint deflection** occurs. Standard assumptions and simplifications are made to ensure long life of electrical drives, linear elasticity, decoupling and independence properties.

A1 Joint deflections are small, so that flexibility effects are limited to the domain of linear elasticity.

A2 The actuators' rotors are modeled as uniform bodies having their center of mass on the rotation axis.

A3 Each motor is located on the robot arm in a position preceding the driven link. (This can be generalized to the case of multiple motors simultaneously driving multiple distal links.)

A4 The angular velocity of the rotors is due only to their own spinning, i. e.,

$$\boxed{{}^{R_i}\omega_{r_i} = \begin{pmatrix} 0 & 0 & \dot{\theta}_{m,i} \end{pmatrix}^{\top}}, \quad i = 1, \ldots, N,$$

Reduced Model

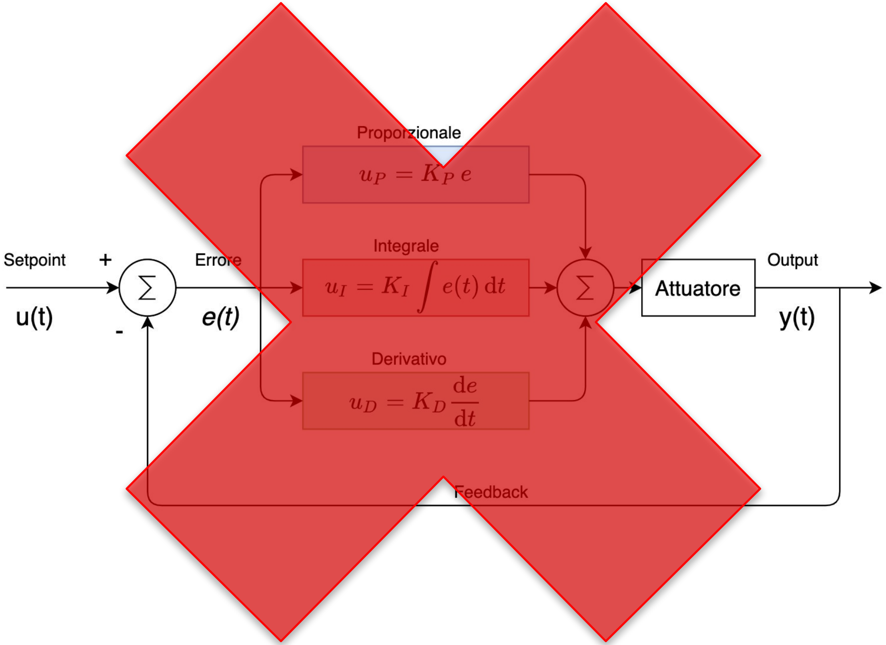
In contrast with the rigid robot, here there is a displacement between motor and link frames, **we need 2N variables**, one for each rigid body. **A1–A4** combined with Euler-Lagrange method lead to the model:

$$\begin{pmatrix} \boxed{\boldsymbol{M}(\boldsymbol{q})}\ddot{\boldsymbol{q}} + \boxed{\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})} + \boxed{\boldsymbol{K}}(\boldsymbol{q}-\boldsymbol{\theta}) \\ \boxed{\boldsymbol{B}}\ddot{\boldsymbol{\theta}} + \boxed{\boldsymbol{K}}(\boldsymbol{\theta}-\boldsymbol{q}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{u} \end{pmatrix} \tag{1}$$

where $\boldsymbol{B}$ is the diagonal inertia matrix of the rotors inertial components around their spinning axes, $\boldsymbol{M}(\boldsymbol{q})$ is the sum of the link inertia matrix $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$ which contains the rotor masses and the other rotor components, $\boldsymbol{K} > \boldsymbol{0}$ the diagonal matrix of joint stiffness, $\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}})$ collects the Coriolis and centrifugal terms and $\boldsymbol{g}(\boldsymbol{q})$ is the gravity.

Control is harder

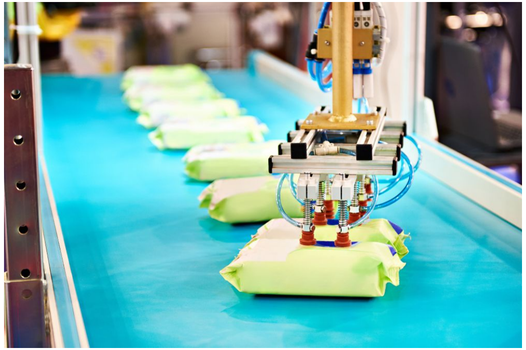
The price for agility and safety given by transmission elements is that we cannot rely on classic controls and therefore, in particular, it's harder to regularize flexible joint robots.



PID limitations

On several tasks there isn't exact knowledge of the gravity vector, hence we cannot use a simple PD law with constant gravity compensation. **Even PID**, the go-to for the rigid case:

- **has no formal proof of global convergence;**
- **is mathematically complex due to the nonlinear nature of the model;**
- **saturation will occur during large transient phases.**



Iterative learning method

This approach is based on the idea of using a PD control loop at motor level and two update rules for learning the correct compensation at the desired point. Under the assumptions a)--d) it assure global convergence to the desired configuration $\mathbf{q}_d$!

Control law:

$$\boldsymbol{u}(t) = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}(t)) - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} + \boldsymbol{u}_{i-1}$$

$$\boldsymbol{u}_i = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_i) + \boldsymbol{u}_{i-1}$$

$$\boldsymbol{\theta}_{d,i} = \boldsymbol{\theta}_i + (\boldsymbol{q}_d - \boldsymbol{q}_i)$$

Recalling:

$$\exists\alpha > 0 : \left\|\frac{\partial^2 U}{\partial \boldsymbol{q}^2}\right\| = \left\|\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}\right\| \leq \alpha, \forall \boldsymbol{q} \in R^N$$

Sufficient Assumptions:

a) $\lambda_{min}(\boldsymbol{K}) > \gamma\alpha$

b) $\lambda_{min}(\boldsymbol{K}_P) > \alpha$

c) $\gamma > 2$

d) $0 < \beta < \frac{\gamma-2}{2\gamma}$

Sketch of proof

a) $\lambda_{min}(\boldsymbol{K}) > \gamma\alpha$

b) $\lambda_{min}(\boldsymbol{K}_P) > \alpha$

c) $\gamma > 2$

d) $0 < \beta < \frac{\gamma-2}{2\gamma}$

$$\boldsymbol{u}(t) = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}(t)) - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} + \boldsymbol{u}_{i-1}$$

$$\boldsymbol{u}_i = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_i) + \boldsymbol{u}_{i-1}$$

$$\boldsymbol{\theta}_{d,i} = \boldsymbol{\theta}_i + (\boldsymbol{q}_d - \boldsymbol{q}_i)$$

$$\exists\alpha > 0 : \left\|\frac{\partial^2 U}{\partial \boldsymbol{q}^2}\right\| = \left\|\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}\right\| \le \alpha, \forall \boldsymbol{q} \in R^N$$

We first define $\boldsymbol{\delta}_i = \boldsymbol{q}_i - \boldsymbol{\theta}_i$, $\boldsymbol{e}_i = \boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_i$ then by proving that $\|\boldsymbol{e}_i\| \to 0$, $\|\Delta\boldsymbol{\theta}_{d,i}\| = \|\boldsymbol{\theta}_{d,i} - \boldsymbol{\theta}_{d,i-1}\| \to 0$ we can show $\boldsymbol{q}_d = \boldsymbol{q}_i$.

To this purpose we need to create a **contraction map** for both of them. This is possible considering that **with a constant gravity approximation we end up in a steady state**, then manipulating the **differences between quantities at consequent iteration** and involving the **structural property** of the gravity term we can properly upper bound, assumptions guarantee that the inequalities do not change sign and that they are enforced.

$$\boldsymbol{g}(\boldsymbol{q}_i) = -\boldsymbol{K}\boldsymbol{\delta}_i = \boldsymbol{u}_i = \frac{\boldsymbol{K}_P}{\beta}(\boldsymbol{\theta}_{d,i} - \boldsymbol{\theta}_i) + \boldsymbol{u}_{i-1}$$

$$\|\boldsymbol{u}_i - \boldsymbol{u}_{i-1}\| = \|\boldsymbol{g}(\boldsymbol{q}_i) - \boldsymbol{g}(\boldsymbol{q}_{i-1})\| \le \alpha\|\boldsymbol{q}_i - \boldsymbol{q}_{i-1}\| \le \alpha(\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\| + \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_{i-1}\|)$$

$$\|\boldsymbol{\delta}_i - \boldsymbol{\delta}_{i-1}\| \le \|\boldsymbol{K}^{-1}\|\|\boldsymbol{g}(\boldsymbol{q}_i) - \boldsymbol{g}(\boldsymbol{q}_{i-1})\| < \frac{\alpha}{\gamma\alpha}\|\boldsymbol{q}_i - \boldsymbol{q}_{i-1}\| \le \frac{1}{\gamma}(\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\| + \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_{i-1}\|)$$

$$\frac{1}{\beta}\|\boldsymbol{K}_P\boldsymbol{e}_i\| < \frac{\gamma\alpha}{\gamma-1}(\|\boldsymbol{e}_i\| + \|\boldsymbol{e}_{i-1}\| + \|\Delta\boldsymbol{\theta}_{d,i-1}\|) \longrightarrow \|\boldsymbol{e}_i\| < \frac{\gamma\beta}{\gamma - 1 - \gamma\beta}(\|\boldsymbol{e}_{i-1}\| + \|\Delta\boldsymbol{\theta}_{d,i-1}\|)$$

Sketch of proof

a) $\lambda_{min}(\boldsymbol{K}) > \gamma\alpha$

b) $\lambda_{min}(\boldsymbol{K}_P) > \alpha$

c) $\gamma > 2$

d) $0 < \beta < \frac{\gamma-2}{2\gamma}$

$$\boldsymbol{u}(t) = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}(t)) - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} + \boldsymbol{u}_{i-1}$$

$$\boldsymbol{u}_i = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_i) + \boldsymbol{u}_{i-1}$$

$$\boldsymbol{\theta}_{d,i} = \boldsymbol{\theta}_i + (\boldsymbol{q}_d - \boldsymbol{q}_i)$$

$$\exists\alpha > 0 : \left\|\frac{\partial^2 U}{\partial \boldsymbol{q}^2}\right\| = \left\|\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}\right\| \leq \alpha, \forall \boldsymbol{q} \in R^N$$

We first define $\boldsymbol{\delta}_i = \boldsymbol{q}_i - \boldsymbol{\theta}_i$, $\boldsymbol{e}_i = \boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_i$ then by proving that $\|\boldsymbol{e}_i\| \rightarrow 0$, $\|\Delta\boldsymbol{\theta}_{d,i}\| = \|\boldsymbol{\theta}_{d,i} - \boldsymbol{\theta}_{d,i-1}\| \rightarrow 0$ we can show $\boldsymbol{q}_d = \boldsymbol{q}_i$.

To this purpose we need to create a **contraction map** for both of them.

$$\begin{pmatrix} \|\boldsymbol{e}_i\| \\ \|\Delta\boldsymbol{\theta}_{d,i}\| \end{pmatrix} < \frac{1}{\gamma - 1 - \gamma\beta}\begin{pmatrix} \gamma\beta & \gamma\beta \\ 1 & 1 \end{pmatrix}\begin{pmatrix} \|\boldsymbol{e}_{i-1}\| \\ \|\Delta\boldsymbol{\theta}_{d,i-1}\| \end{pmatrix}$$

this is the mapping we are looking for since thanks to **d)** the matrix has eigenvalues strictly inside the unit circle.

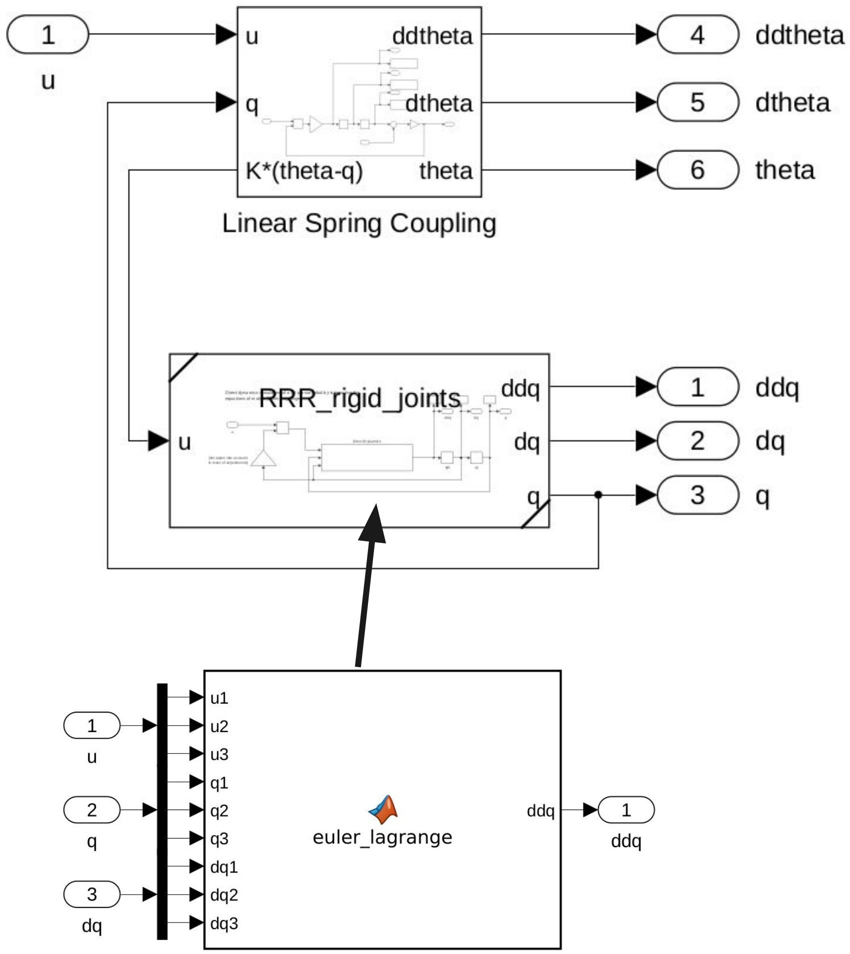
Then from $\boldsymbol{\theta}_{d,i} = \boldsymbol{\theta}_i + (\boldsymbol{q}_d - \boldsymbol{q}_i)$ we can easily derive the thesis.

Implementation on Simulink

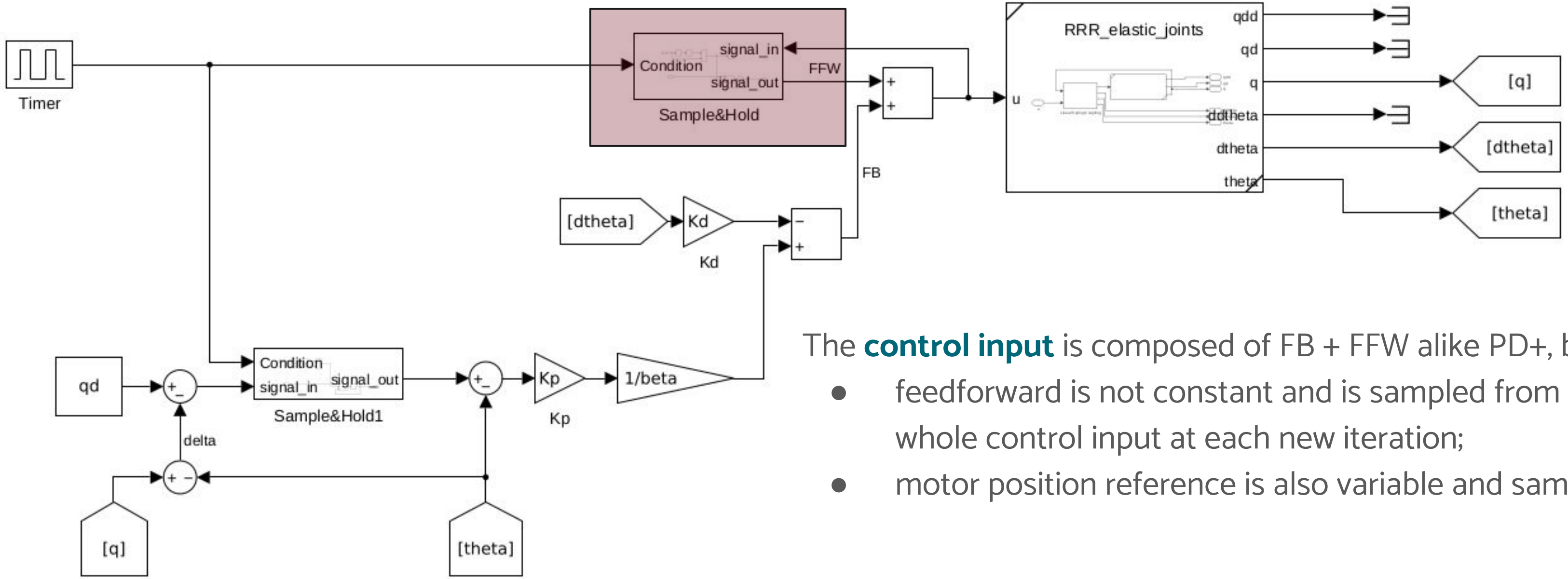
Forward dynamics via Matlab Function Block:

- define symbolic expression ddq = ...;
- compile a Matlab Function Block ddq = directDynamics(q, dq, u).

Additional block which exchanges elastic forces between state variables.



Iterative control law

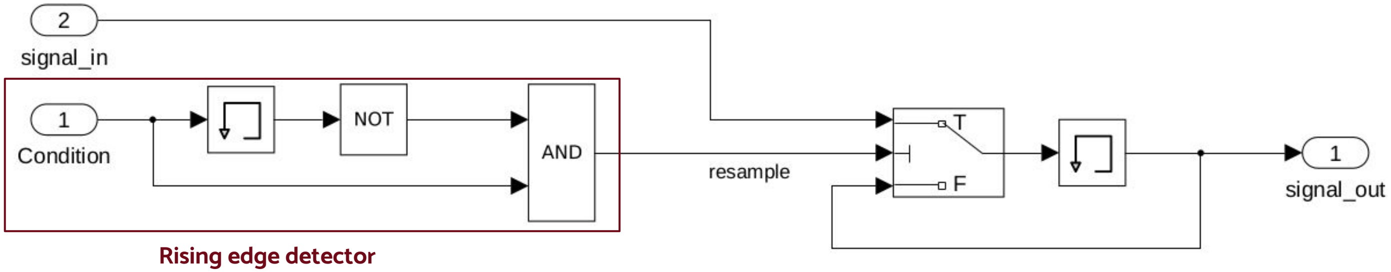


The **control input** is composed of FB + FFW alike PD+, but:

- feedforward is not constant and is sampled from the whole control input at each new iteration;
- motor position reference is also variable and sampled;

Sample and Hold

- To update FFW term and reference value, we make use of a **Sample&Hold** block, which outputs a sampled `signal_in`, resampling it when `condition` has a rising edge.



- **Ideally** sampling happens when the system is at ~steady state (velocities close to zero), so `condition` would be some steady state detection mechanism (for example a check on velocity norm).
  In our case it is **hard to dampen link velocities** due to our modeling choices and the check would never trigger. We iterate at periodic intervals instead. **Thus `condition` is just a clock**.

Simulations

Links have the same length 1 [m]
Uniformly distributed masses:

- m1 = 10 [kg]
- m2 = 7.5 [kg]
- m3 = 5 [kg]

⇩

$$\alpha \approx 340$$

at q = [-pi/2; 0; 0]
(estimated through sampling random configurations)

- Three different stiffness configurations to test the limits of the iterative approach:
  - **HIGH** stiffness
  - **MEDIUM** stiffness
  - **LOW** stiffness

- We also test the hypotheses of the proof to demonstrate that in some cases we can attain convergence despite having low gains / soft joints.

High stiffness

K = diag{14210, 29800, 13500} [Nm rad^-1]

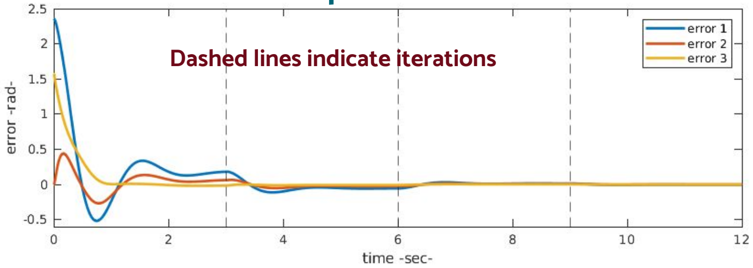
Kp = diag{600, 500, 400}   **Kd** = diag{200, 200, 200}

γ = 35   β = 0.47

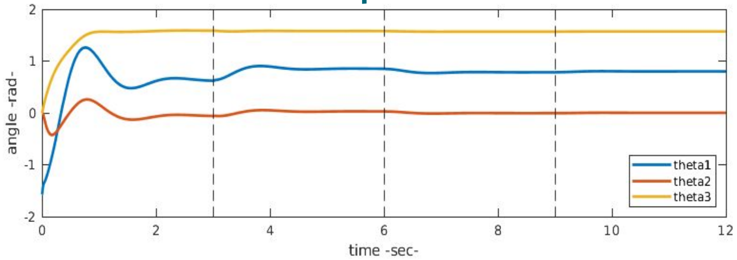
- Very rigid joints produce a behavior that is very similar to the rigid case
- No proof's hypotheses are violated

$$q_0 = (-\pi/2, 0, 0)^T \rightarrow q_d = (\pi/4, 0, \pi/2)^T$$

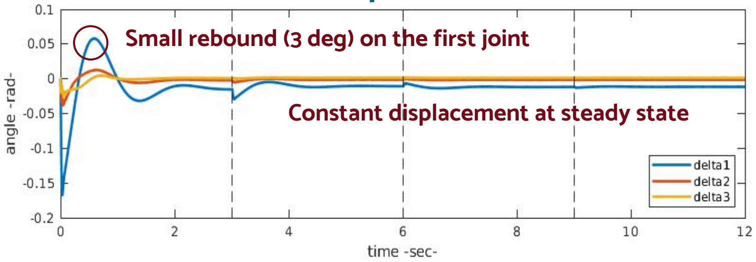
Link position error



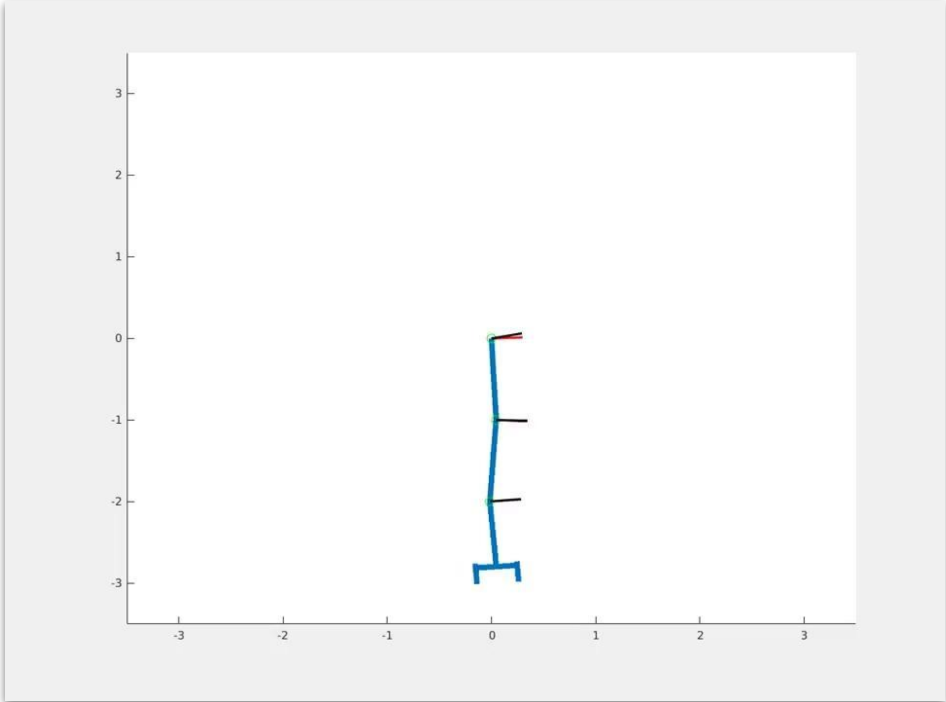
Motor positions



Link displacements



High stiffness – video



Red marker indicates link positions

Black marker indicates motor positions

Medium stiffness

K = diag{2100, 4500, 1500} [Nm rad^-1]

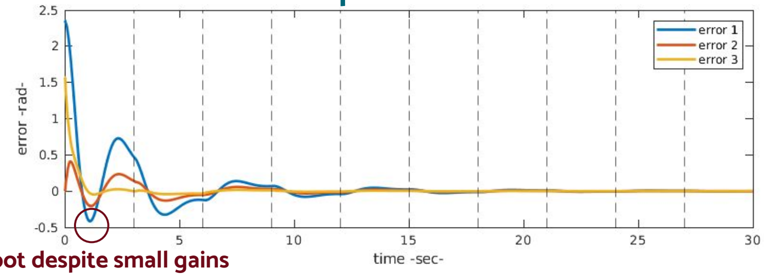
Kp = diag{200, 200, 200}   **Kd** = diag{150, 150, 150}

γ = 5   β = 0.3

- Predictably slower convergence
- Violated hypotheses:
  - **Kp < α**
  - **K3 < γα**

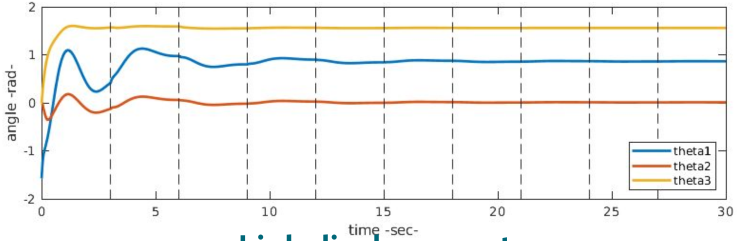
$$q_0 = (-\pi/2, 0, 0)^T \rightarrow q_d = (\pi/4, 0, \pi/2)^T$$

Link position error

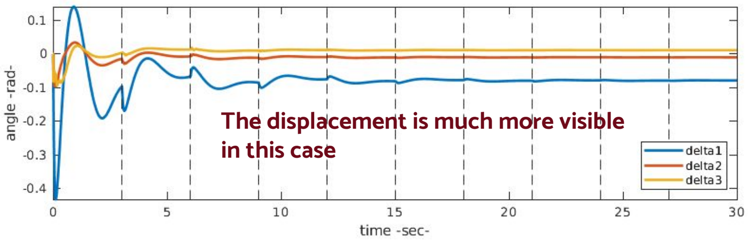


Big overshoot despite small gains

Motor positions



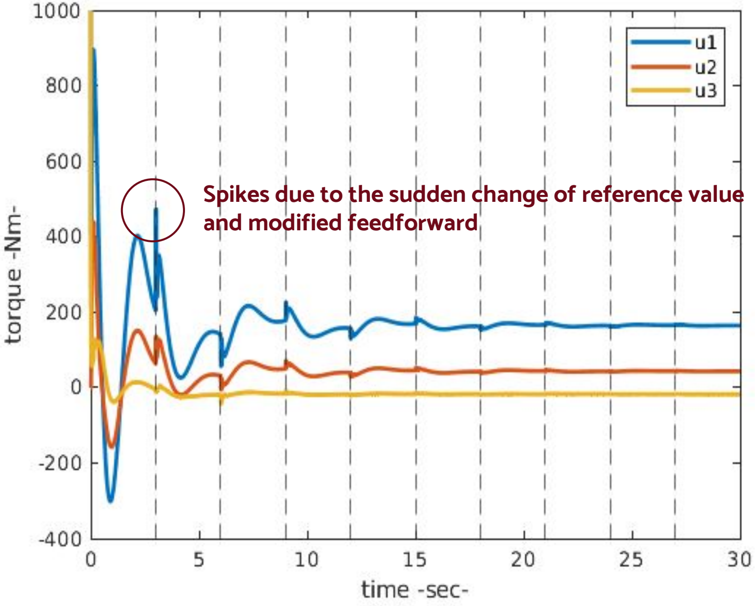
Link displacements



Medium stiffness

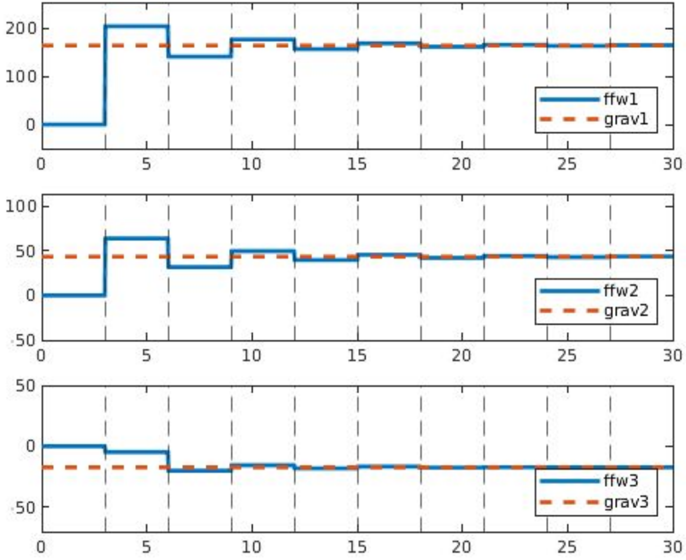
At steady state the robot balances gravity at the desired configuration, the control effort is just the feedforward term

Control effort



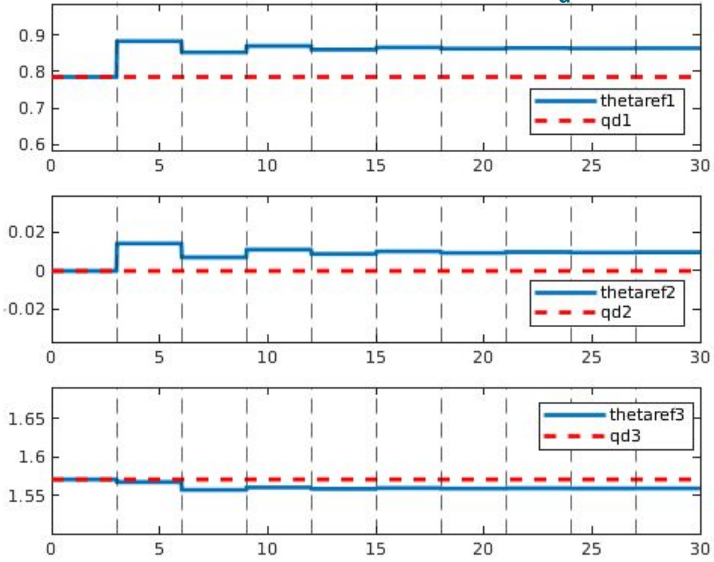
Spikes due to the sudden change of reference value and modified feedforward

Gravity compensation (ffw term)

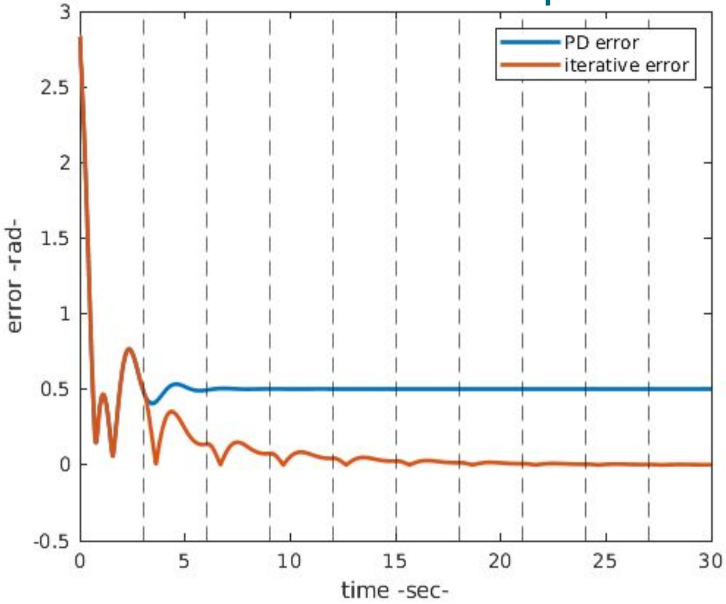


Medium stiffness

Motor position reference θ_d



Iterative and PD comparison



Identical behavior before the first iteration. Then, PD controller cannot recover the constant error while the iterative law learns to compensate for it

Low stiffness – Sim. 1

K = diag{1500, 1000, 500} [Nm rad^-1]

Kp = diag{7.5, 5, 0.75}

Kd = diag{2000,1000,800}

γ = 2.01

β = 1/400

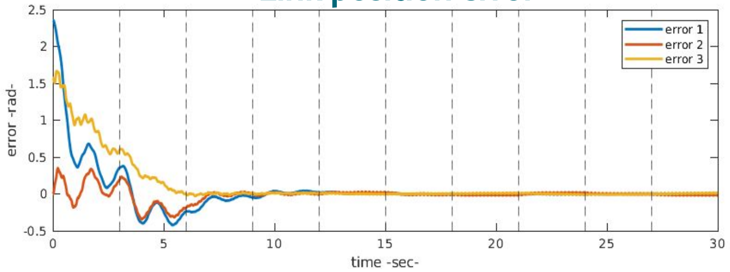
- Violated hypotheses:
  - Kp < α
  - K < γα

High derivative gains allow the motors to slowly move to their target, without oscillating

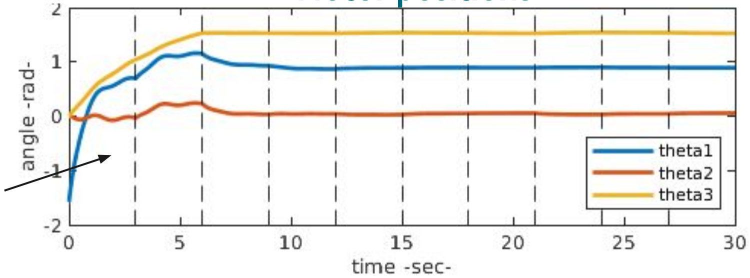
$$q_0 = (-\pi/2, 0, 0)^T \rightarrow q_d = (\pi/4, 0, \pi/2)^T$$

Convergence is NOT achieved: despite the fact that motors are still, the links keep oscillating at high frequencies

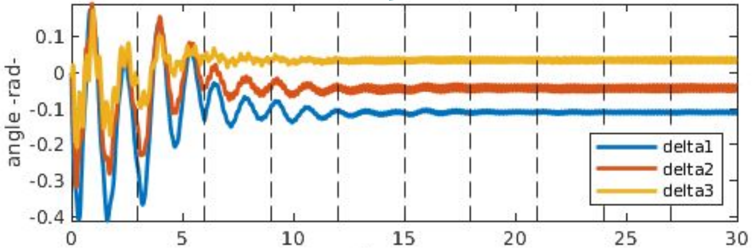
Link position error



Motor positions



Link displacements



Medium vs low stiffness comparison

$$q_0 = (-\pi/2, 0, 0)^T \to q_d = (\pi/4, 0, \pi/2)^T$$

Medium stiffness

Low stiffness

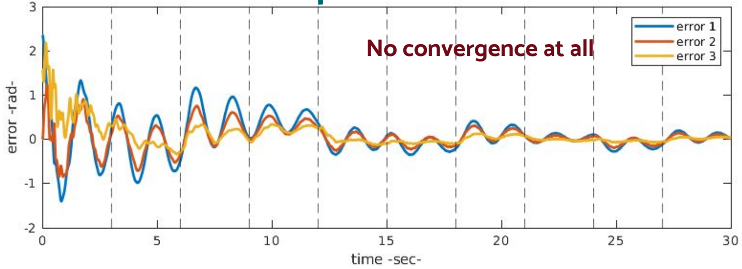
Low stiffness – Sim. 2

What happens if we lower the derivative gains?

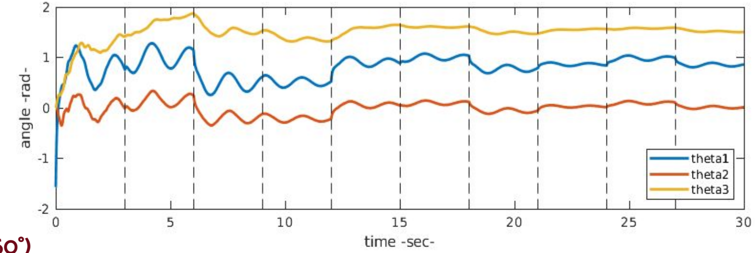
It **should** not matter, because there's no requirement for Kd in the proof, but we're already violating many of the hypotheses

With the values **Kd** = diag{200, 200, 200} we get nowhere near convergence

Link position error

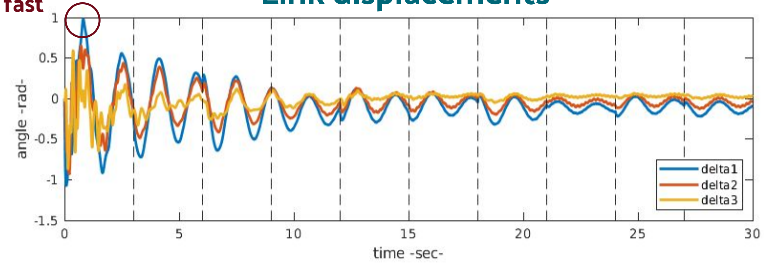


Motor positions



Very high displacements (around 60°) because the motors accelerate so fast

Link displacements



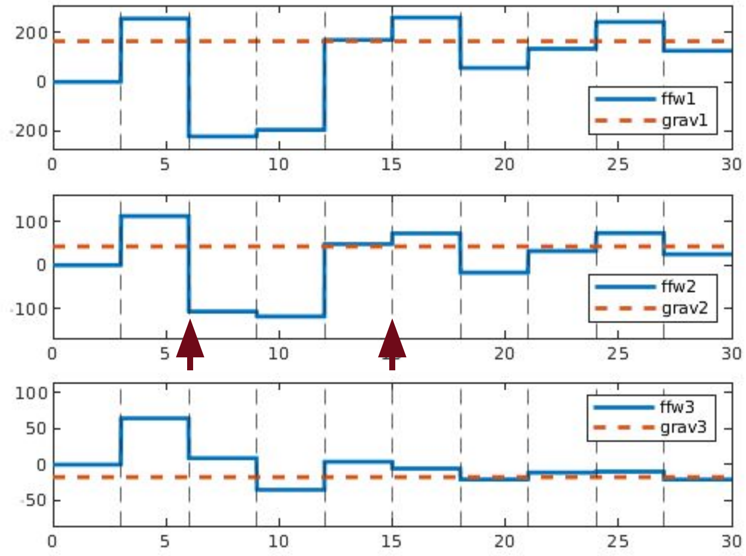
Low stiffness – Sim. 2

Why do we need such high derivative gains in low stiffness configurations?

- Avoid oscillations on motor positions which in turn make the links oscillate even more
- If we iterate when the robot is far from a steady state, we get very bad estimates for the updated variables

We could increase the iteration time, but that implies longer convergence times (>30s) which in practice is not useful

Gravity compensation on low stiffness configuration



"Bad" iterations can be highlighted especially 6 and 15 seconds into the simulation

Conclusions

Notwithstanding the notorious challenge in control elastic joint robots, by this extension of the rigid case we have accomplished astonishing results for the regulation task.

+ **Does not require knowledge about the robot and converges even with unknown payloads;**
+ **Globally Asymptotically stable (with easily satisfiable conditions);**
+ **Provides knowledge about the gravity term at the desired configuration;**
- **No indication about the number of iterations necessary for convergence (and consequentially for the convergence time);**
- **Requires measures also on the link side;**
- **Down to very soft robots need to be rethought.**

References

- **Papers:**

  - [1] P. Tomei. A simple PD controller for robots with elastic joints. IEEE Transactions on Automatic Control, 36(10):1208–1213, 1991.
    [2] A. De Luca and S. Panzieri. Learning Gravity Compensation in Robots: Rigid Arms, Elastic Joints, Flexible Links. Int. J. Adapt. Control Signal Process., 7(5):417–433, 1993.
    [3] A. De Luca and S. Panzieri. End-effector regulation of robots with elastic elements by an iterative scheme. International Journal of Adaptive Control and Signal Processing, 10(4-5):379–393, 1996.
    [4] B. Siciliano and O. Khatib. Springer Handbook of Robotics. Springer-Verlag, Berlin, Heidelberg, 2007.
    [5] A. De Luca and B. Siciliano. An Asymptotically Stable Joint PD Controller for Robot Arms with Flexible Links Under Gravity. In Proceedings of the 31st IEEE Conference on Decision and Control, pages 325–326 vol.1, 1992.

- **Slides:**
  - https://github.com/pietro-nardelli/sapienza-ppt-template (restyle of Sapienza NLP group slides)

Thank you for the attention!

made and presented by:

Dennis Rotondi - 1834864
rotondi.1834864@studenti.uniroma1.it
Fabio Scaparro - 1834913
scaparro.1834913@studenti.uniroma1.it
Academic Year 2021-2022
Sapienza University of Rome