SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER, CONTROL AND
MANAGEMENT ENGINEERING ANTONIO RUBERTI

Gravity learning for elastic joint robots

ROBOTICS 2 FINAL PROJECT

MASTER IN ARTIFICIAL INTELLIGENCE AND ROBOTICS

Professor:
Alessandro De Luca

Students:
Dennis Rotondi 1834864
Fabio Scaparro 1834913

Academic Year 2021/2022

Contents

Introduction **2**

1 Elastic Joint Robots: Dynamic Modeling **3**
1.1 Reduced Model 3
1.2 Complete Model 3

2 Elastic Joint Robots: Position Regulation **4**
2.1 PID limitations 4
2.2 Iterative Scheme 4
2.2.1 Proof of convergence 5

3 Implementation **7**
3.1 Simulink 7
3.2 Simulations 8
3.2.1 High stiffness 8
3.2.2 Medium stiffness 10
3.2.3 Low stiffness 10

Conclusions **14**

References **14**

Introduction

Properly regularize machine with flexible junctions (see Figure 2) is not a toy problem. As a matter of fact, the presence of elasticity at joint level is very common in industrial robots that heavily relies on motion transmission/reduction elements such as: belts, long shafts, cables, harmonic drives, or cycloidal gears Figure 1; aiming to relocate the actuators next to the robot base, thus improving power/dynamic efficiency. Flexible elements are also preferred for physical human–robot interaction, since they allow a decoupling between actuators and the lighter links. As always there is a trade-off: increasing the agility and safety of the robot we pay a cost in controlling it with a more complex and sophisticated law. Therefore many different approaches have been proposed [1, 2, 3] and in what follow we'll report our reasoning and simulations applying the learning one [3] adapted to our study case, underling the differences with respect to the rigid model studied during the course and giving a proof of convergence as well.

Figure 1: Some of the most common transmissions: 1) spur gears, 2) lead screws & worm gearing, 3) toothed belts & chains, 4) harmonic drives, 5) transmission shafts.

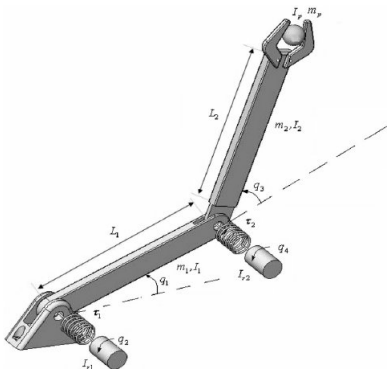

Figure 2: Example of a two-link flexible joint robot Manipulator.

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.,

$$^{Ri}\boldsymbol{\omega}_{ri} = \left(0 \;\; 0 \;\; \dot{\theta}_{m,i}\right)^{\top}, \quad i = 1, \ldots, N,$$

Figure 3: Standard assumptions made for modelling a robot with its elasticity.

1 Elastic Joint Robots: Dynamic Modeling

To accurately model the elastic joints some standard assumptions are needed, we'll base our dissertation on **A1, A2, A3** and the simplification **A4** (true engaging large reduction ratios), cleverly described in [4] *section 13.1.1* and summarized in Figure 3, so as to ensure long life of electrical drives, linear elasticity, decoupling and independence properties.

1.1 Reduced Model

In contrast with the rigid robot, here there is a displacement between motor and link frames, this requires to take into account $2N$ moving rigid bodies. Hence, a very convenient choice is the use of $2N$ generalized coordinates, N for the link $\boldsymbol{q}$ and N for the motor $\boldsymbol{\theta}$ positions. Paying attention to the fact that the rotors contribute to the kinetic energy and the joint deflections add an elastic component to the total potential energy, the usual Euler-Lagrange method can be applied yielding:

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

where $\boldsymbol{B}$ is the constant diagonal inertia matrix collecting 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, and $\boldsymbol{K} > \boldsymbol{0}$ the diagonal matrix of joint stiffness. Equations (1) are also known as reduced model.

1.2 Complete Model

Removing the simplification, we have to consider also the inertial couplings between the rotors and the previous links in the robot chain under the square matrix $\boldsymbol{S}(\boldsymbol{q})$, attaining the complete dynamic model:

$$\begin{pmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{S}(\boldsymbol{q}) \\ \boldsymbol{S}^T(\boldsymbol{q}) & \boldsymbol{B} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{\theta}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{c}_1(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{\theta}}) \\ \dot{\boldsymbol{c}}_2(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{pmatrix} + \begin{pmatrix} \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}(\boldsymbol{q}-\boldsymbol{\theta}) \\ \boldsymbol{K}(\boldsymbol{\theta}-\boldsymbol{q}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{u} \end{pmatrix} \tag{2}$$

moreover in presence of energy-dissipating effects, the non-conservative generalized forces appear on the right side of (2), for example viscous friction $\boldsymbol{F}_\theta$, $\boldsymbol{F}_q$ of the two sides of the transmissions and spring damping $\boldsymbol{D}$ of the joints (all matrices diagonal and positive definite) give rise to the term:

$$\begin{pmatrix} -\boldsymbol{F}_q\dot{\boldsymbol{q}} - \boldsymbol{D}(\dot{\boldsymbol{q}}-\dot{\boldsymbol{\theta}}) \\ -\boldsymbol{F}_\theta\dot{\boldsymbol{\theta}} - \boldsymbol{D}(\dot{\boldsymbol{\theta}}-\dot{\boldsymbol{q}}) \end{pmatrix} \tag{3}$$

Eventually, it's interesting to note that when $\boldsymbol{K} \to \infty$, then $\boldsymbol{\theta} \to \boldsymbol{q}$ while the elastic torque $\boldsymbol{\tau}_j = \boldsymbol{K}(\boldsymbol{\theta}-\boldsymbol{q}) \to \boldsymbol{u}$, making (2) collapse into the standard fully rigid model.

2 Elastic Joint Robots: Position Regulation

As premised, to reach the same performances of rigid case we need a bigger effort on control design. We'll focus our attention on the classical formulation (1): neglecting the dissipation effects that always help regularizing. In particular working on a 3R planar arm in the vertical plane, so under gravity ($\implies \boldsymbol{g}(\boldsymbol{q})$ not always zero).

2.1 PID limitations

As stated in [2], given an exact knowledge of the gravity vector, it's possible to asymptotically stabilize also this kind of robots using a proportional-derivative law [1]. However this is difficult to assume, for example in picking-up tasks where an on-line identification would be required. A standard approach is to use a PID, consisting of an integral term in addition to the linear PD law, for (unknown) gravity compensation. Unluckily there is no formal proof of global convergence, the result holds only locally around the desired configuration, it's mathematically complex due to the nonlinear nature of the robot and, commonly, saturation will occur during large transient phases.

2.2 Iterative Scheme

To achieve set-point regulation of the robot end-effector we have adopted the iterative one-stage scheme proposed in [3], which is based on the idea of using a simple PD control loop at motor level and two update rules for learning the correct compensation at the desired point:

$$\boldsymbol{u}(t) = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}(t)) - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} + \boldsymbol{u}_{i-1} \tag{4}$$

$$\boldsymbol{u}_i = \frac{1}{\beta}\boldsymbol{K}_P(\boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_i) + \boldsymbol{u}_{i-1} \tag{5}$$

$$\boldsymbol{\theta}_{d,i} = \boldsymbol{\theta}_i + (\boldsymbol{q}_d - \boldsymbol{q}_i) \tag{6}$$

where $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are positive definite gain matrices, the subscript i indicates the equilibrium reached at steady state of iteration i, $\boldsymbol{u}_{i-1}$ a constant FFW ($\boldsymbol{u}_0 = \boldsymbol{0}$ usually) and $\boldsymbol{\theta}_{d,i-1}$ the current estimate of the a priori unknown desired motor position, its most correct initialization is the desired tip position whether the arm were fully rigid. On the wake of [1, 4] that solve this problem provided exact gravity compensation, we are working with motor quantities; instead of adopting exclusively link variables that result in performance limit. In elastic joint robots not only the feed-forward term need an iterative update, but also the motor desired position, what reflect the rigid case is that $\boldsymbol{g}(\boldsymbol{q})$ contains only trigonometric and/or linear terms in $\boldsymbol{q}$, so the usual structural property holds:

$$\exists \alpha > 0 : \left\| \frac{\partial^2 U}{\partial \boldsymbol{q}^2} \right\| = \left\| \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} \right\| \leq \alpha, \forall \boldsymbol{q} \in R^N \tag{7}$$

The goal is to bring the robot to a desired configuration ($\boldsymbol{q} = \boldsymbol{q}_d$, $\dot{\boldsymbol{q}} = \boldsymbol{0}$, $\ddot{\boldsymbol{q}} = \boldsymbol{0}$) while acting on the motor variables $\boldsymbol{\theta}$. In absence of gravity, it is straightforward to say that $\boldsymbol{\theta}_d = \boldsymbol{q}_d$, because you want the motors and the links to be in the same position. Otherwise, in presence of gravity effects, it is necessary to account for an offset which generates an elastic force that balances the gravity. This can be easily understood by considering the single spring mass equation, where a mass oscillates along a reference position θ in the presence of gravity:

$$m\ddot{q} + k(q - \theta) = mg$$

At steady state, we have the configuration $q^* = \frac{mg}{k} + \theta$. If we want to reach q_d we need to balance the offset in the reference position, so that:

$$\theta_d = q_d - \frac{mg}{k}$$

In the iterative control law, we estimate both the gravity term $\boldsymbol{g}(\boldsymbol{q}_d)$ and $\boldsymbol{\theta}_d$ at the same time, without knowing a single detail about the robot. The downside of this approach is that we need another sensor for each joint, either to measure $\boldsymbol{q}$ or the displacement $\boldsymbol{\delta} = \boldsymbol{\theta} - \boldsymbol{q}$ directly. Note that in theory, after waiting infinite time needed for exact convergence, if you know $\boldsymbol{K}$ and $\boldsymbol{g}$ (which is the equal to the control effort read at the end of each iteration in our method) you could relate $\boldsymbol{\theta}$ and $\boldsymbol{q}$ at steady state from the first equation of (1): $\boldsymbol{\theta} = \boldsymbol{K}^{-1}\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{q}$.

2.2.1 Proof of convergence

In this subsection we will show how the link position, exploiting the control law (4) with updates (5) and (6), globally converges to $\boldsymbol{q} = \boldsymbol{q}_d$, provided that these four (sufficient) assumptions are satisfied:

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

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

c) $\gamma > 2$

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

Proof. Foremost we define the variables $\boldsymbol{\delta}_i$, displacement between motor and link at steady state, and $\boldsymbol{e}_i$, steady state error at motor level, as:

$$\boldsymbol{\delta}_i = \boldsymbol{q}_i - \boldsymbol{\theta}_i \tag{8}$$

$$\boldsymbol{e}_i = \boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_i \tag{9}$$

Let's start proving that:

$$\|\boldsymbol{e}_i\| \to 0 \tag{10}$$

$$\|\Delta\boldsymbol{\theta}_{d,i}\| = \|\boldsymbol{\theta}_{d,i} - \boldsymbol{\theta}_{d,i-1}\| \to 0 \tag{11}$$

At steady state, from the model equations (1) and (5) you have:

$$\boldsymbol{g}(\boldsymbol{q}_i) = -\boldsymbol{K}\boldsymbol{\delta}_i = \boldsymbol{u}_i = \frac{\boldsymbol{K}_P}{\beta}(\boldsymbol{\theta}_{d,i} - \boldsymbol{\theta}_i) + \boldsymbol{u}_{i-1} \tag{12}$$

From (12) with (7) and (8), exploiting the triangle inequality, follows that:

$$\|\boldsymbol{u}_i - \boldsymbol{u}_{i-1}\| = \|\boldsymbol{g}(\boldsymbol{q}_i) - \boldsymbol{g}(\boldsymbol{q}_{i-1})\| \leq \alpha\|\boldsymbol{q}_i - \boldsymbol{q}_{i-1}\| \leq \alpha(\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\| + \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_{i-1}\|) \tag{13}$$

Using (12), (13) and assumption a):

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

then

$$\|\boldsymbol{\delta}_i - \boldsymbol{\delta}_{i-1}\| < \frac{1}{\gamma - 1}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\| \tag{14}$$

Combining (13) and (14) recalling (9):

$$\|\boldsymbol{u}_i - \boldsymbol{u}_{i-1}\| < \frac{\gamma\alpha}{\gamma - 1}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\| = \frac{\gamma\alpha}{\gamma - 1}\| - \boldsymbol{e}_i + \boldsymbol{e}_{i-1} + \boldsymbol{\theta}_{d,i-1} - \boldsymbol{\theta}_{d,i-2}\|$$

noting that $\boldsymbol{u}_i - \boldsymbol{u}_{i-1} = \frac{\boldsymbol{K}_P}{\beta}\boldsymbol{e}_i$

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

which, together with assumption b), generates the following inequality for the error:

$$\|\boldsymbol{e}_i\| < \frac{\gamma\beta}{\gamma - 1 - \gamma\beta}(\|\boldsymbol{e}_{i-1}\| + \|\Delta\boldsymbol{\theta}_{d,i-1}\|) \tag{15}$$

To find a similar result for $\Delta\boldsymbol{\theta}_{d,i}$, thanks to (14) we show that:

$$\|\Delta\boldsymbol{\theta}_{d,i}\| = \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_{i-1}\| < \frac{1}{\gamma - 1}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\| \leq \frac{1}{\gamma - 1}(\|\boldsymbol{e}_i\| + \|\boldsymbol{e}_{i-1}\| + \|\Delta\boldsymbol{\theta}_{d,i-1}\|)$$

and involving (15)

$$\|\Delta\boldsymbol{\theta}_{d,i}\| < \frac{1}{\gamma - 1 - \gamma\beta}(\|\boldsymbol{e}_{i-1}\| + \|\Delta\boldsymbol{\theta}_{d,i-1}\|) \tag{16}$$

Combining (15) and (16) you get the set of inequalities:

$$\begin{pmatrix} \|\boldsymbol{e}_i\| \\ \|\Delta\boldsymbol{\theta}_{d,i}\| \end{pmatrix} < \frac{1}{\gamma - 1 - \gamma\beta}\begin{pmatrix} \gamma\beta & \gamma\beta \\ 1 & 1 \end{pmatrix}\begin{pmatrix} \|\boldsymbol{e}_{i-1}\| \\ \|\Delta\boldsymbol{\theta}_{d,i-1}\| \end{pmatrix} \tag{17}$$

It follows that $\|\boldsymbol{e}_i\| < \|\boldsymbol{e}_{i-1}\|$ and $\|\Delta\boldsymbol{\theta}_{d,i}\| < \|\Delta\boldsymbol{\theta}_{d,i-1}\|$, i.e. we have a contraction mapping, iff the matrix:

$$\frac{1}{\gamma - 1 - \gamma\beta}\begin{pmatrix} \gamma\beta & \gamma\beta \\ 1 & 1 \end{pmatrix}$$

has eigenvalues strictly inside the unit circle. One of them is obviously zero, the other one is $\frac{\gamma\beta+1}{\gamma-1-\gamma\beta}$, which is lower than 1 if assumption d) is valid. Assumption c) under the hood has allowed to keep the right sign of inequalities completing the proof for (10) and (11). At the limit $i \to \infty$, since (10) must be true:

$$\boldsymbol{\theta}_{d,i-1} = \boldsymbol{\theta}_i$$

that with (6) yields:

$$\boldsymbol{\theta}_{d,i} - \boldsymbol{\theta}_{d,i-1} = \boldsymbol{q}_d - \boldsymbol{q}_i$$

eventually for (11) in this situation:

$$\boldsymbol{q}_d = \boldsymbol{q}_i \tag{18}$$

Where $\boldsymbol{q}_i$ is the equilibrium reached at steady state after iteration i. □

The assumptions for this proof guarantee also that the steady state reached during each iteration is unique [2] and it, together with its existence, can be proved through a classical Lyapunov argument similarly to [1, 5], being each step of this iterative procedure interpreted as a PD with constant approximate gravity compensation FFW.

3 Implementation

3.1 Simulink

We will now discuss the technical details of the implementation used for simulating the robot. We used Simulink as our main simulator for this project. The 3R rigid robot has been implemented using a Matlab function block that computes the direct dynamics followed by two integrators. To turn this into an elastic joint robot, we put a block that takes care of the spring coupling between motor and link by exchanging elastic forces between $\boldsymbol{q}$ and $\boldsymbol{\theta}$. To implement the iterative control law, we crafted a "sample and hold block" Figure 4 which samples the input only when a certain boolean signal has a rising edge. This has been used to update the terms $\boldsymbol{u}_i$ and $\boldsymbol{\theta}_{d,i}$ in the control law. The update condition was first set to a threshold thr on $\dot{\boldsymbol{\theta}}$ to detect the steady state:

$$upd : \|\dot{\boldsymbol{\theta}}\| < thr = 10^{-3}$$

This worked quite well on the rigid robot but far less on the elastic joint one, as a matter of fact in most simulations this condition was never met, resulting in no iterations. This is caused by the high frequency oscillations in the model that never let the velocities cancel. This behavior arises from the fact that we did not model any kind of dissipation in the robot, particularly by including relative spring damping $\boldsymbol{D}\dot{\boldsymbol{\delta}}$ those oscillations would automatically dampen. Another possible solution is to conduct the update at certain time steps where you assume the robot had enough time

to reach an approximate steady state; this way, the quantities updated are estimated (i.e. $\boldsymbol{u}_i \neq \boldsymbol{g}(\boldsymbol{q}_i)$) but in practice we have found that despite this the robot reaches the desired configuration, the fastest results are attained when the conditions are in or.

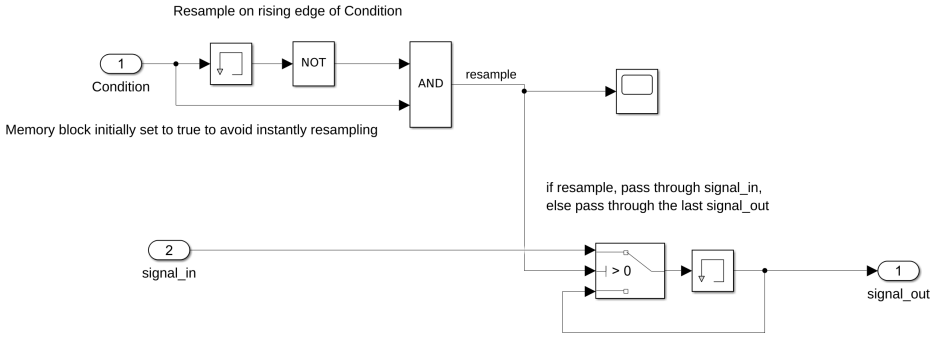

Figure 4: The "sample and hold block" used in our controllers, exploiting boolean logic it's possible to output the correct signal during each iteration.

3.2 Simulations

We've carried out some simulations testing the performance of the iterative controller with various cases of stiffness down to very soft joints, while keeping constant the other robot parameters: the links have equal length $l = 1$ [m], with uniformly distributed masses $m_1 = 10$, $m_2 = 7.5$, $m_3 = 5$ [kg]. We've also tried different gain matrices to understand how much the sufficient assumptions could be relaxed. Using a random sampling of the q-space we have been able to estimate $\alpha \approx 340$.

3.2.1 High stiffness

To validate our script we started using the huge levels of stiffness we found in [1]: $K_1 = 14210$, $= 29800$, $K_3 = 13500$ [Nm rad^{-1}], in this situation it is possible to play with γ and β in such a way that the gains are not very high while completely satisfying the convergence assumptions. Being into the bounds even when $\gamma = 35$, it's possible to set $\beta = 0.47$ and with $\boldsymbol{K}_P = \text{diag}\{600, 500, 400\}$, $\boldsymbol{K}_D = \text{diag}\{200, 200, 200\}$ we succeeded easily in all our trials. We observe it in our first simulation that consist in reaching $\boldsymbol{q}_d = (\pi/4, 0, \pi/2)^T$ from $\boldsymbol{q}_0 = (-\pi/2, 0, 0)^T$ at rest: Figures 5–7 show the errors and the applied torques (control effort) over 12 seconds, with a new iteration each 3 seconds only four are needed to converge. It is interesting to note how in Figure 6 there is nearly no displacement between $\boldsymbol{q}$ and $\boldsymbol{\theta}$, indeed as we said this is a feature of elastic joint robots and we are in a configuration where it's hard to spot the difference with the rigid case, in a certain sense we are using the same method validated for that kind of robots but with stronger assumption so one can expect to perform very well.

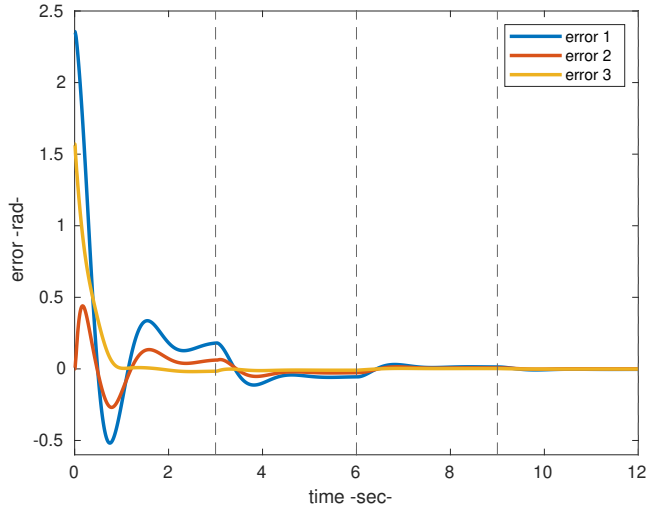

Figure 5: Simulation 1: joint level error.

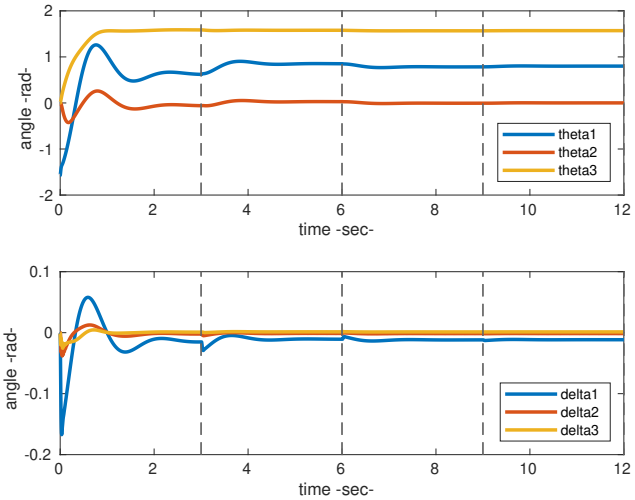

Figure 6: Simulation 1: θ and δ evolution.

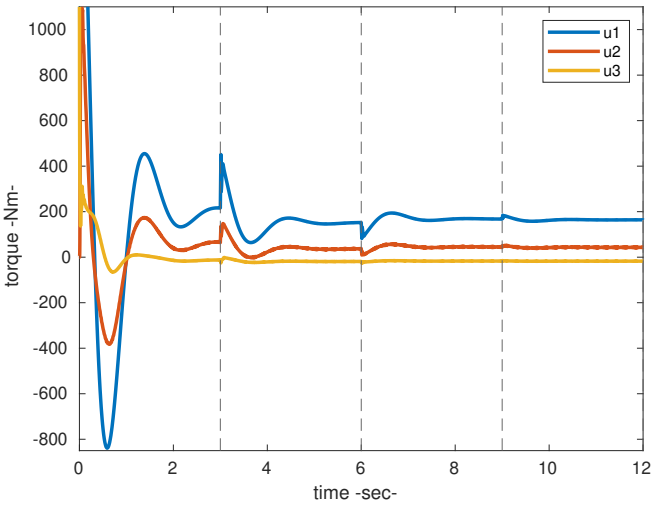

Figure 7: Simulation 1: applied torque.

3.2.2 Medium stiffness

After that, to get meaningful results we considerably scale down the stiffness to what we call "medium value" according to our investigations: $K_1 = 2100$, $K_2 = 4500$, $K_3 = 1500$ [Nm rad^{-1}]. Setting $\gamma = 5$, $\beta = 0.3$, we've configured the control with little gains $\boldsymbol{K}_P = \text{diag}\{200, 200, 200\}$, $\boldsymbol{K}_D = \text{diag}\{150, 150, 150\}$. Predictably the convergence will be slower, but it is not the only problem; we are not satisfying many assumption of the theorem: a) due to K_3 and b) due to the fact that each element of $\boldsymbol{K}_P$ is lower than α. Nevertheless, as it's possible to see in Figures 8–13 we are performing very well both on simulation 2 (which aim to reproduce sim. 1 with the new parameters) and simulation 3 (which brings the arm in $\boldsymbol{q}_d = (0, 0, 0)^T$ from the same initial conditions of sim. 2). This suggests for example that in c) β is not "strictly" constrained and, in general, that a)–d) are not necessary conditions. However, 3 seconds step iteration regardless of the robot' state, requires seven updates for Sim. 2 and five for sim. 3. In these cases is much easier to identify the displacement from Figures 9&12 since the elasticity is not negligible. In order to complete this analysis we show how in sim. 2 (very similar results apply to all the success case) the control term converges to the gravity one in the desired configuration Figure 14 and how slowly $\boldsymbol{\theta}_d$ departs from $\boldsymbol{q}_d$ as the time pass Figure 15. Lastly, if one is curious to verify which is the steady state error of the simple PD controller in Figure 16 we even compare its performance with the iterative method assuring to have the same $\boldsymbol{K}_P$, indeed we have identical behaviour before the first update.

3.2.3 Low stiffness

To test the limits of the iterative controller we further soften the stiffness to the values: $K_1 = 1500$, $K_2 = 1000$, $K_3 = 500$ [Nm rad^{-1}]. With these parameters we violate the conditions of convergence, event with the smallest possible value of γ which is 2. As a consequence we get a tiny value for $\beta = 1/400$, and since our proportional gain is divided by beta, we have to set it very low to avoid unwanted oscillations. In this simulation (from $\boldsymbol{q}_0 = (-\pi/2, 0, 0)^T$ at rest to $\boldsymbol{q}_d = (\pi/4, 0, \pi/2)^T$) we have set $\boldsymbol{K}_P = \text{diag}\{7.5, 5, 0.75\}$, $\boldsymbol{K}_D = \text{diag}\{2000, 1000, 800\}$. The small proportional gains nowhere near satisfy condition c) of the convergence proof. The high derivative gain values are justified by the fact that we want to excite as less oscillations as possible on motor positions while slowly moving them to $\boldsymbol{\theta}_d$, in order to keep link positions close to motor positions Figure 18. Here we do not have perfect convergence, but we get pretty close to it after four iterations with just some very small oscillations on link positions around $\boldsymbol{q}_d$ of amplitude 0.03 [rad] Figure 17. By lowering $\boldsymbol{K}_D$ to $\text{diag}\{200, 200, 200\}$ we do not get convergence at all, and a oscillatory behavior arises on $\boldsymbol{\theta}$ Figures 20–21. This also translates in poor joint torque profiles Figure 22.

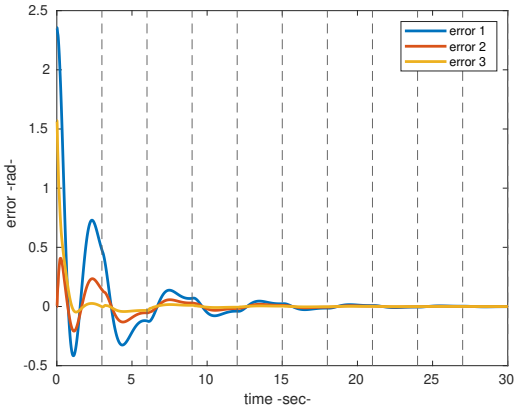

Figure 8: Sim. 2: joint level error.

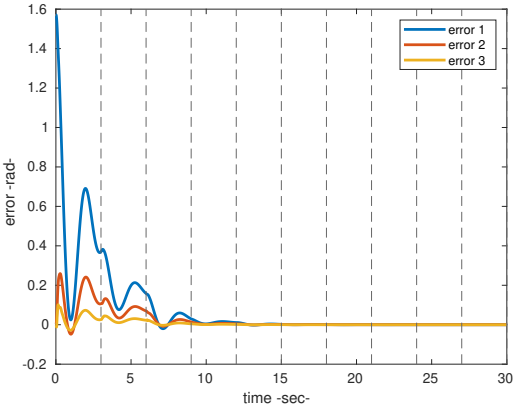

Figure 11: Sim. 3: joint level error.

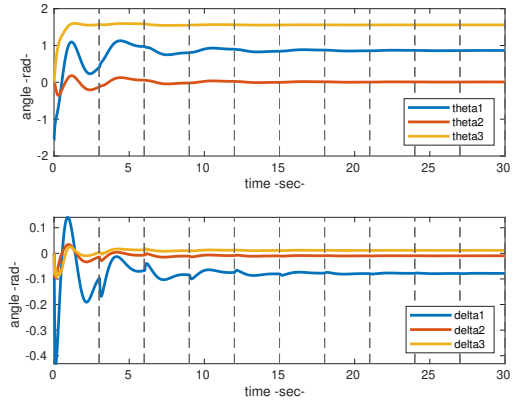

Figure 9: Sim. 2: $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ evolution.

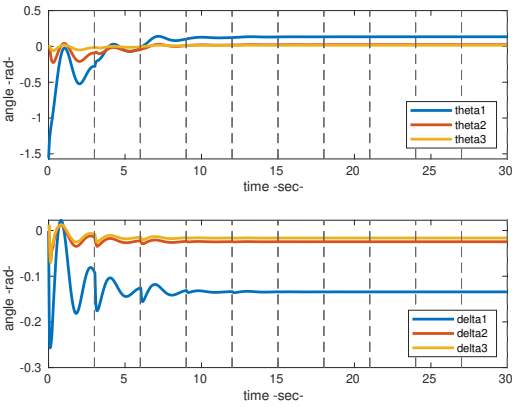

Figure 12: Sim. 3: $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ evolution.

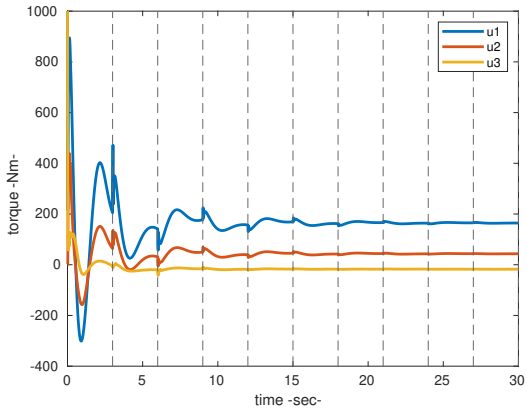

Figure 10: Sim. 2: applied torque.

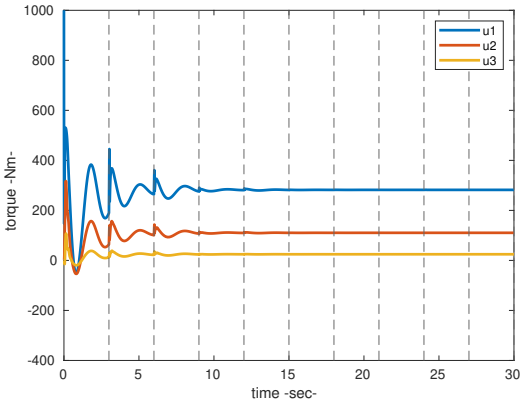

Figure 13: Sim. 3: applied torque.

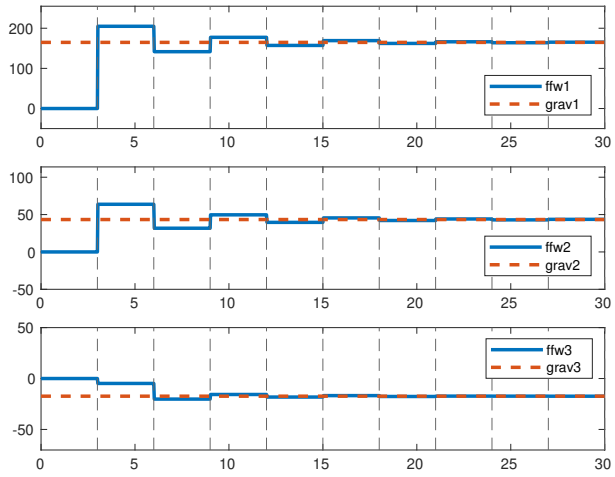

Figure 14: Simulation 2: evolution of $\boldsymbol{u}_{t-i}$ w.r.t. $\boldsymbol{g}(\boldsymbol{q}_d)$.

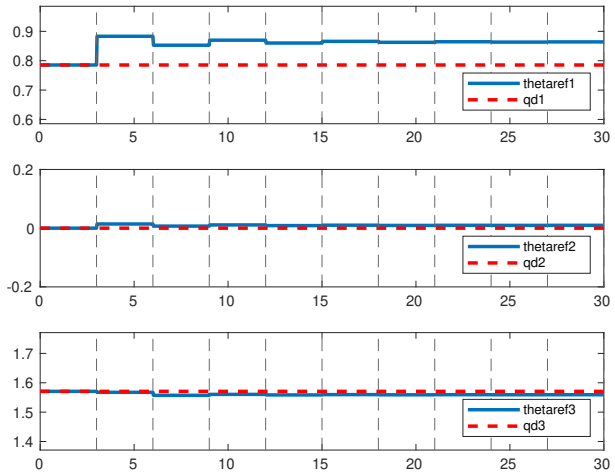

Figure 15: Simulation 2: evolution of $\boldsymbol{\theta}_d$ w.r.t. $\boldsymbol{q}_d$.

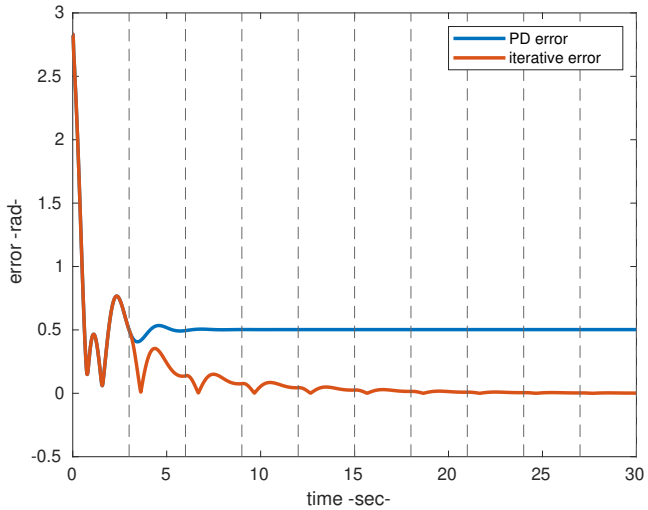

Figure 16: Simulation 2: PD vs Iterative Scheme.

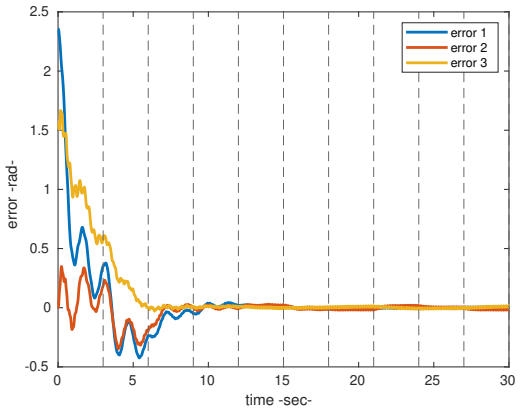

Figure 17: Sim. 4: joint level error.

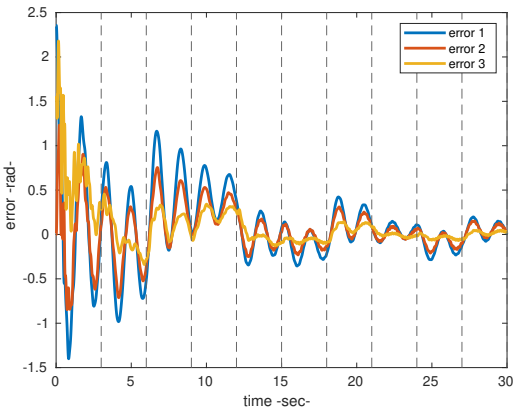

Figure 20: Sim. 5: joint level error.

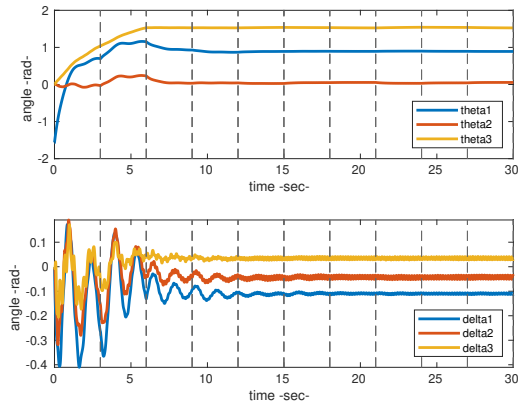

Figure 18: Sim. 4: $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ evolution.

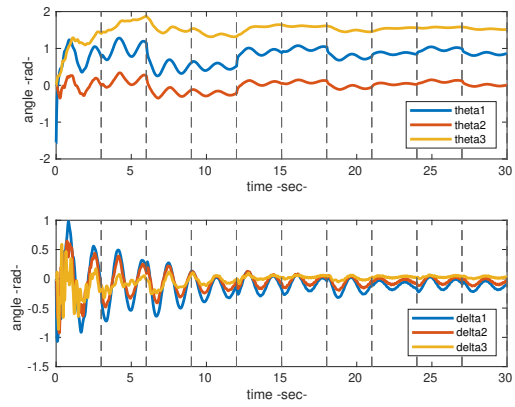

Figure 21: Sim. 5: $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ evolution.

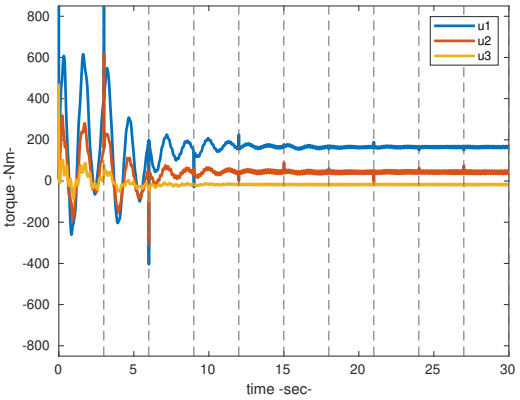

Figure 19: Sim. 4: applied torque.

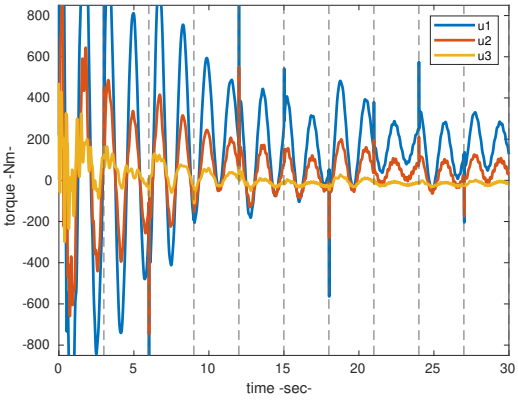

Figure 22: Sim. 5: applied torque.

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. In practice the sufficient conditions for convergence are very easy to satisfy, and even when severely violated the desired configuration can still be reached, hinting that there could be a further refinement. Unfortunately, down to very soft joints this approach needs to be rethought. Completing this project has been an educational journey through the whole pipeline of solving a control problem in robotics: starting from modelling the robot and ending up in simulating the environment for the trials. In these weeks we have been able to use many of the tools learned over the last year in Robotics 1 and Robotics 2, understanding how every single component of the control law affect the evolution of the state variables and having set strong fundamentals on which build upon for future projects.

References

[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.