Advanced Control Systems Report

Emanuele Feola, VR474837

February 2023

Contents

1 Homework 1

1.1 Assignment description

- Compute the DH Table
- Compute direct, inverse, differential kinematics

1.2 Denavit Hartenberg convention

Figure 1 shows the world reference frame.
Figure 2 shows the manipulator and its frames expressed according to the Denavit Hartenberg convention.
Table 1.2 shows the DH table of the manipulator. It has been cross-checked using the Robotics Toolbox by Peter Corke.

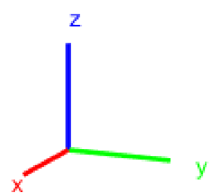

Figure 1: World reference frame

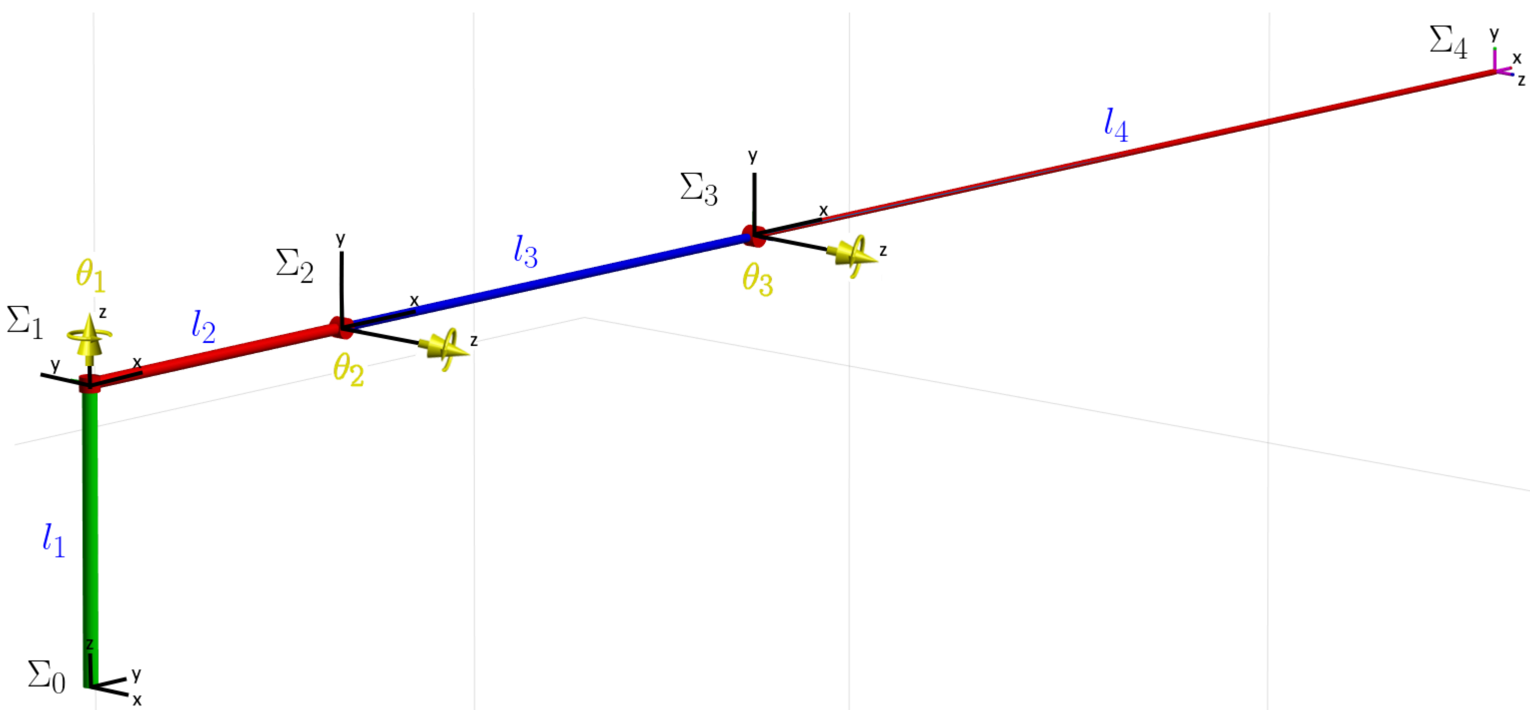

Figure 2: Manipulator's frames axis according to DH convention

Σ_i	a_i	α_i	d_i	θ_i
0-1	0	0	l_1	$\pi/2$
1-2	l_2	$\pi/2$	0	θ_1
2-3	l_3	0	0	θ_2
3-4	l_4	0	0	θ_3

1.3 Direct kinematics

The homogenous transformation matrices are computed for each row of the DH table. By multiplying $H_1^0 H_2^1 H_3^2 H_4^3$ we obtain the final transformation H_4^0.

$$H_1^0 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_2^1 = \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & l_2 \cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & l_2 \sin(\theta_1) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_3^2 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_3 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_3 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_4^3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_4 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & l_4 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_4^0 = \begin{pmatrix} -\cos(\theta_2+\theta_3)\sin(\theta_1) & \sin(\theta_2+\theta_3)\sin(\theta_1) & \cos(\theta_1) & -\sin(\theta_1)\left(l_2 + l_4 \cos(\theta_2+\theta_3) + l_3 \cos(\theta_2)\right) \\ \cos(\theta_2+\theta_3)\cos(\theta_1) & -\sin(\theta_2+\theta_3)\cos(\theta_1) & \sin(\theta_1) & \cos(\theta_1)\left(l_2 + l_4 \cos(\theta_2+\theta_3) + l_3 \cos(\theta_2)\right) \\ \sin(\theta_2+\theta_3) & \cos(\theta_2+\theta_3) & 0 & l_1 + l_4 \sin(\theta_2+\theta_3) + l_3 \sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1.4 Inverse kinematics

Performing summing and squaring of the displacement vector of H_4^0 leads to:

$$l_1{}^2 + 2\sin(\theta_2)\, l_1 l_3 + 2\sin(\theta_2+\theta_3)\, l_1 l_4 + l_2{}^2 + 2\cos(\theta_2)\, l_2 l_3 + 2\cos(\theta_2+\theta_3)\, l_2 l_4 + l_3{}^2 + 2\cos(\theta_3)\, l_3 l_4 + l_4{}^2$$

A few attempts have been made to solve for the joint angles θ_1, θ_2, θ_3, but without success.

1.5 Differential kinematics

1.5.1 Geometric jacobian

We can express the geometric jacobian $J(q)$ with respect to Σ_0 as a function of the joint configuration q. The jacobian has been evaluated with a few random joint configurations q and compared with the Robotics System Toolbox jacobian evaluated with the same joint configuration.

$$J(q) = \begin{pmatrix} -\cos(\theta_1)\left(l_2 + l_4 \cos(\theta_2+\theta_3) + l_3 \cos(\theta_2)\right) & \sin(\theta_1)\left(l_4 \sin(\theta_2+\theta_3) + l_3 \sin(\theta_2)\right) & l_4 \sin(\theta_2+\theta_3)\sin(\theta_1) \\ -\sin(\theta_1)\left(l_2 + l_4 \cos(\theta_2+\theta_3) + l_3 \cos(\theta_2)\right) & -\cos(\theta_1)\left(l_4 \sin(\theta_2+\theta_3) + l_3 \sin(\theta_2)\right) & -l_4 \sin(\theta_2+\theta_3)\cos(\theta_1) \\ 0 & l_4 \cos(\theta_2+\theta_3) + l_3 \cos(\theta_2) & l_4 \cos(\theta_2+\theta_3) \\ 0 & \cos(\theta_1) & \cos(\theta_1) \\ 0 & \sin(\theta_1) & \sin(\theta_1) \\ 1 & 0 & 0 \end{pmatrix}$$

1.5.2 Analytical jacobian

Considering the ZYZ set of euler angles and recalling the following relationships we can compute the analytical jacobian.

$$T(\phi_e) = \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\theta \\ 0 & c_\varphi & s_\varphi s_\theta \\ 1 & 0 & c_\theta \end{bmatrix}$$

$$T_A(\phi_e) = \begin{bmatrix} \mathbb{I}_3 & \emptyset_3 \\ \emptyset_3 & T(\phi_e) \end{bmatrix}$$
$$J(q) = T_A(\phi_e) J_A(q)$$

It follows the symbolic expression of the analytical jacobian $J_A(q)$ with respect to Σ_0 as a function of the joint configuration q as:

$$J_A(q) = \begin{pmatrix} -\cos(\theta_1)\ (l_2 + l_4\ \cos(\theta_2+\theta_3) + l_3\ \cos(\theta_2)) & \sin(\theta_1)\ (l_4\ \sin(\theta_2+\theta_3) + l_3\ \sin(\theta_2)) & l_4\ \sin(\theta_2+\theta_3)\ \sin(\theta_1) \\ -\sin(\theta_1)\ (l_2 + l_4\ \cos(\theta_2+\theta_3) + l_3\ \cos(\theta_2)) & -\cos(\theta_1)\ (l_4\ \sin(\theta_2+\theta_3) + l_3\ \sin(\theta_2)) & -l_4\ \sin(\theta_2+\theta_3)\ \cos(\theta_1) \\ 0 & l_4\ \cos(\theta_2+\theta_3) + l_3\ \cos(\theta_2) & l_4\ \cos(\theta_2+\theta_3) \\ 1 & -\frac{\cos(\phi-\theta_1)\ \cos(\theta)}{\sin(\theta)} & -\frac{\cos(\phi-\theta_1)\ \cos(\theta)}{\sin(\theta)} \\ 0 & -\sin(\phi-\theta_1) & -\sin(\phi-\theta_1) \\ 0 & \frac{\cos(\phi-\theta_1)}{\sin(\theta)} & \frac{\cos(\phi-\theta_1)}{\sin(\theta)} \end{pmatrix}$$

2 Homework 2

2.1 Assignment description

- Compute the kinetic energy
- Compute the potential energy

2.2 Preliminary computation

For every link the following properties have been computed:

- Center of mass
  - w.r.t Σ_i
  - w.r.t Σ_1
    * NB: w.r.t Σ_1 (not w.r.t Σ_0, i.e. not w.r.t base frame)
- Partial Jacobian
- Inertia tensor
  - w.r.t CoM_i
  - w.r.t Σ_i by adding the Steiner contribution
- Inertia matrix $B(q)$
  - given the following link properties: mass, inertia tensor, the partial jacobian and the rotation matrix R_i^0
  - $B(q)$ is positive definite
  - $B(q)$ is symmetric
- Kinetic energy
  - given B(q) and the symbolic variable $\dot{q}$
- Potential energy
  - given the link mass, the gravity acceleration vector g w.r.t. Σ_1 and the CoM w.r.t Σ_1

2.3 Links

The links have the following masses and lengths:

- link 1: 0.5 kg, 1 m
- link 2: 10 kg, 1.2 m
- link 3: 5 kg, 2 m
- link 4: 3.7 kg, 3.7 m

2.4 Centers of mass

The following vectors represent the position of the CoM of each link with respect to Σ_i.

$$p_{l_2}^2 = \begin{pmatrix} -\frac{l_2}{2} \\ 0 \\ 0 \end{pmatrix} p_{l_3}^3 = \begin{pmatrix} -\frac{l_3}{2} \\ 0 \\ 0 \end{pmatrix} p_{l_4}^4 = \begin{pmatrix} -\frac{l_4}{2} \\ 0 \\ 0 \end{pmatrix}$$

2.5 Potential energy

Given g the gravity acceleration vector expressed with respect to Σ_1, once the above data has been computed for each link, then the overall potential energy of the manipulator can be computed as the sum of the energy contributions of each link.

$$g = \begin{pmatrix} 0 \\ 0 \\ -9.81 \end{pmatrix}$$

It follows the symbolic formula of the potential energy $U(\theta)$ of the manipulator. The symbolic variables representing the lengths of the links have been replaced with their corresponding actual values, resulting in $U(\theta)$ being a function of the joint configuration.

$$U(\theta) = 148.8 * sin(\theta_2 + \theta_3) + 209.9 * sin(\theta_2)$$

2.5.1 Observations

- The potential energy of the manipulator does not depend on the first joint, i.e. there is no θ_1 in the above formula.
- It has been checked with matlab's function *fminsearch* that the potential energy assumes its maximum value when

$$q = \begin{pmatrix} 0 \\ \pi/2 \\ 0 \end{pmatrix}$$

It follows an image of the manipulator in the above mentioned joint configuration, i.e. the joint configuration in which the potential energy is at its maximum value.

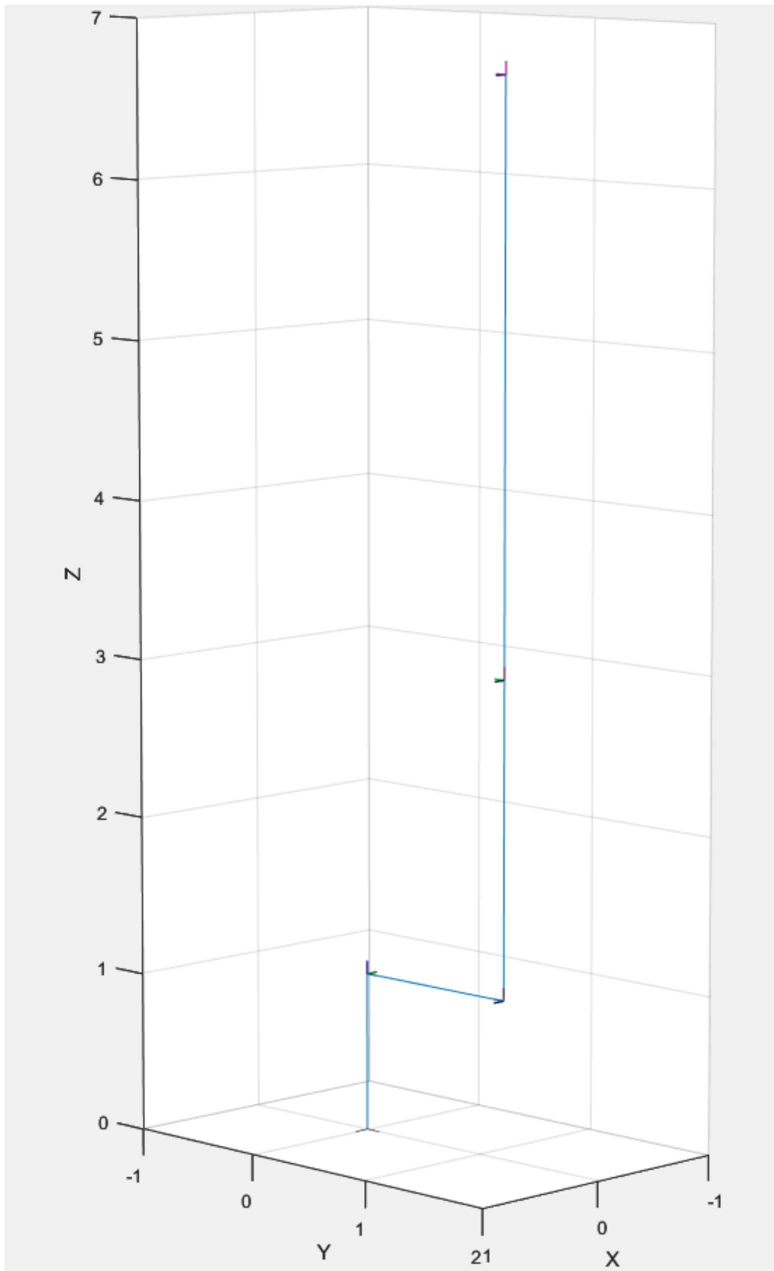

Figure 3: Manipulator's highest potential energy value configuration

2.6 Kinetic energy

Once the above data has been computed for each link, the overall kinetic energy of the manipulator is given by the sum of the energy contributions of each link. The symbolic formula of the kinetic energy is not reported here because it's too long and difficult to cross-check.

3 Homework 3

3.1 Assignment description

- Compute the equations of motion (dynamic model)

3.2 Lagrangian formulation

Given a desired joint configuration q, joint velocity vector $\dot{q}$ and joint acceleration vector $\ddot{q}$, the goal is to find the joint torque command necessary to drive the manipulator in the given configuration. In other words we are solving an inverse dynamics problem.
We are solving the inverse dynamics problem by computing the expression that represents the dynamic model of the manipulator:

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

The symbolic expression of the inertia matrix $B(q)$ was computed in the previous assignment.
The vector $g(q)$ should not be confused with the gravity vector shown before, since $g(q)$ is the vector of moments generated at the manipulator joints due to the gravity force acting on it.
The vector $g(q)$ can be computed by differentiating the previously computed matrix $U(\theta)$, i.e. the potential energy of the manipulator.
The 'Coriolis and centrifugal' matrix $C(q, \dot{q})$ is computed by choosing the Christoffel symbols of the first type and differentiating the elements of the matrix $B(q)$ with respect to the joint variables.

3.3 Gravity contributions considerations

The matrix of gravity contributions can be checked, at least on an intuitive level, by setting both the joint velocity vector $\dot{q}$ and joint acceleration vector $\ddot{q}$ to 0, while setting the joint configuration vector q to a convenient set of values.
Four different joint configurations and their corresponding joint moments due to the gravity forces are analyzed to provide an example.
The following figures show the manipulator in different configurations specifying for each pose the joint configuration in degrees.

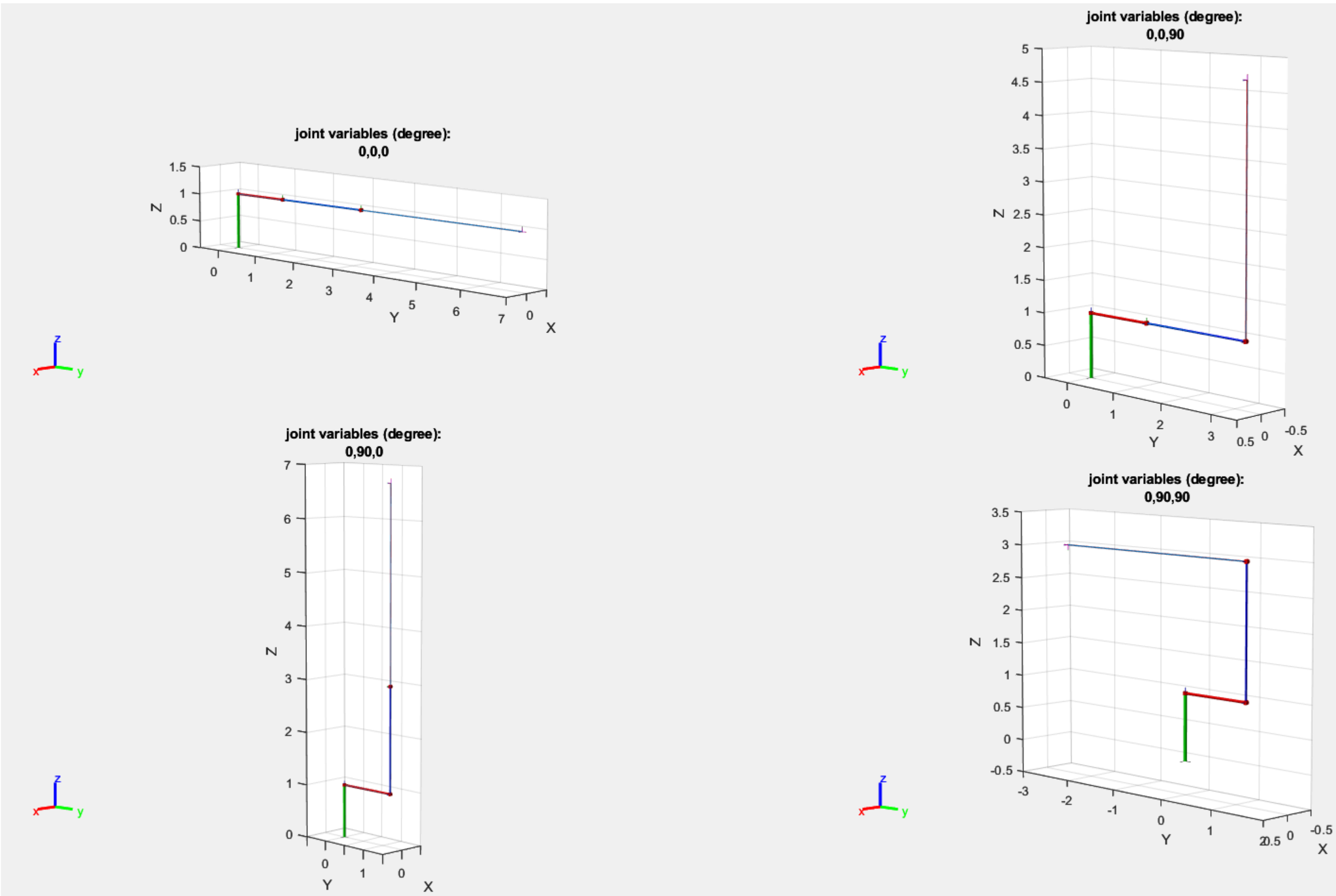

Figure 4: Manipulator in different joint configurations

3.3.1 Configuration 1 - home configuration $q = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$

The resulting $\tau = [\tau_1, \tau_2, \tau_3]^T$ is:

$$\tau = \begin{pmatrix} 0 \\ 358.7 \\ 148.8 \end{pmatrix}$$

Since the gravity force is parallel to the rotation axis of the first joint, the torque due to gravity on the first joint is 0.
The same consideration applies for the next joint configurations, i.e. the torque of the first joint will be 0 for the same reason in Configuration 2/3/4.
The torque on the second joint is higher than the third one, because the second joint has to compensate for both the weight of the second and third links.

3.3.2 Configuration 2: $q = \begin{pmatrix} 0 & 0 & \pi/2 \end{pmatrix}$

$\tau = [\tau_1, \tau_2, \tau_3]^T$ is:

$$\tau = \begin{pmatrix} 0 \\ 209.9 \\ 0 \end{pmatrix}$$

Even though the gravity force is perpendicular to the third joint's axis of rotation, the force arm is 0 and as a consequence the generated momentum on the third joint is also 0.
$\tau_2 = 358.7 - 148.8 = 209.9$, i.e. the difference between τ_2 and τ_3 of Configuration 1.

3.3.3 Configuration 3: $q = \begin{pmatrix} 0 & \pi/2 & 0 \end{pmatrix}$

$\tau = [\tau_1, \tau_2, \tau_3]^T$ is:

$$\tau = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

The same reasoning of Configuration 2 applies for the second joint in this configuration.
In this particular joint configuration the momentum due to gravity forces is null.

3.3.4 Configuration 4: $q = \begin{pmatrix} 0 & \pi/2 & \pi/2 \end{pmatrix}$

$\tau = [\tau_1, \tau_2, \tau_3]^T$ is:

$$\tau = \begin{pmatrix} 0 \\ -148.8 \\ -148.8 \end{pmatrix}$$

The generated torque is the same for the second and third joint.
This is due to the fact that both joints 'feel' only the weight of the last link.

4 Homework 4

4.1 Assignment description

- Compute the RNE formulation

4.2 Newton-Euler Inverse Dynamics

The goal is the same as the previous assignment, but this time we are solving the inverse dynamics problem using a recursive algorithm and we are not computing explicitly the matrices that appear in the lagrangian formulation of the manipulator.
Given the same input $\begin{pmatrix} q & \dot{q} & \ddot{q} \end{pmatrix}$ the output torque τ should be the same as the one computed in the previous assignment.
By executing the RNE algorithm with certain input vectors we can also compute the same matrices we found in the previous assignment, i.e. $B(q)$, $C(q, \dot{q})$ and $g(q)$.

4.3 Previous assignment comparison

The computed torques given as input $\begin{pmatrix} q & \dot{q} & \ddot{q} \end{pmatrix}$ have been compared with the previous assignment output.
The output is the same.
The comparison code can be found in the 'assignment_3_4.m' file.

5 Homework 5

5.1 Assignment description

- Compute the dynamic model in the operational space

5.2 Dynamic model in the operational space

In the previous assignments we considered the dynamic model of the manipulator as a function of the generalized coordinates q, $\dot{q}$.
The goal of this assignment is to rewrite the same dynamic model as a function of the cartesian coordinate

$$x = \begin{pmatrix} p \\ \phi \end{pmatrix}$$

of the end effector.
To do this, we use the relationships that bind the joint space to the cartesian space.
The resulting matrices are omitted since they are quite long and of difficult understanding.

6 Homework 6

6.1 Assignment description

- Design the Joint Space PD control law with gravity compensation
- What happens if $g(q)$ is not taken into account?
- What happens if the gravity term is set constant and equal to $g(q_d)$ within the control law?
- What happens if q_d is not constant (e.g. $q_d(t) = q_d + noise$)?

6.2 Joint space PD control law with gravity compensation

The manipulator started from its home position, i.e. $q = [0, 0, 0]$ and the target trajectory is a step function that starts at $t = 1s$.
The values of K_p and K_d have been chosen such that the influence of the gravity compensation can be appreciated by comparing the following images.
If higher K_p and K_d values are set, then we get a faster system response, that is what we usually want, but in this case to highlight the differences between compensated and non-compensated scenarios, the PD constants are set lower than usual and thus the system response is quite slow.
It can be seen in Figure 1 that if the gravity compensation is active, then the steady-state error is 0.

6.2.1 Considerations

We assume to know perfectly $g(q)$.
If $\hat{g}(q) \mathrel{!}= g(q)$, i.e. we are applying a gravity compensation that is different from the real one, then we have an offset in steady-state. In the case of a steady-state offset due to a mismatch between the real and applied gravity contributions matrix, we would need to add to the controller an integral action.

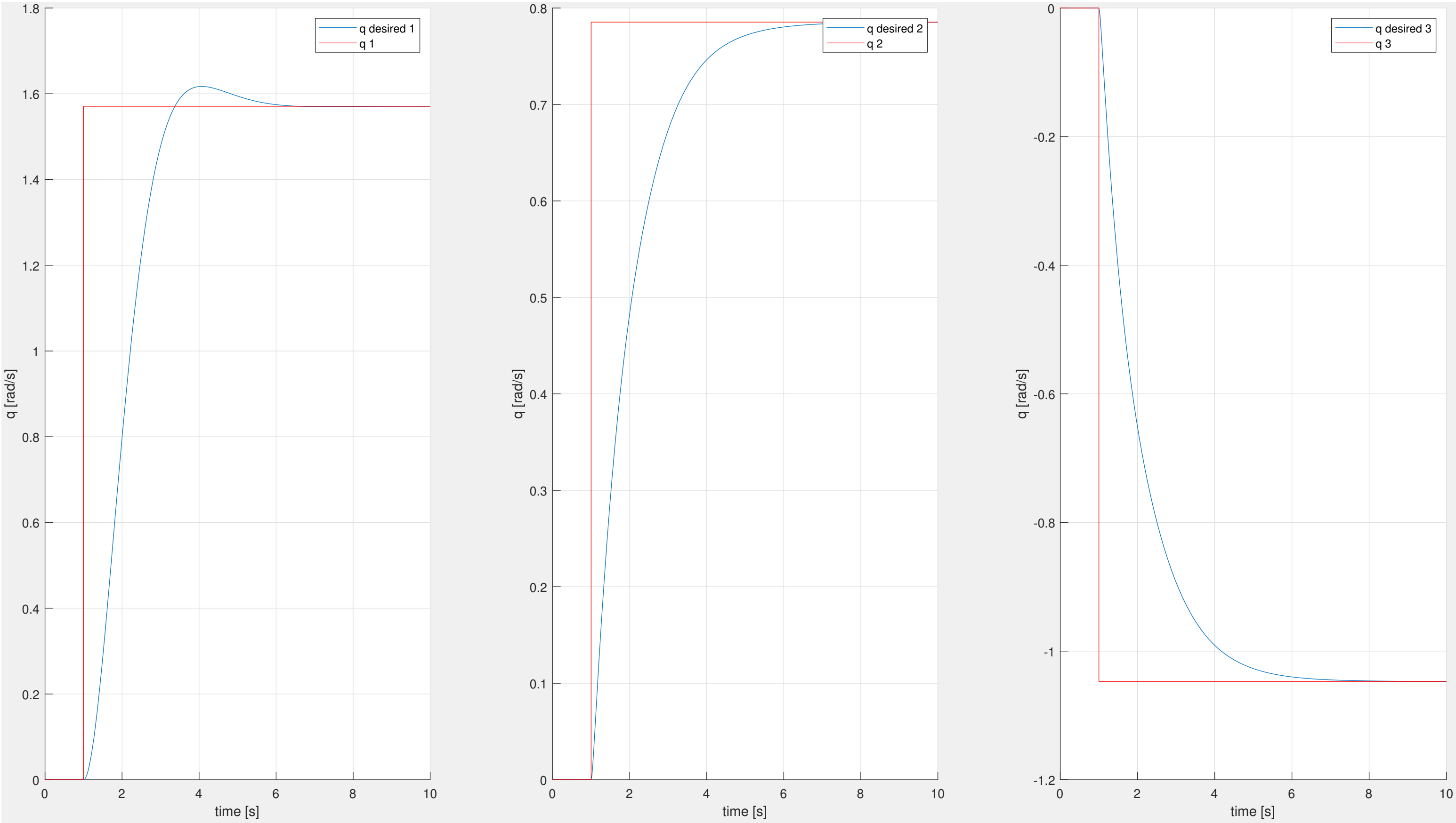

Figure 5: Joint Space PD Control with gravity compensation

6.3 Joint space PD control law without gravity compensation

It can be seen that if the gravity compensation is disabled, then the steady-state error is not 0: we have a offset on q_1, q_2, i.e. the joints affected by gravity cannot reach the desired position.
It can also be noticed that in the time interval $[0, 1]$ the joint variables q_2 and q_3 fall.
This is due to the fact that the controller action does not start until the reference is different from 0, hence before $t = 1s$ there is nothing that prevents the last two revolute joints to fall towards the ground due to the gravity force.

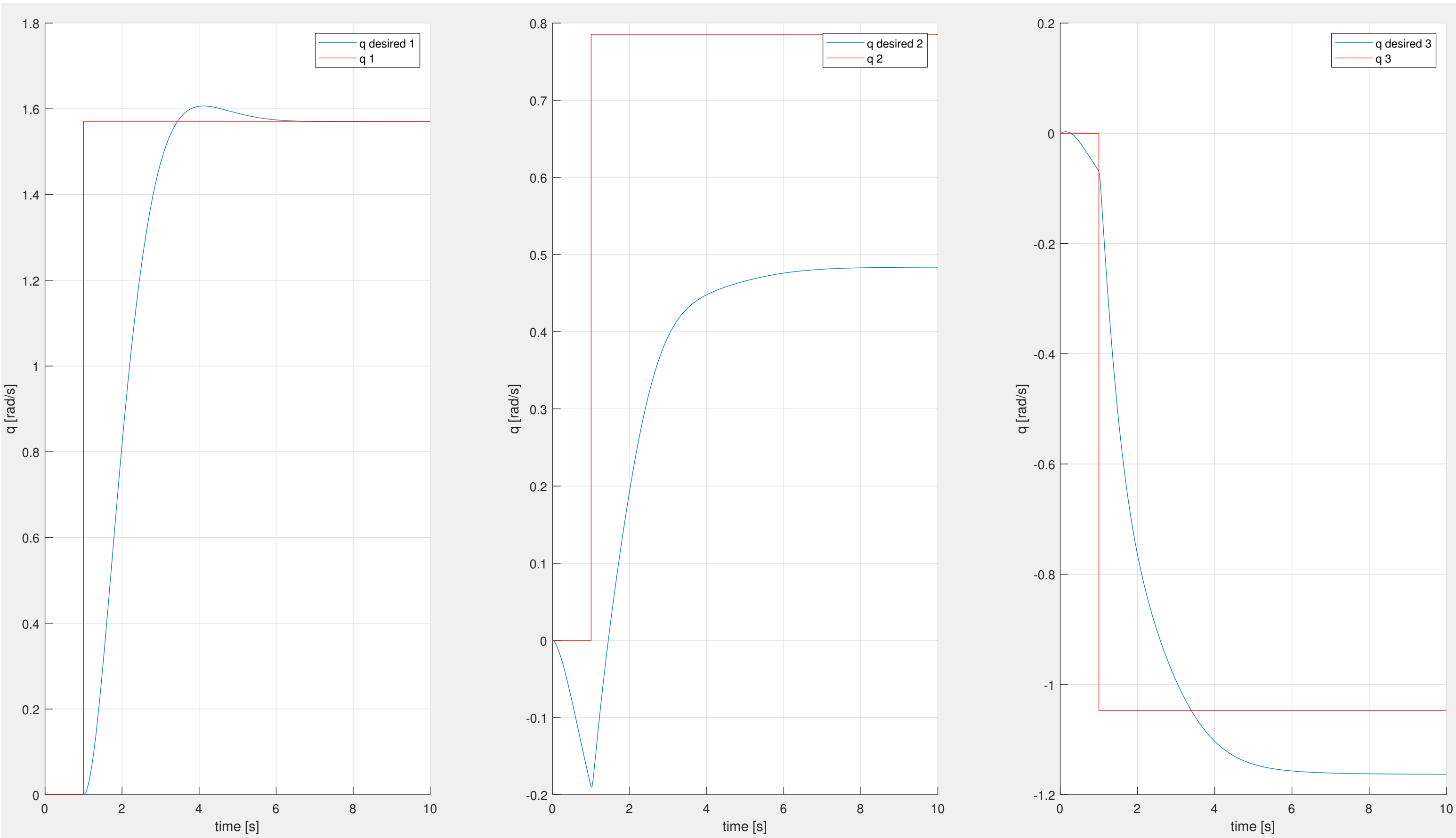

Figure 6: Joint Space PD Control without gravity compensation

6.4 Joint space PD control law with constant gravity compensation

If gravity forces are compensated with $g(q_d)$ instead of $g(q)$, we still get 0 steady state error, but there is a difference in the transient. In the following figure, by looking at the second joint we can see the difference between the two cases: constant gravity compensation and time-varying gravity compensation.

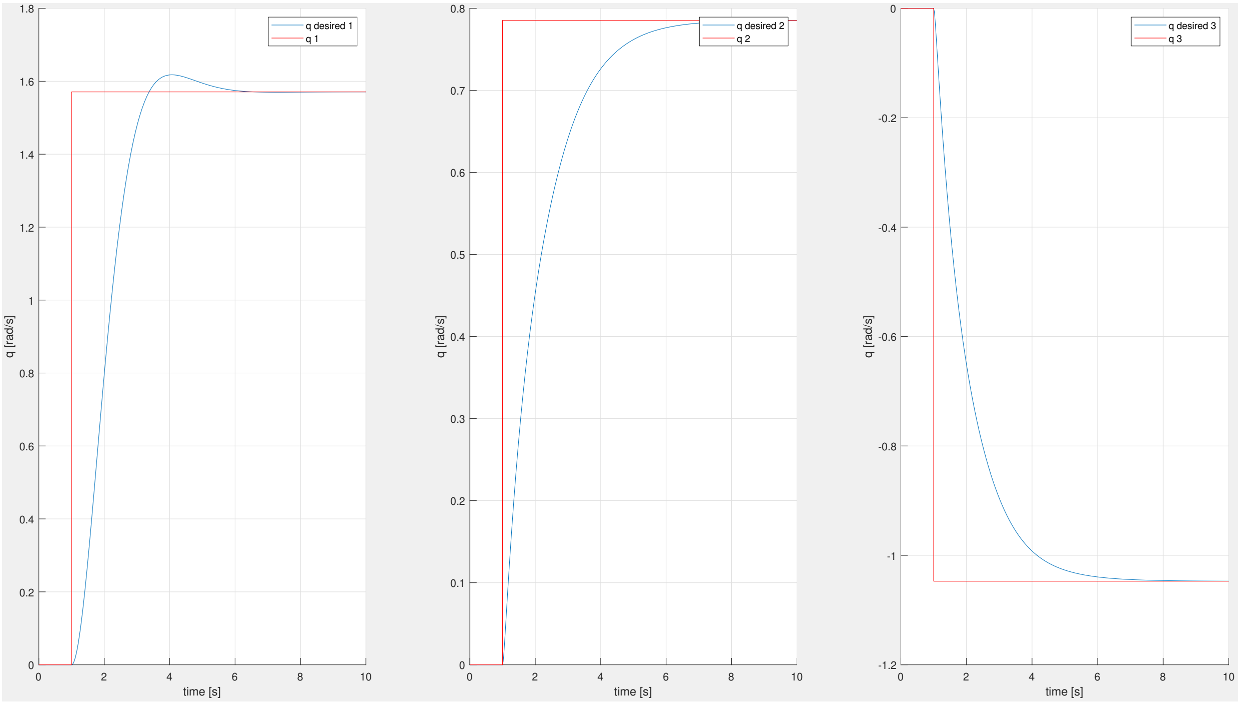

Figure 7: Joint Space PD Control with constant gravity compensation

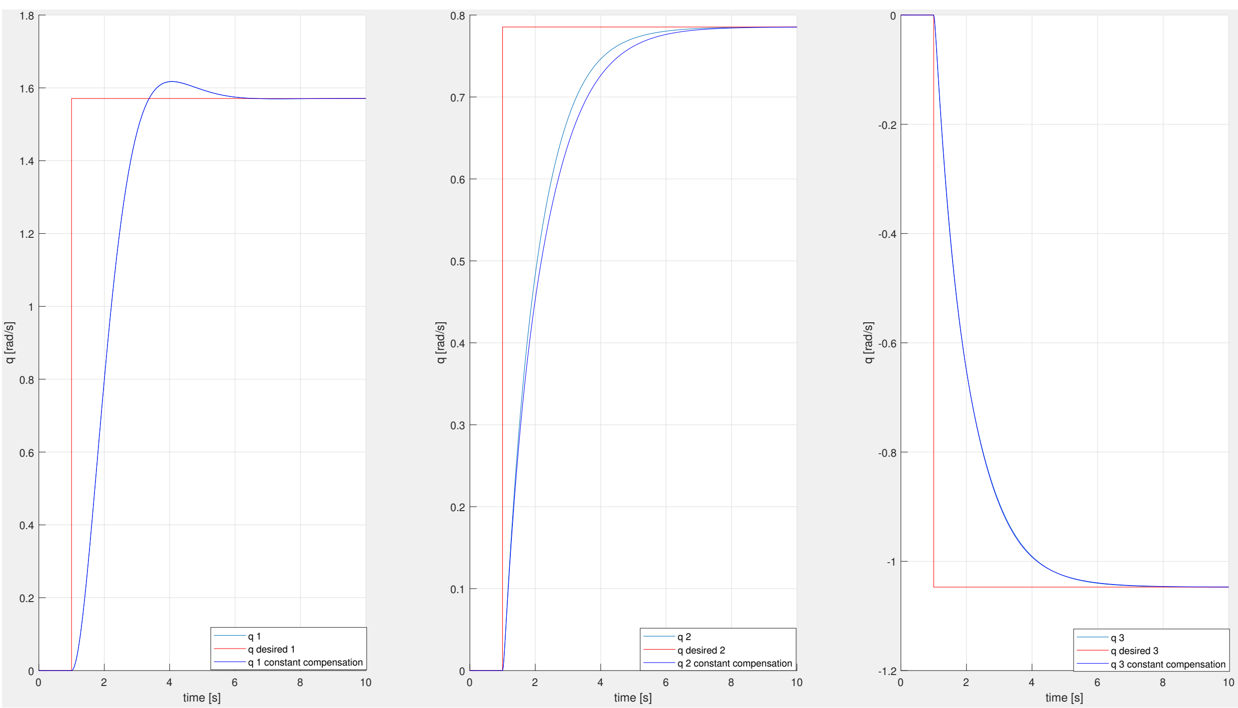

Figure 8: Joint Space PD Control with constant vs time-varying gravity compensation comparison

6.5 Joint space PD control law with gravity compensation and noisy reference

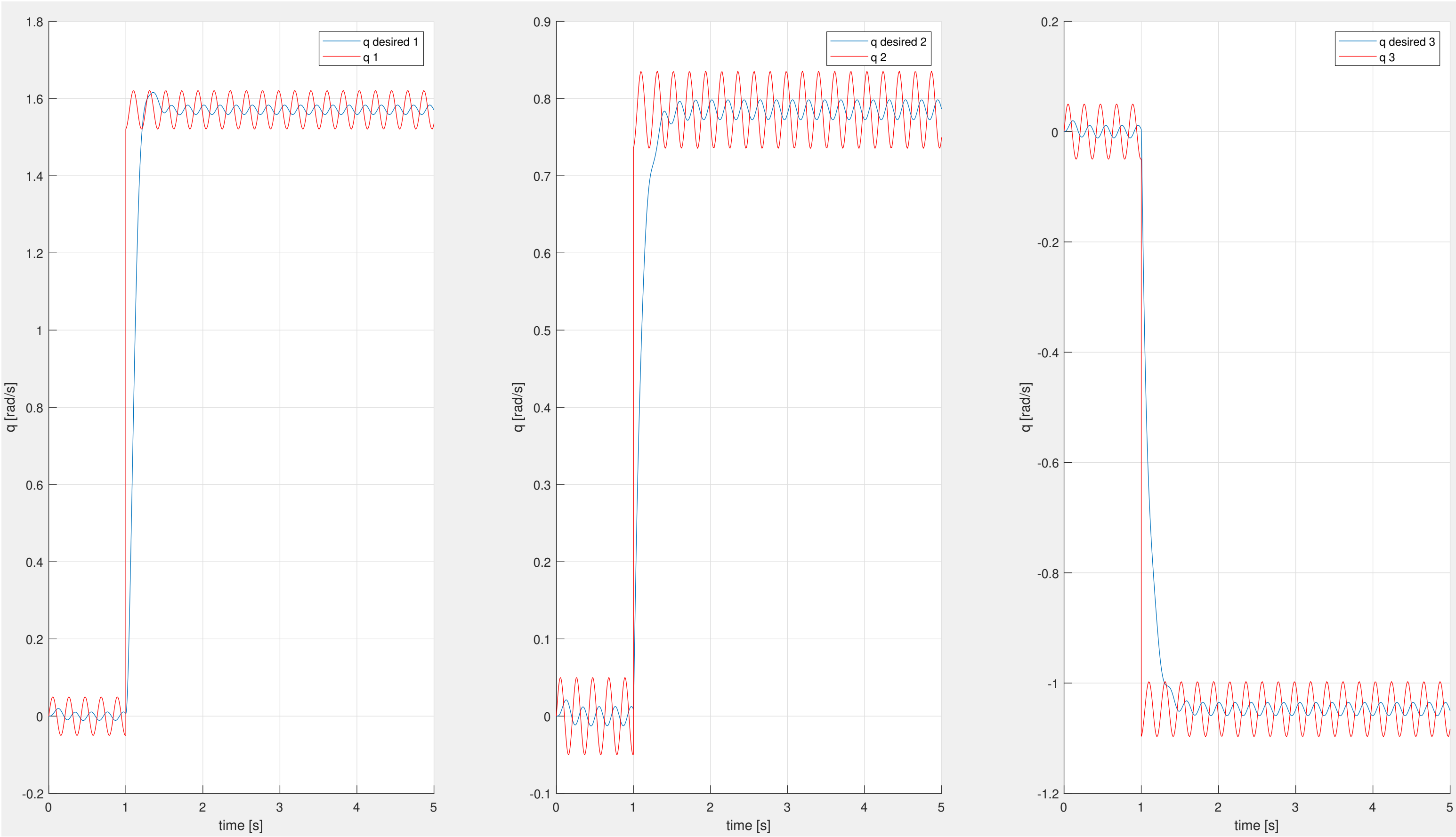

Figure 9: Joint Space PD Control with gravity compensation and noisy reference

7 Homework 7

7.1 Assignment description

- Design the Joint Space Inverse Dynamics Control law
- Check that in the nominal case the dynamic behaviour is equivalent to the one of a set of stabilized double integrators
- Check the behavior of the control law when the B, C, g used within the controller are different than the “true ones” B;C; g (e.g. slightly modify the masses, the frictions, ...)
- What happens to the torque values when the settling time of the equivalent second order systems is chosen very small?

7.2 Considerations

This control scheme is composed of two different parts:

- non linear state feedback
- stabilizing linear controller (PD controller)

Assuming perfect knowledge of the $B(q)$, $C(q, \dot{q})$, $g(q)$ matrices we get a perfect cancellation of the dynamic terms. In a real case scenario questions about sensitivity and robustness should be taken into account.
If K_p and K_d are diagonal matrices then we have n decoupled second order systems. The n homogeneous second-order differential equation on the position error $\tilde{q}(t)$ converge to zero depending on the choice of K_p and K_d.

7.3 Joint Space Inverse Dynamics

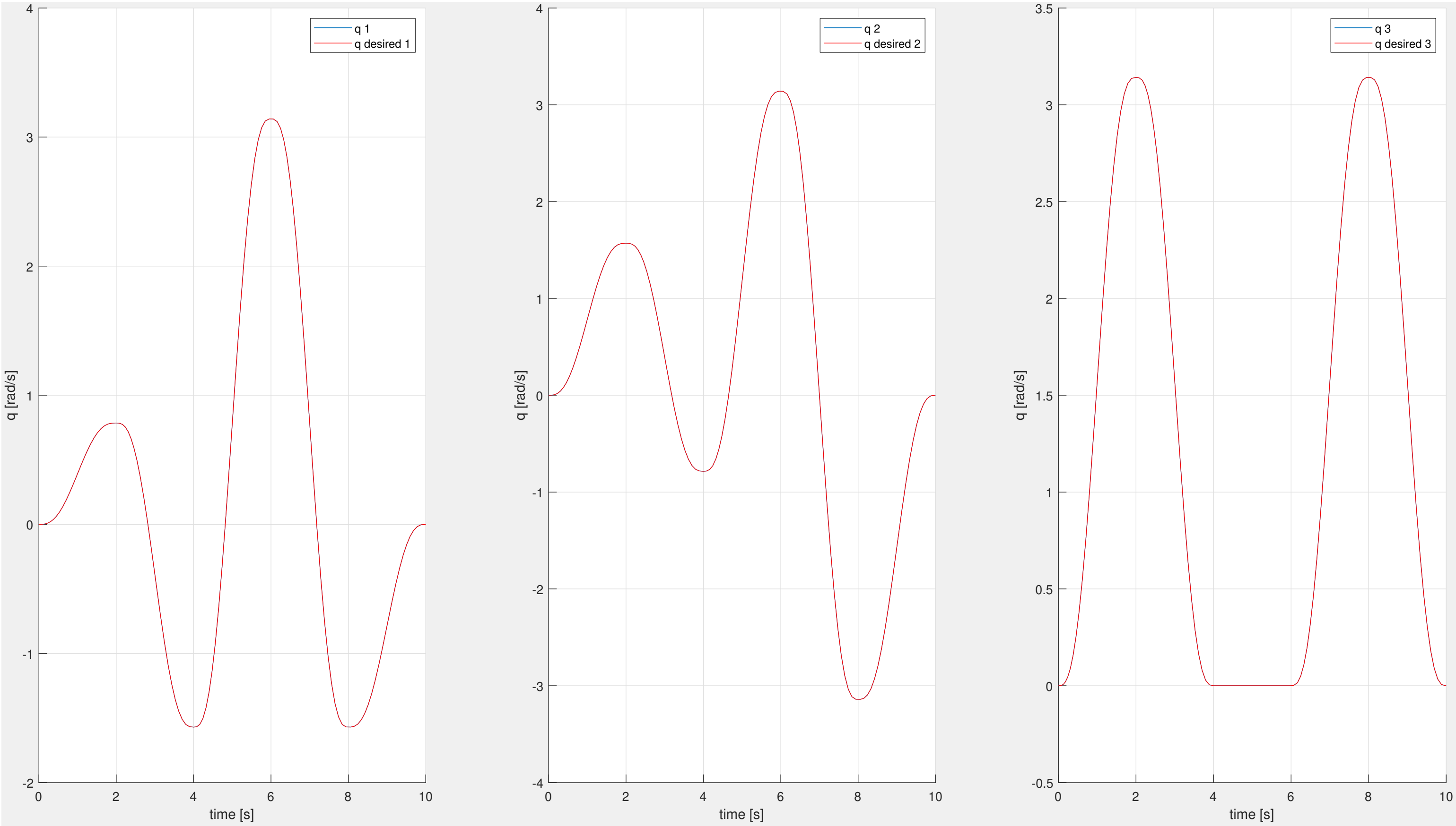

Figure 10: Joint Space inverse dynamics - joint positions

7.4 Joint Space Inverse Dynamics with wrong matrices

The matrix $B(q)$ has been multiplied by a gain of 0.1, while $n = C(q, \dot{q}) + g(q)$ has been multiplied by a gain of 0.2.

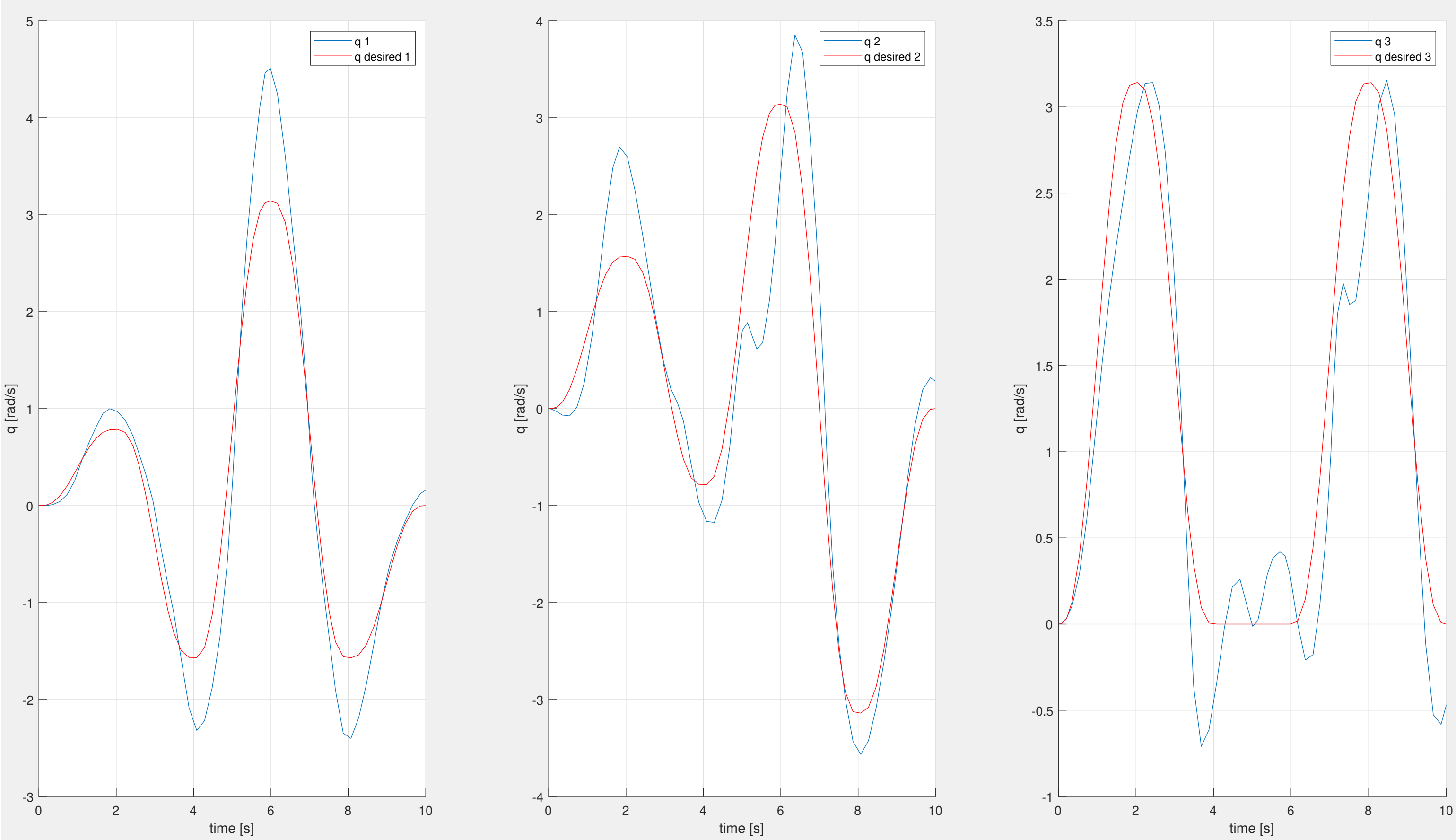

Figure 11: Joint Space inverse dynamics with wrong matrices - joint positions

7.5 Double integrators

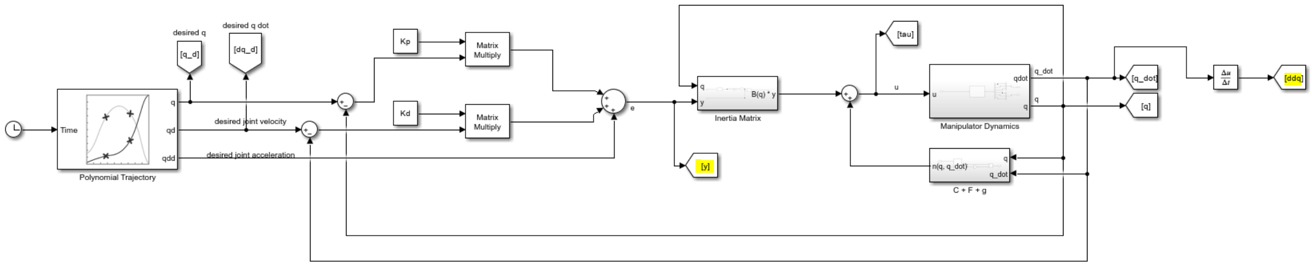

Figure 12: Joint Space inverse dynamics - schema

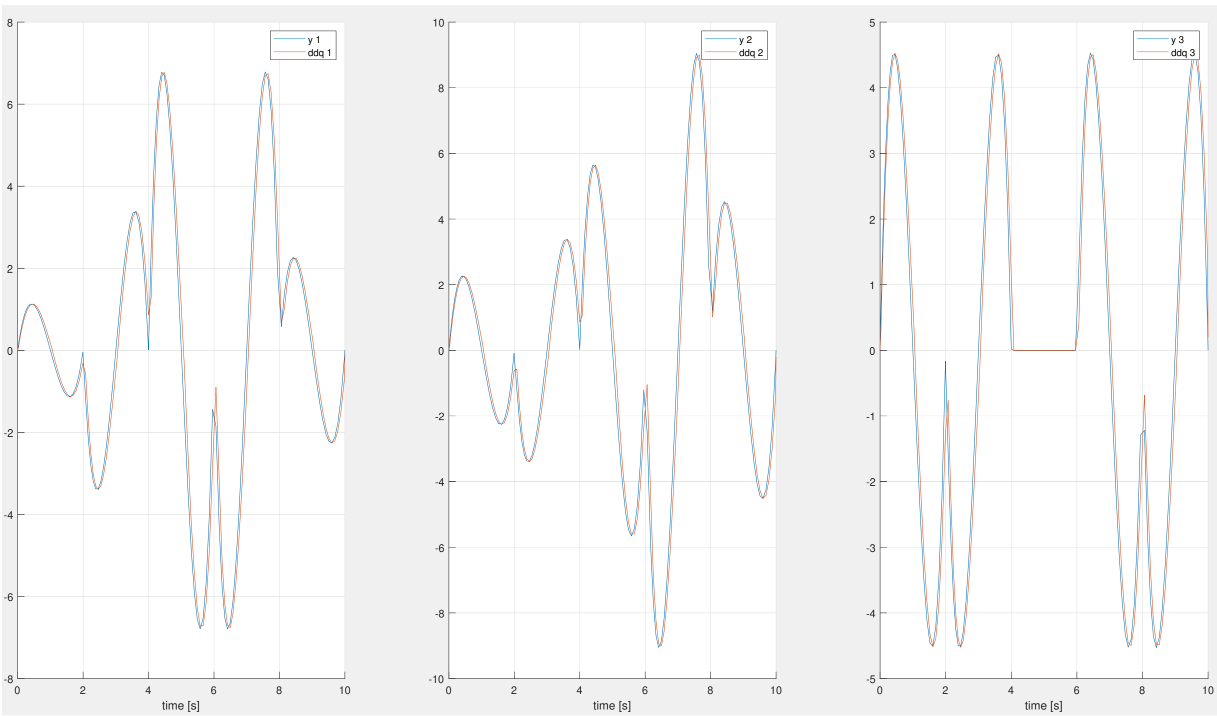

Figure 13: Joint Space inverse dynamics - $y = \ddot{q}$

8 Homework 8

8.1 Assignment description

- Implement in Simulink the Adaptive Control law for the a 1-DoF link under gravity.
- Choose the dynamic parameters and their initial estimates, and set the desired trajectory as
  1. $q_d(t) = Asin(wt)$
  2. $\ddot{q}_d(t)$ a periodic square wave $\pm A$

8.2 Considerations

Two starting conditions are compared:

- $\hat{\theta}_0 = 0$, i.e. we start without any indication about the dynamic parameters of the manipulator
- $\hat{\theta}_0 = \theta * 0.7$, i.e. we start close to the real dynamic parameters of the manipulator

It follows the real vector of dynamic parameters θ.
NB: The $C(q, \dot{q})$ matrix is not taken into account because we have 1 DOF only.

$$\theta = \begin{pmatrix} I \\ F \\ G \end{pmatrix} = \begin{pmatrix} 5 \\ 10 \\ -19.6 \end{pmatrix}$$

8.3 Adaptive Control with $\hat{\theta}_0 = 0$

The initial estimate of the dynamic parameters is set to 0.

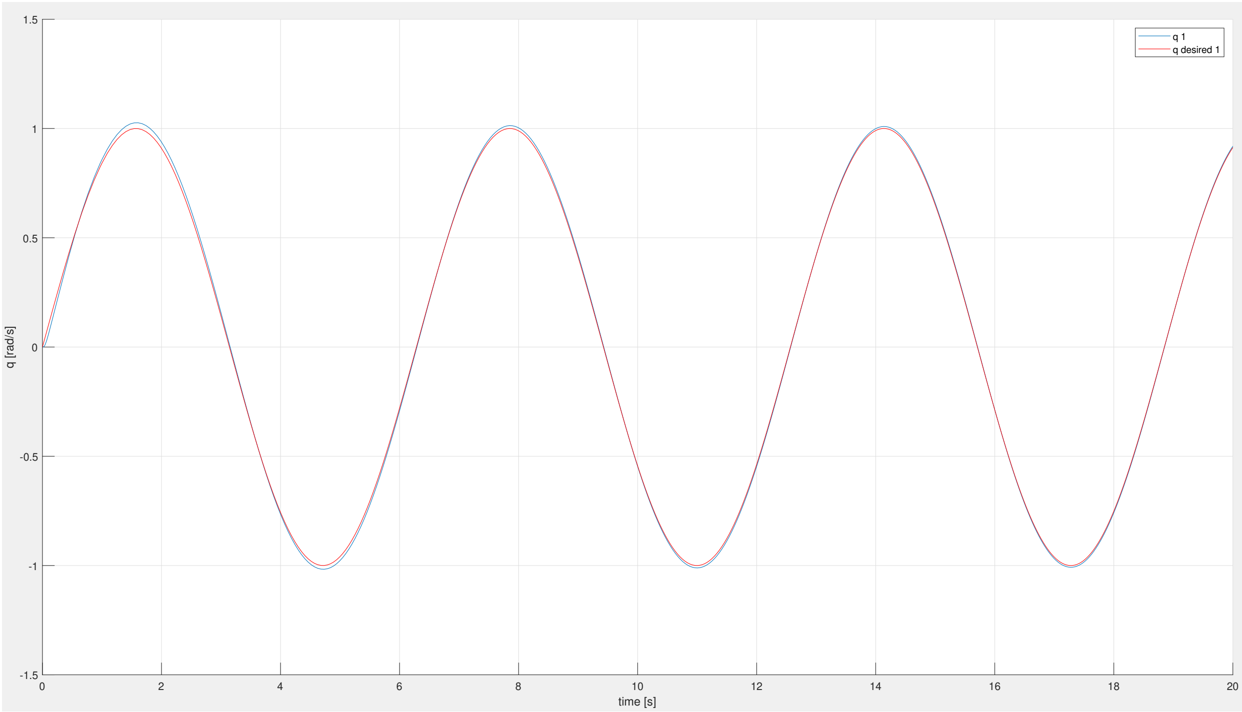

Figure 14: Adaptive control - joint trajectory

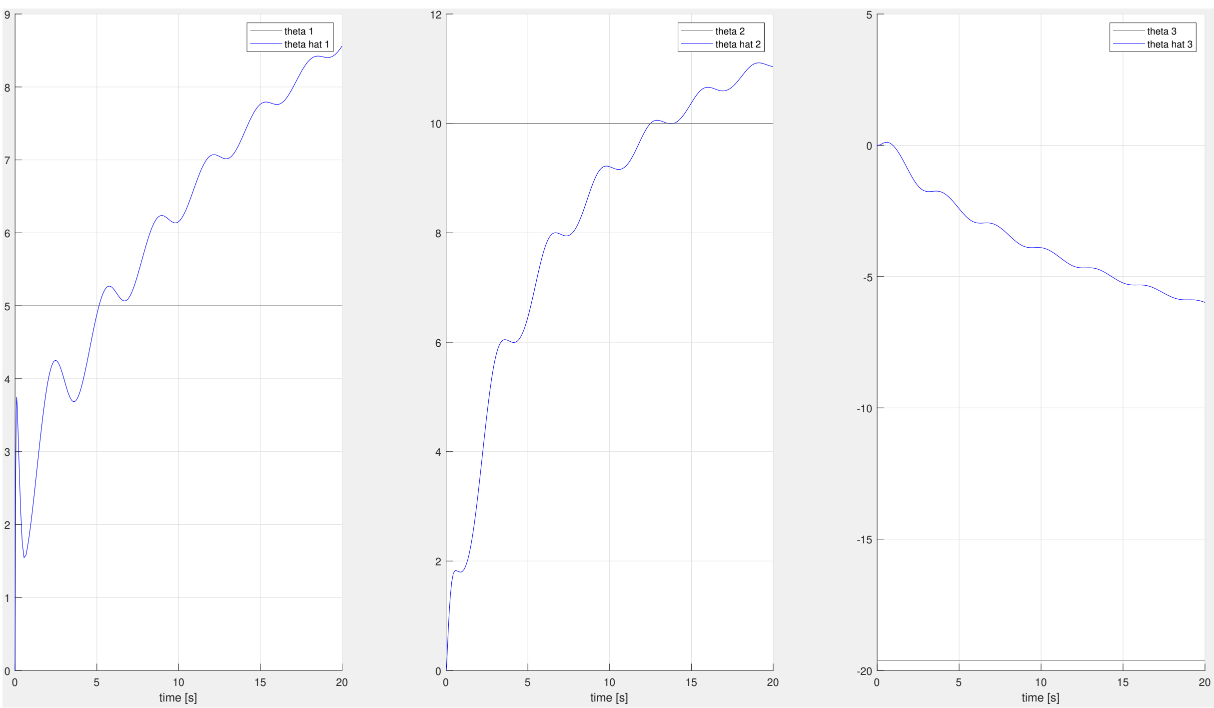

Figure 15: Adaptive control - theta hat trajectory

8.4 Adaptive Control with $\hat{\theta}_0 = 0.7\theta$

The initial estimate of the dynamic parameters is set to the true values θ multiplied by a gain of 0.7. Even though $\hat{\theta}$ is not converging to θ, the trajectory error is negligible.

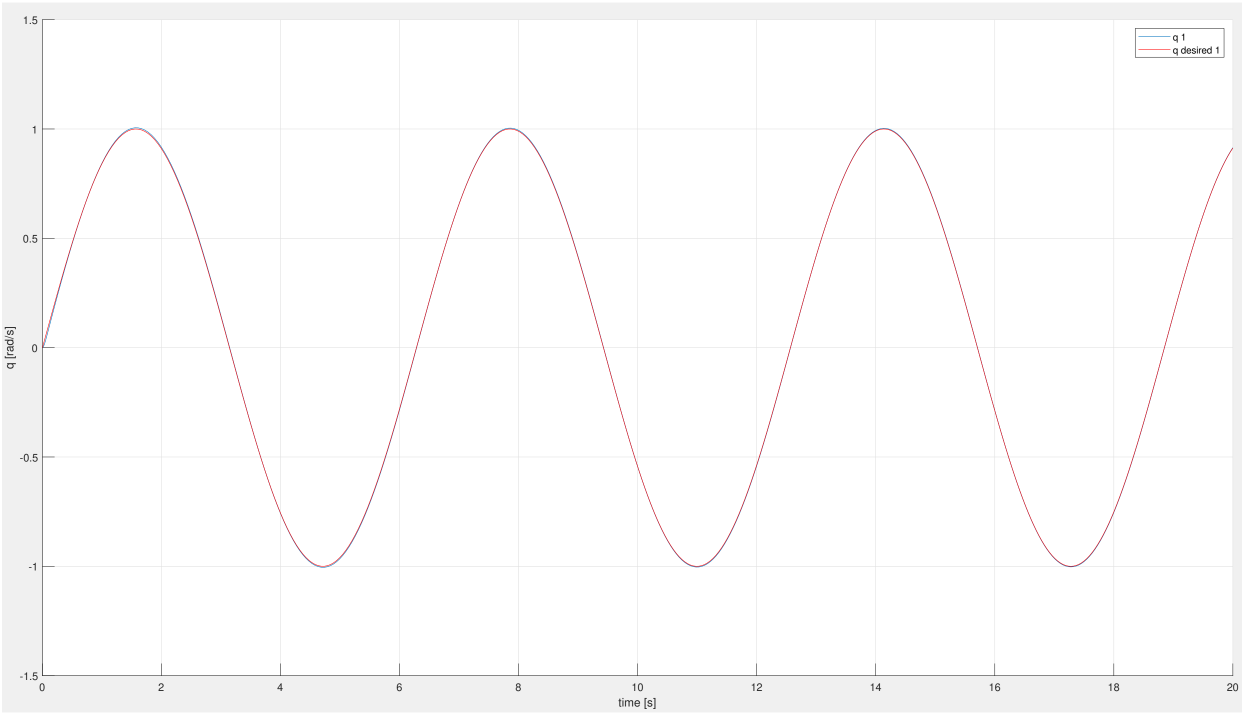

Figure 16: Adaptive control - joint trajectory

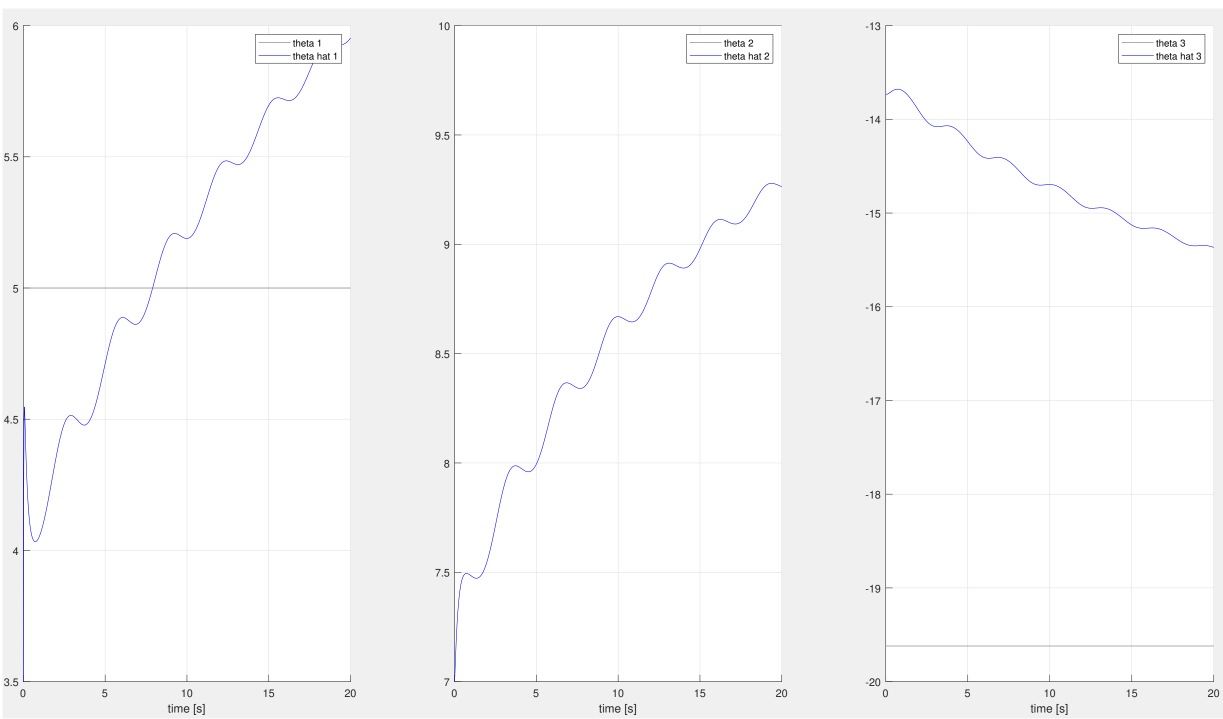

Figure 17: Adaptive control - theta hat trajectory

8.5 Periodic square wave acceleration input

The desired joint trajectory was computed from a periodic acceleration square wave.
The initial estimate of the dynamic parameters is set to the true values θ multiplied by a gain of 0.7.

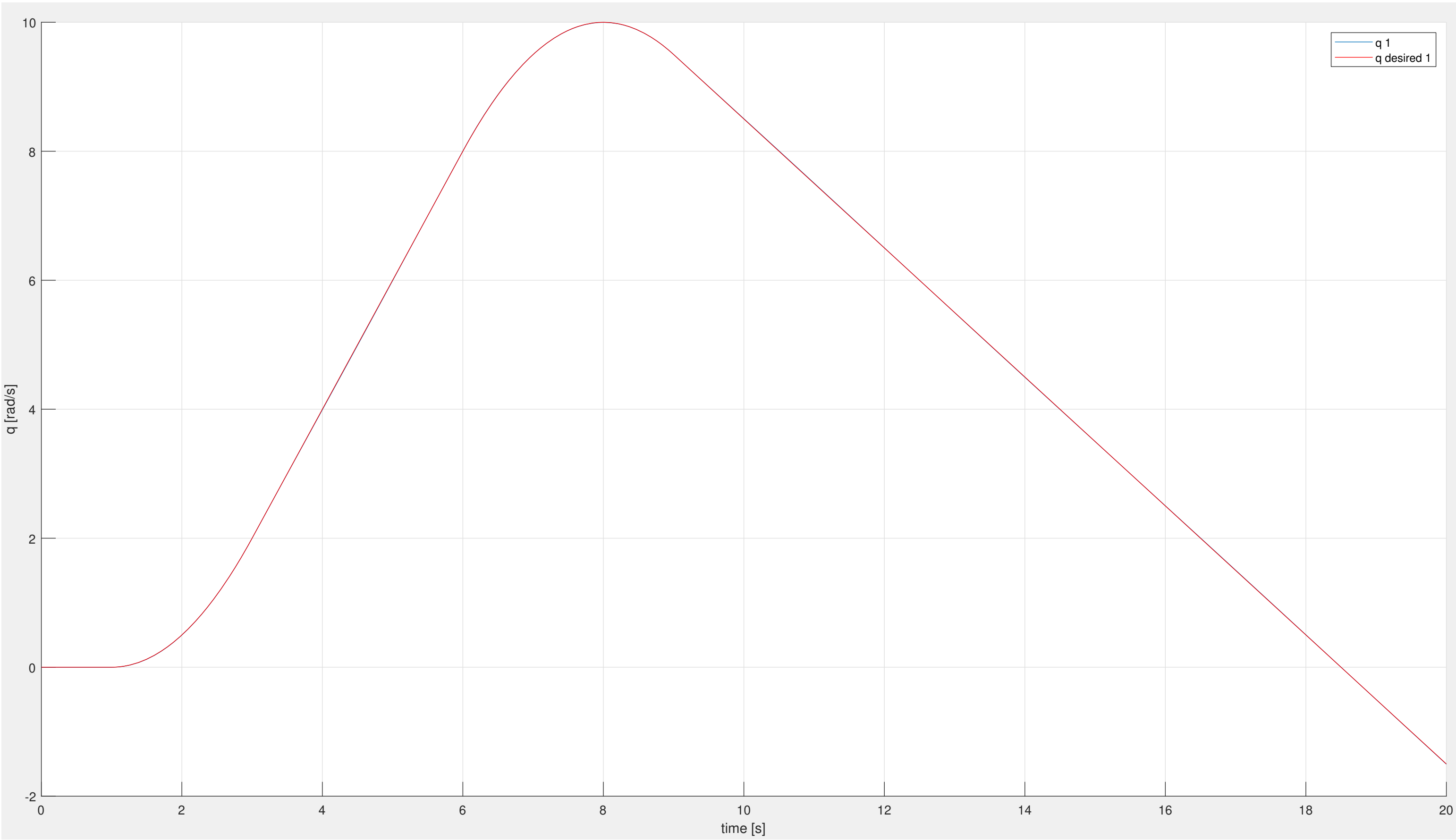

Figure 18: Adaptive control - joint trajectory

9 Homework 9

9.1 Assignment description

- Design the Operational Space PD control law with gravity compensation

9.2 Considerations

If the analytical jacobian has full rank, then we are sure to achieve 0 steady-state error, i.e. the system converges to $x = x_d$.

9.3 Operational Space PD control law with gravity compensation

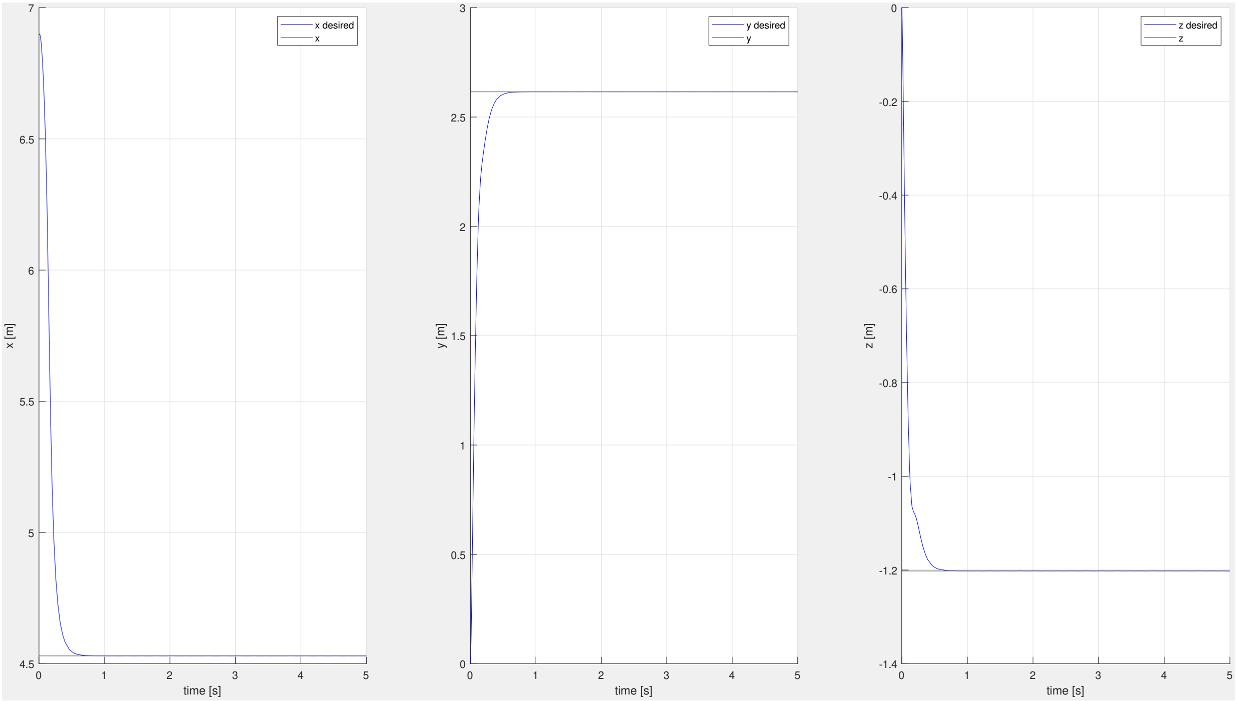

Figure 19: Operational Space PD control law with gravity compensation - ee position

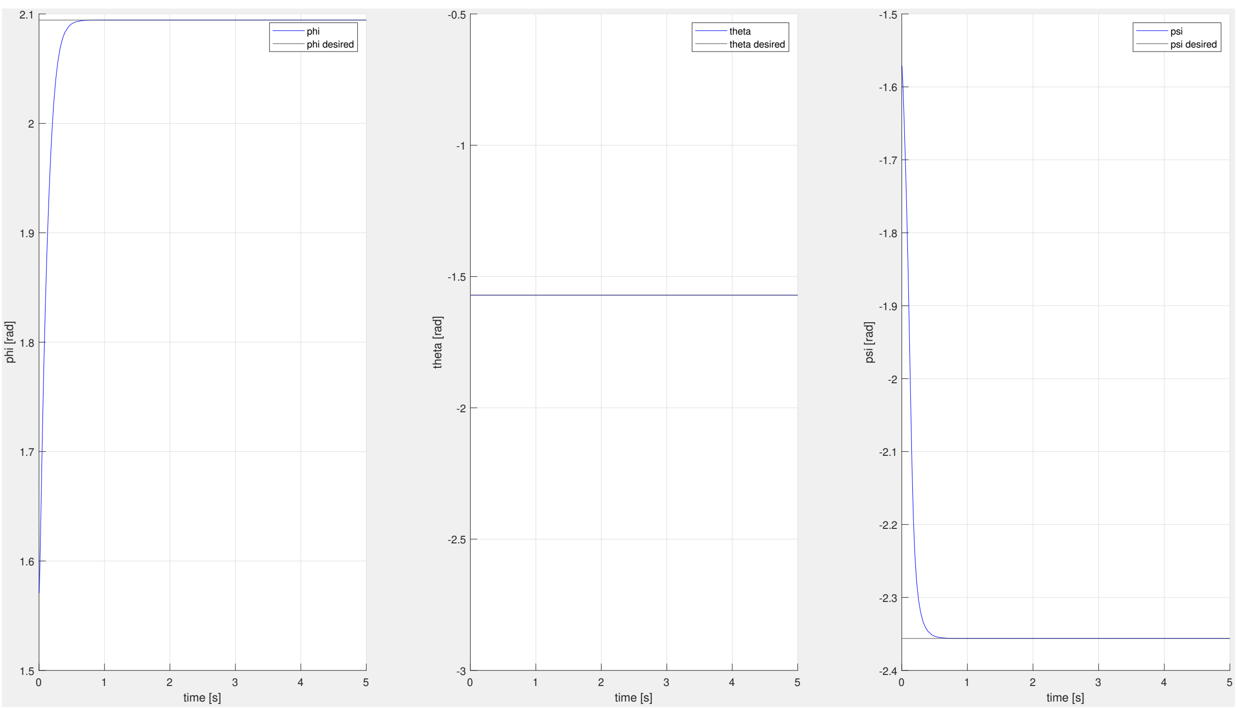

Figure 20: Operational Space PD control law with gravity compensation - ee orientation

9.4 Operational Space PD control law without gravity compensation

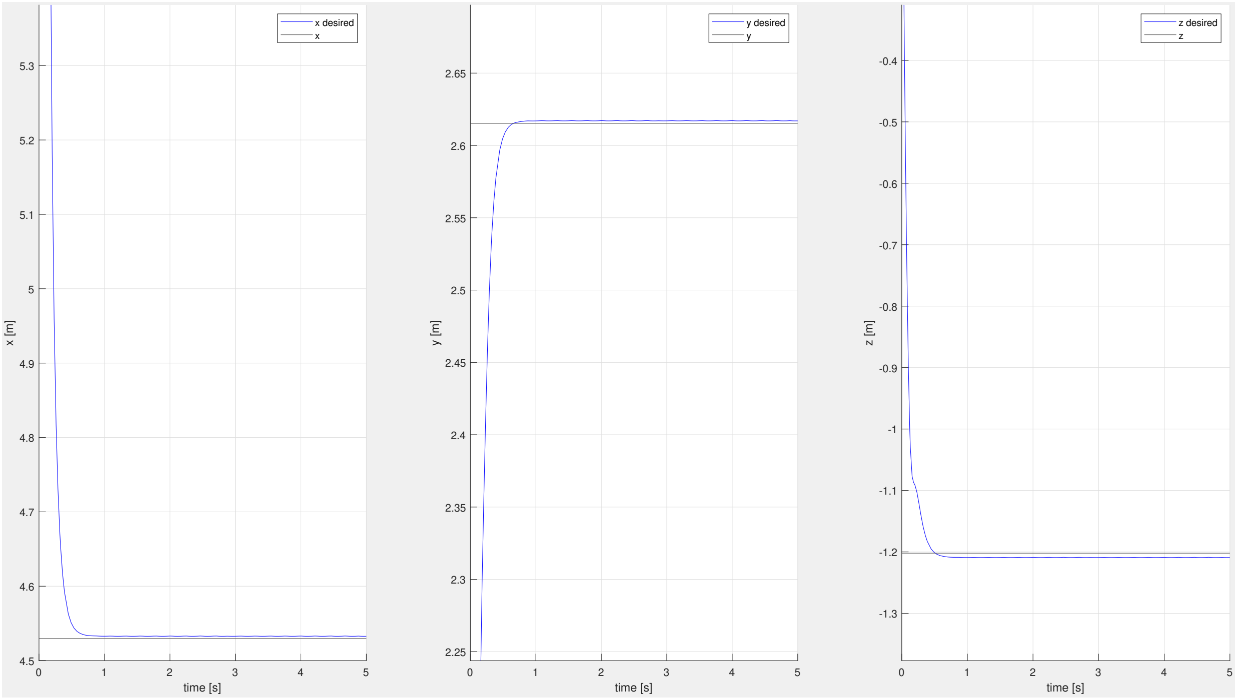

Figure 21: Operational Space PD control law without gravity compensation - ee position

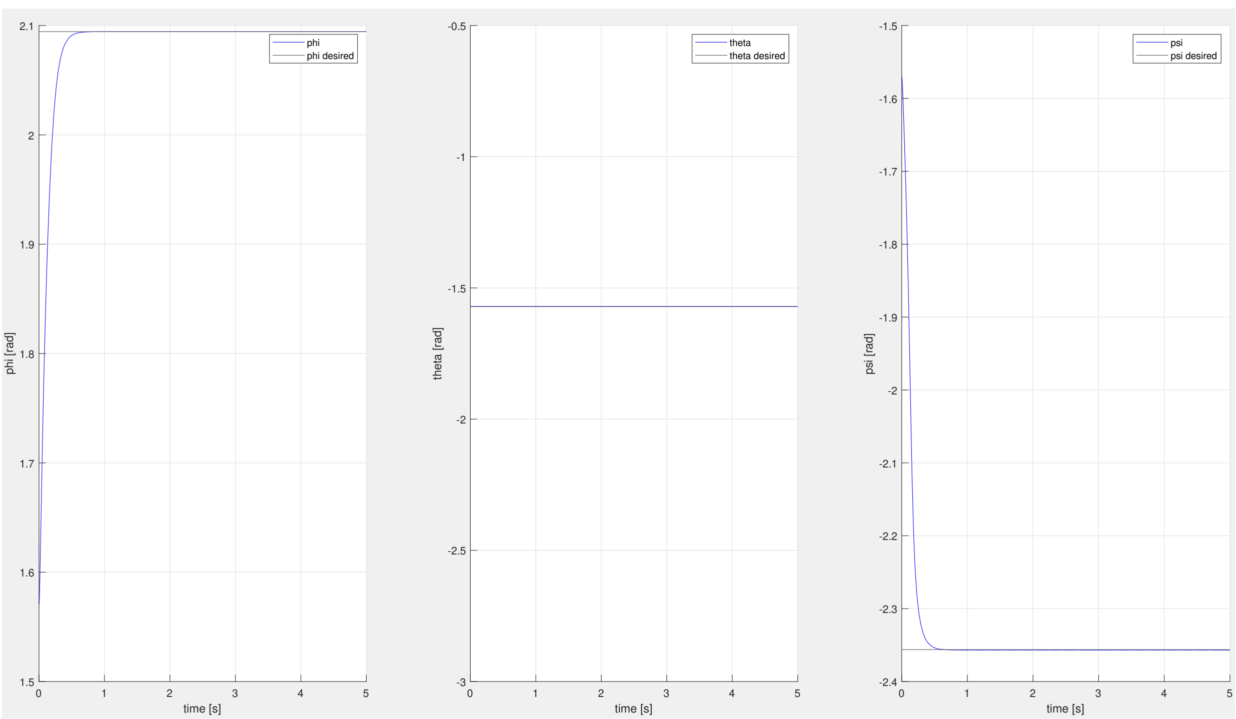

Figure 22: Operational Space PD control law without gravity compensation - ee orientation

10 Homework 10

10.1 Assignment description

- Design the Operational Space Inverse Dynamics Control law

10.2 Considerations

We have n second-order linear differential equations for the error in the operational space $\tilde{x}$.
The velocity of the convergence of the trajectory error to zero depends on the values of K_p and K_d.

10.3 Operational Space Inverse Dynamics Control law

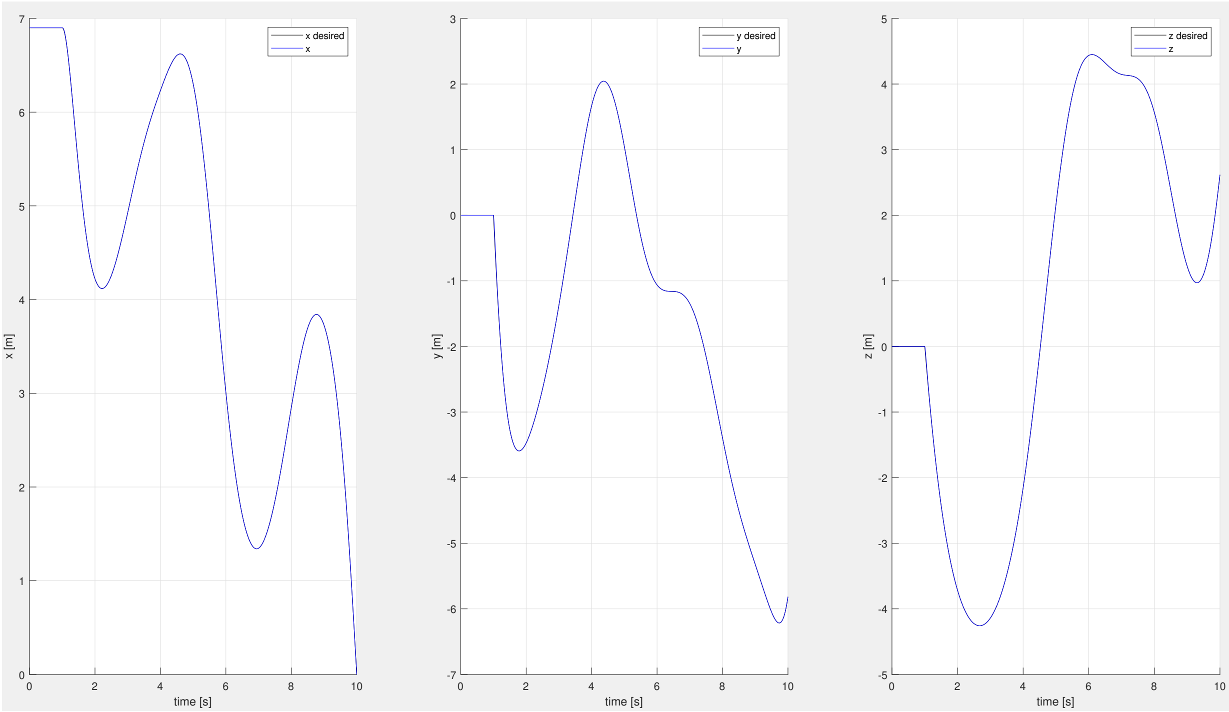

Figure 23: Operational Space Inverse Dynamics Control law - ee position

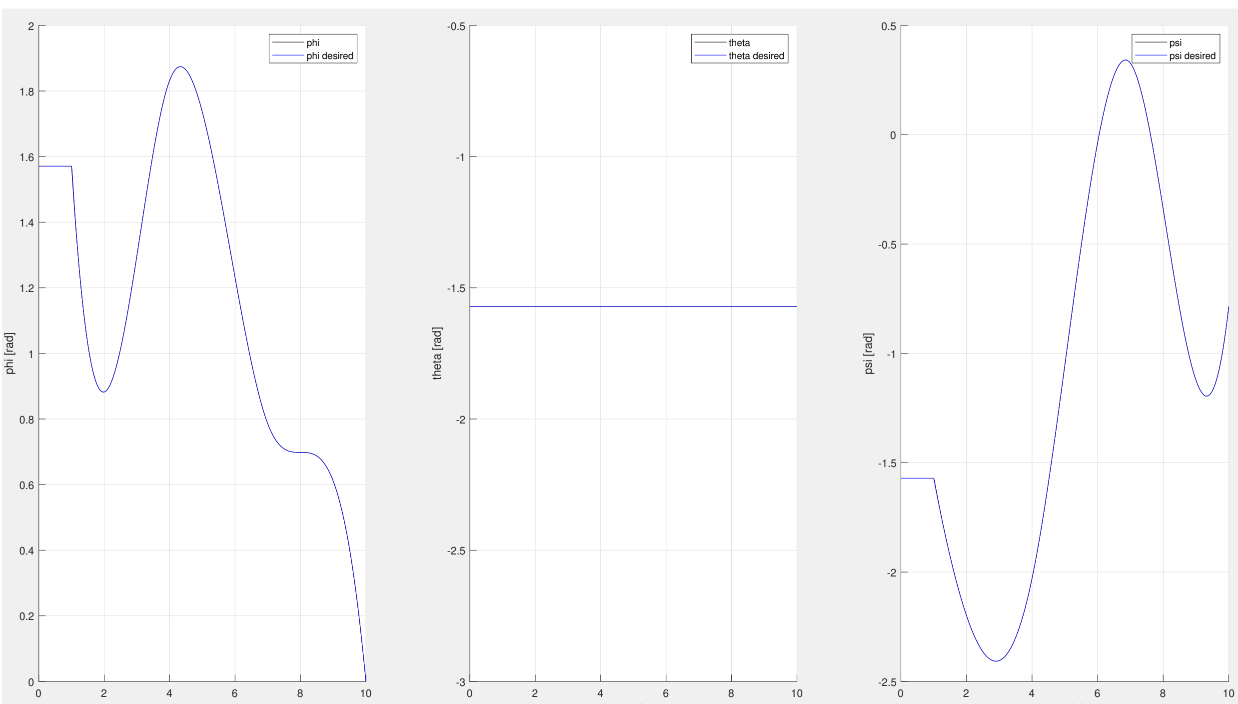

Figure 24: Operational Space Inverse Dynamics Control law - ee orientation

11 Homework 11

11.1 Assignment description

- Study the compliance control. Simulate the two "extreme" cases when the end-effector is interacting with the environment (considered as a planar surface)

11.2 Considerations

The environment is modeled as a spring of stiffness K that acts along the global z-axis, i.e. the spring generates an elastic force when 'compressed' along the global z-axis.
To enter in contact with the environment the manipulator needs to rotate the second or third joint in order to produce a displacement on the global z-axis.
Since both the joints that can produce a displacement on the global z-axis are revolute, two considerations must be noted:

- the generated displacement is not linear
- the generated displacement does not concern only the z-axis, but also the x-axis

In the following example, the chosen trajectory concerns the movement of the third joint only, i.e. the second joint does not rotate. The following images show:

- the starting configuration of the manipulator
  - no contact with environment, the spring is not active
- the manipulator desired configuration
  - contact with environment, positive displacement, elastic force > 0
- the manipulator configuration at the point of contact with the environment rest position
  - contact with environment, zero displacement, elastic force $= 0$

NB: The desired configuration is *beyond* the environment rest position.

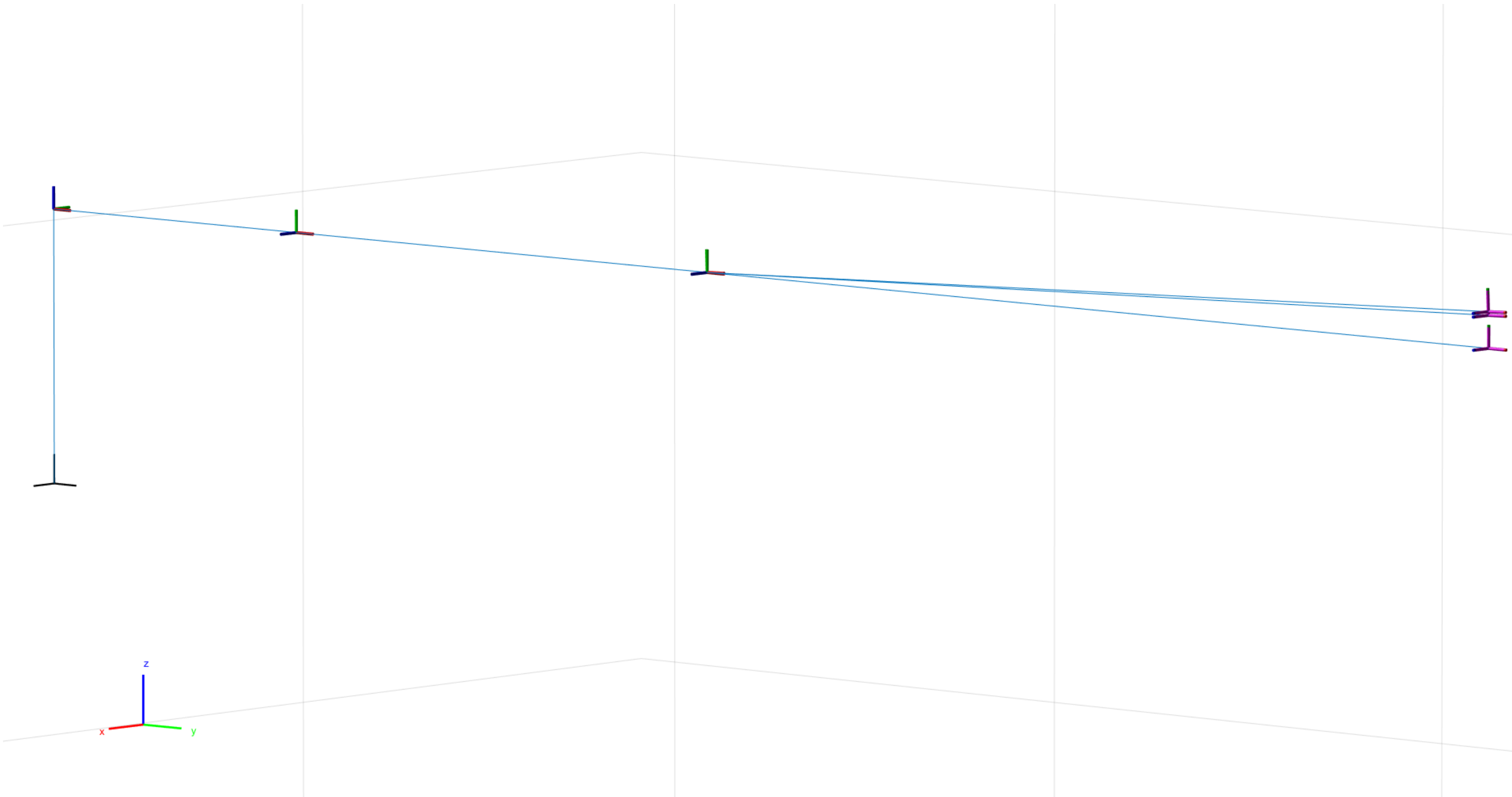

Figure 25: The three different manipulator configurations

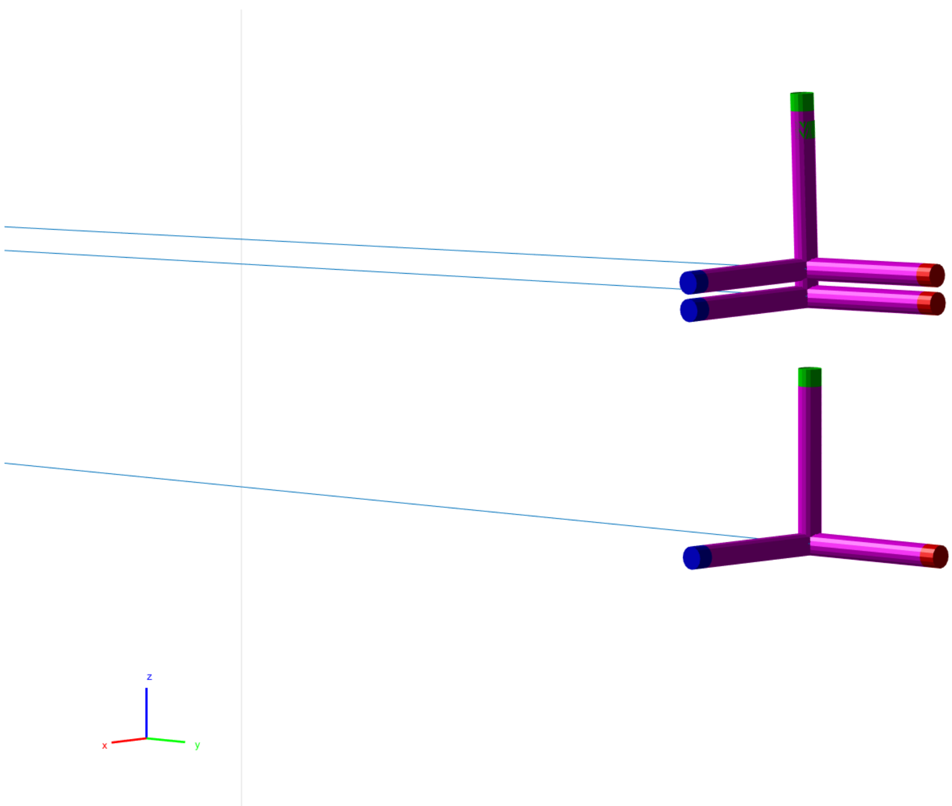

Figure 26: From bottom to top: manipulator start configuration, configuration of first contact, desired configuration

11.3 Compliance control - $k_e >> k_p$

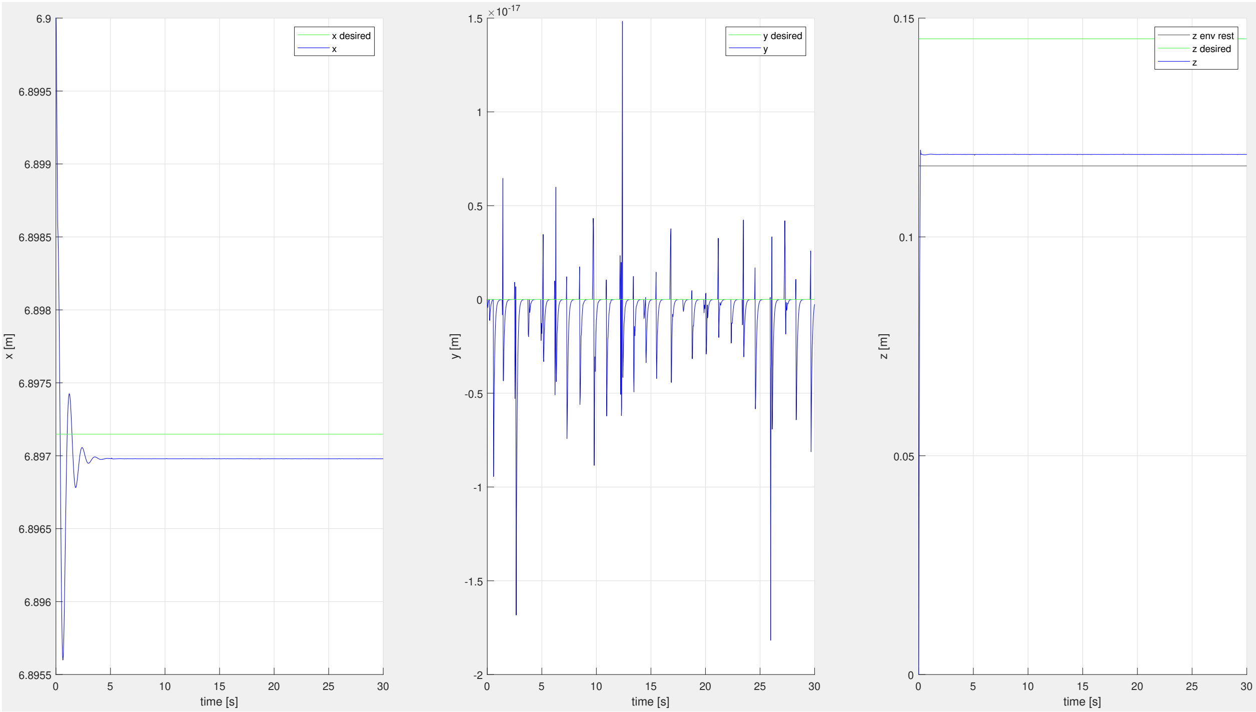

Figure 27: Compliance control - ee position

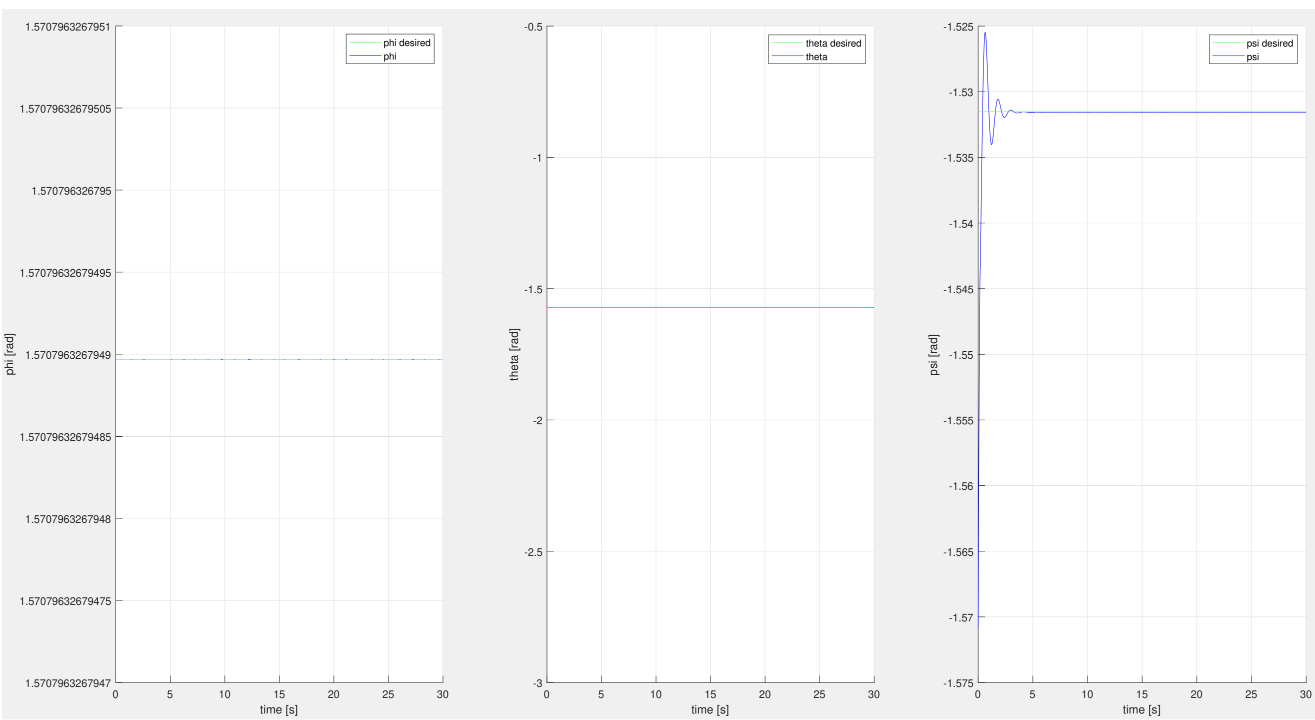

Figure 28: Compliance control - ee orientation

11.4 Compliance control - $k_e << k_p$

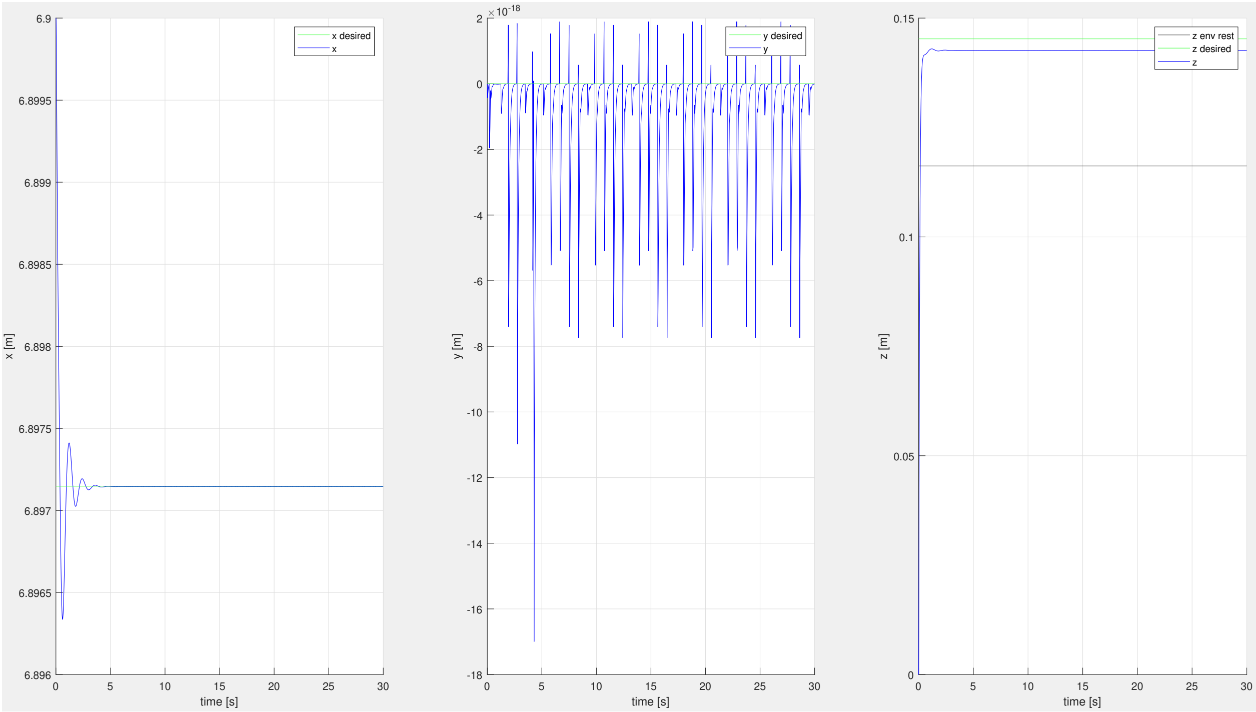

Figure 29: Compliance control - ee position

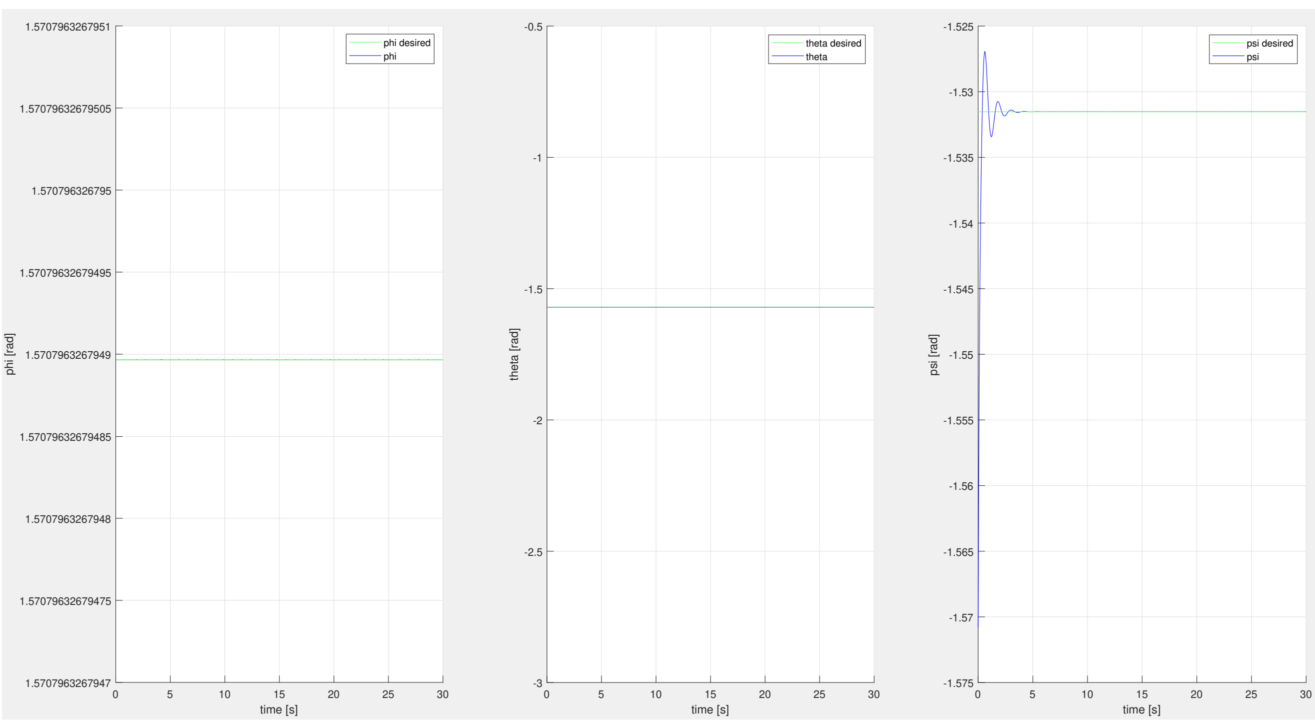

Figure 30: Compliance control - ee orientation

12 Homework 12

12.1 Assignment description

- Implement the impedance control in the operational space.

12.2 Considerations

The same considerations of the previous assignment also apply to this assignment.
As it can be seen from the following images, the trajectory is different form the previous one because the manipulator comes into contact with the environment two times, with different desired configurations.

12.3 Impedance control - $k_e >> kp$

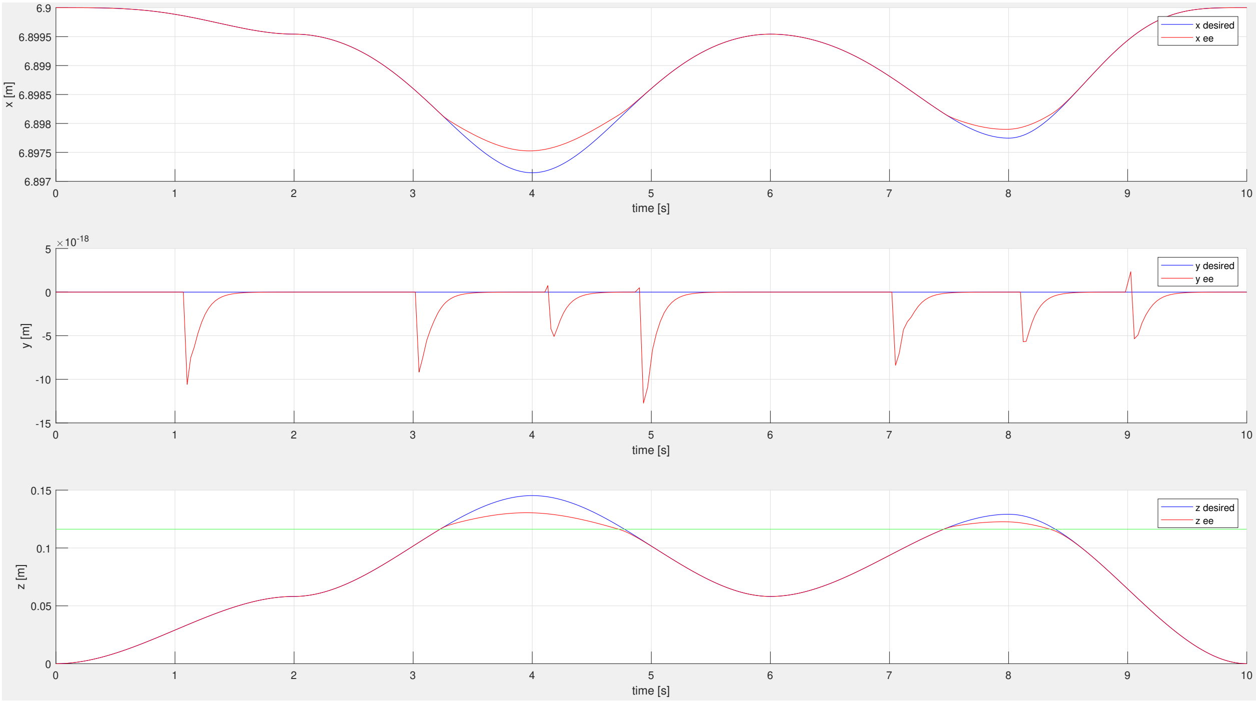

Figure 31: Impedance control - ee position

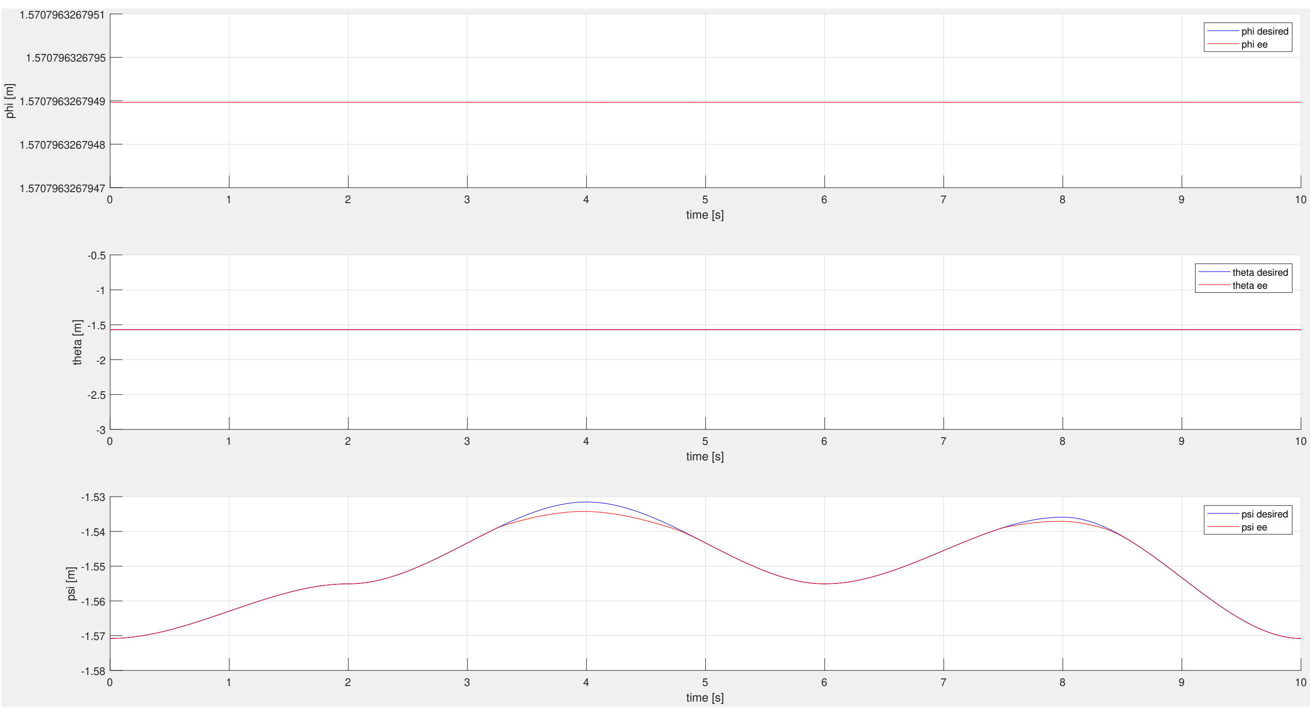

Figure 32: Impedance control - ee orientation

12.4 Impedance control - $k_e << kp$

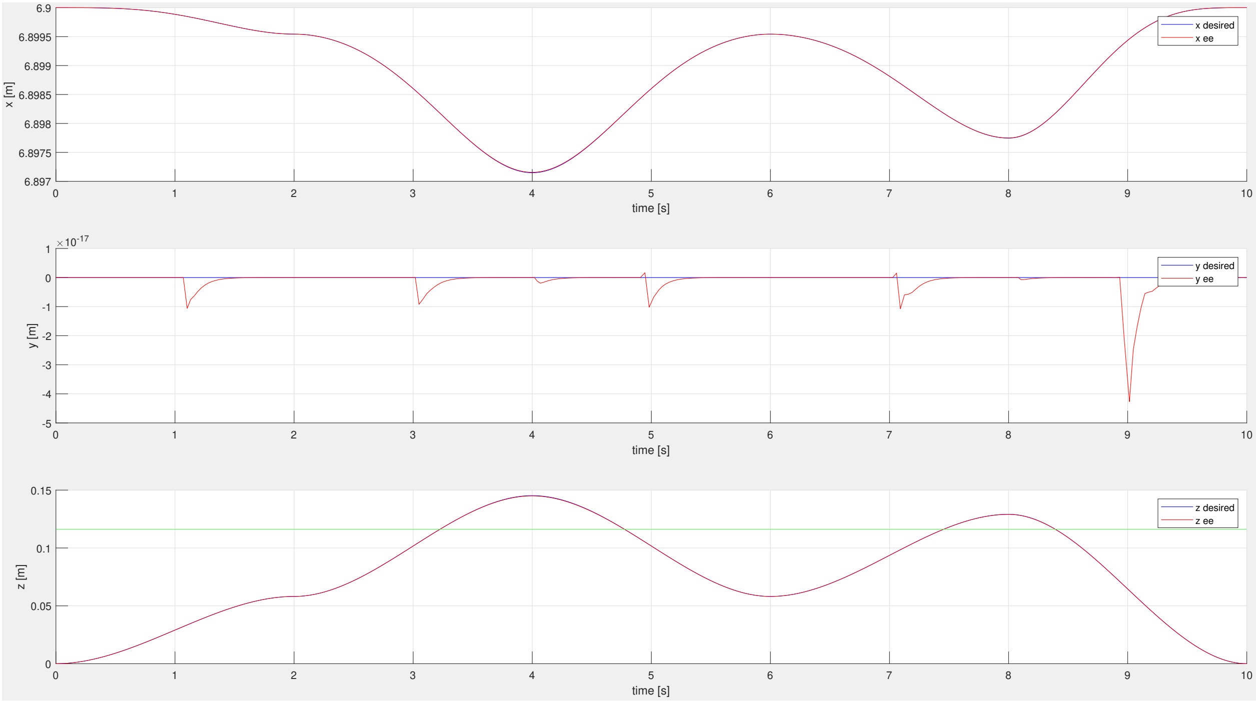

Figure 33: Impedance control - ee position

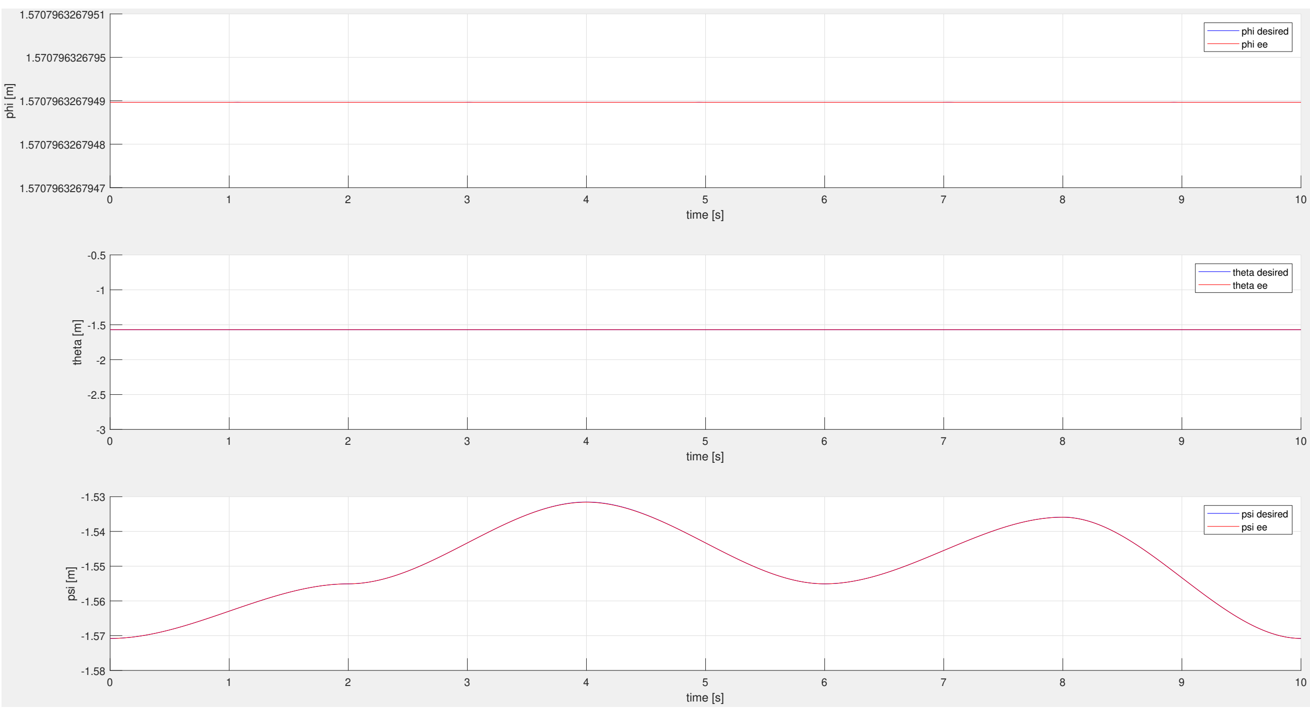

Figure 34: Impedance control - ee orientation

12.5 Impedance control - wrenches

As it can be seen from the second plot, the torque applied to the first joint due to the interaction forces with the environment is always 0.
This is because

$$h_e = \begin{pmatrix} 0 \\ 0 \\ h_z \end{pmatrix} \in null(J^T)$$

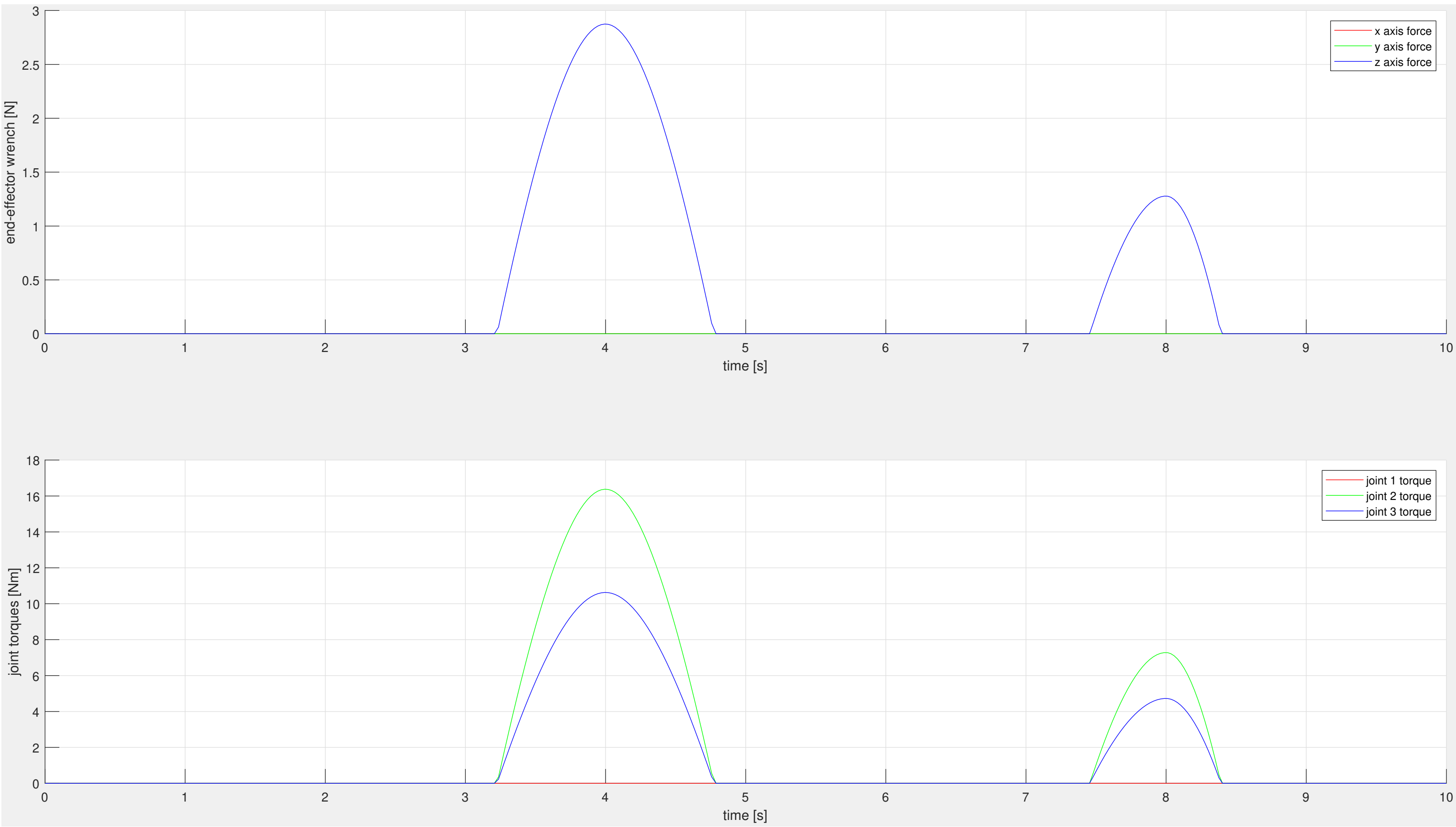

Figure 35: Impedance control - wrench in the cartesian space (top), wrench in the joint space (bottom)

13 Homework 13

13.1 Assignment description

- Implement the Force Control with Inner Position Loop.

13.2 Considerations

To be sure to stay away from kinematic singularities, the manipulator start configuration q_0 is the following one:

$$q_0 = \begin{pmatrix} 0 \\ \pi/2 \\ -pi/2 \end{pmatrix}$$

The S-function initial conditions have been updated to take into account the new starting configuration of the manipulator.

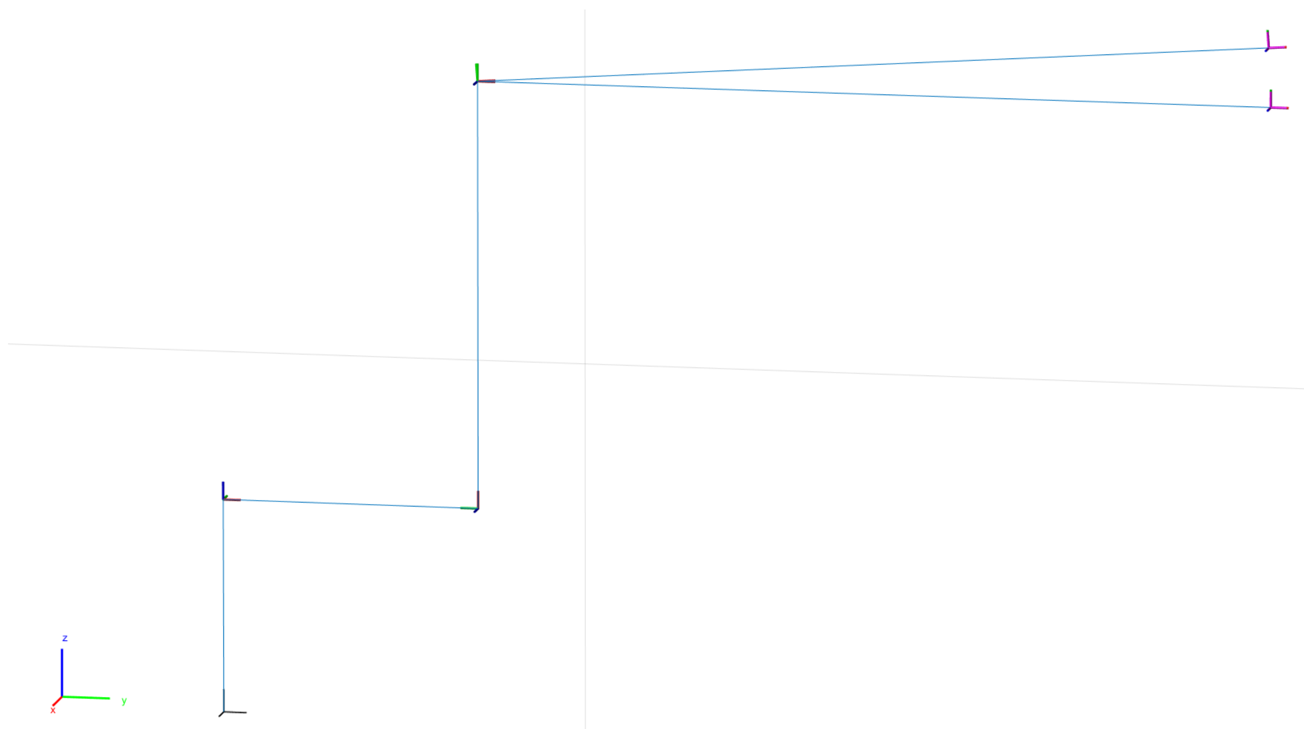

Figure 36: Force Control with Inner Position Loop - manipulator's start configuration (no contact), manipulator's configuration in contact with environment

13.3 Force Control with Inner Position Loop

NB: the position reference (x_{ref}, y_{ref}, z_{ref}) is the output of the outer force controller.

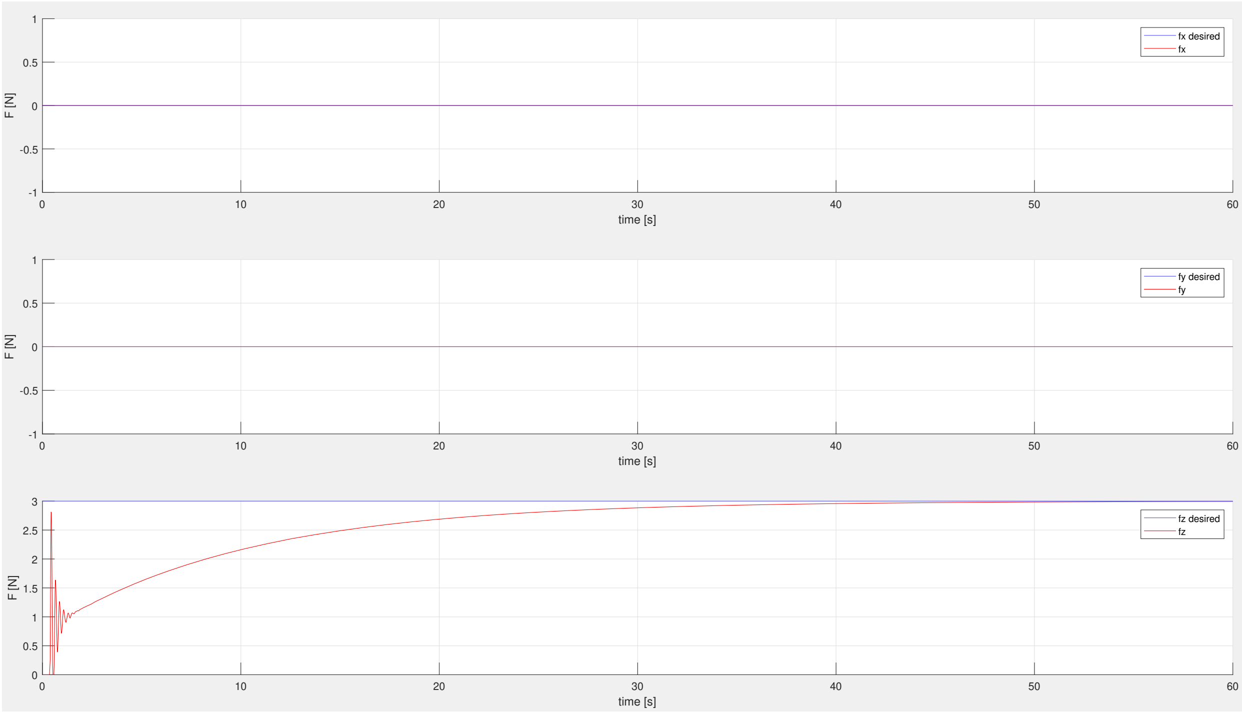

Figure 37: Force Control with Inner Position Loop - ee wrench

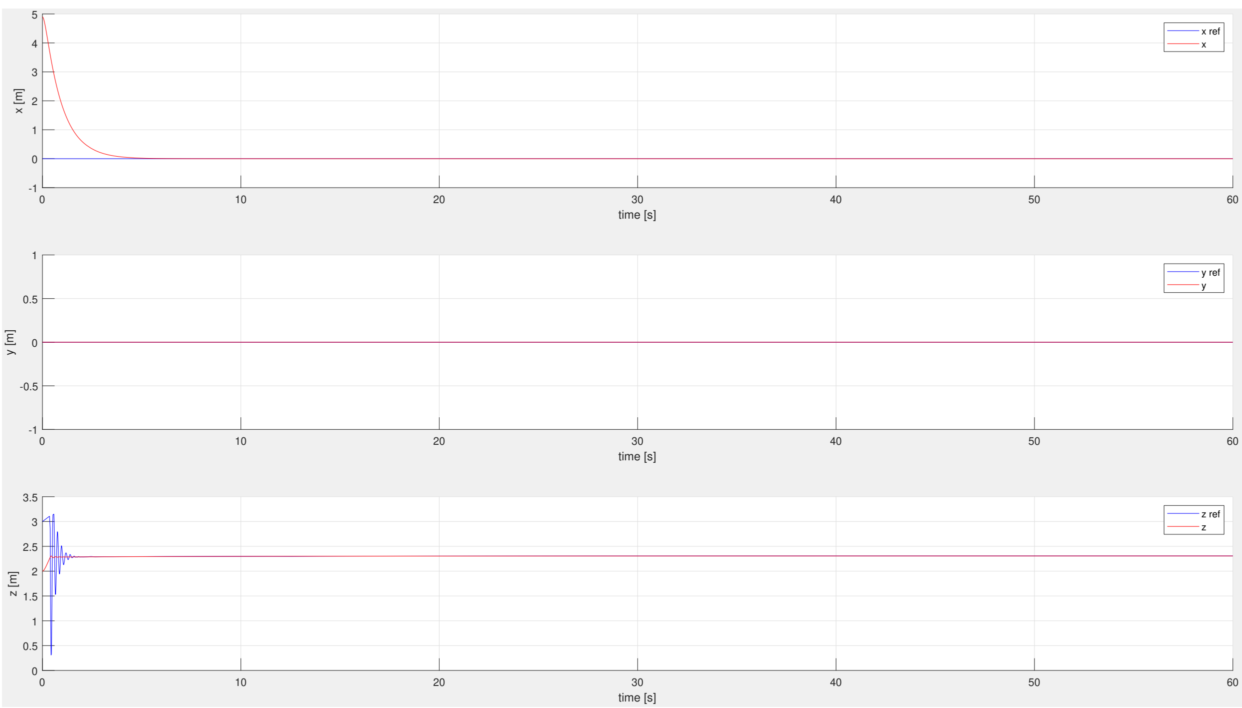

Figure 38: Force Control with Inner Position Loop - ee position

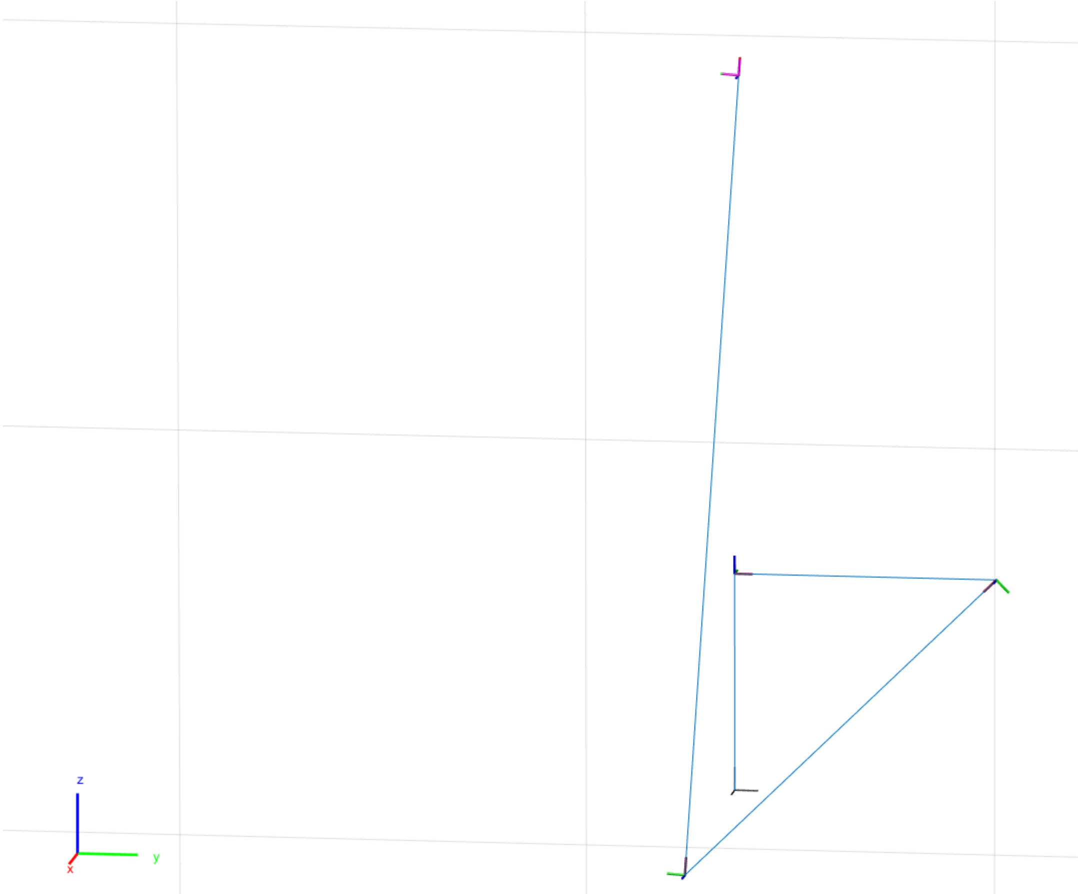

Figure 39: Force Control with Inner Position Loop - manipulator's final configuration

14 Homework 14

14.1 Assignment description

- Implement the Parallel Force/Position Control.

14.2 Considerations

The same considerations of the previous assignment also apply to this assignment.
The position of contact with the environment (Figure 34) is added to the PI force controller output as the reference for the internal position loop.

14.3 Force Control with Inner Position Loop

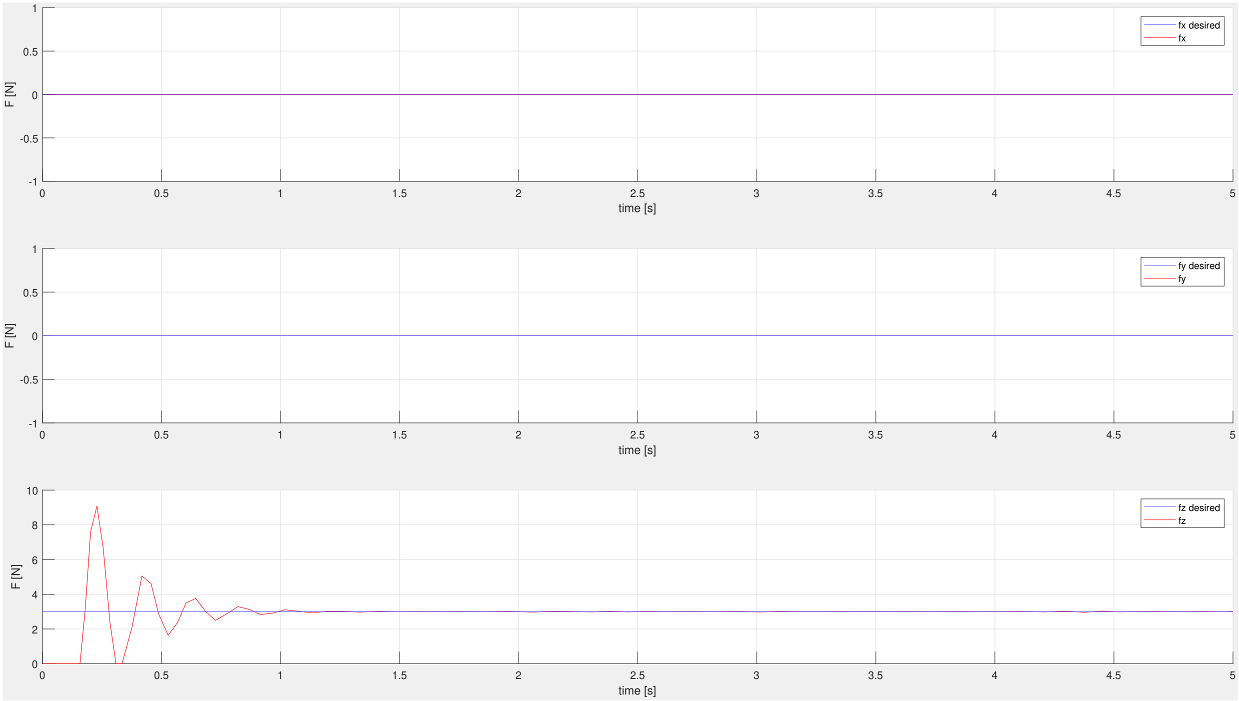

Figure 40: Parallel Force/Position Control - ee wrench

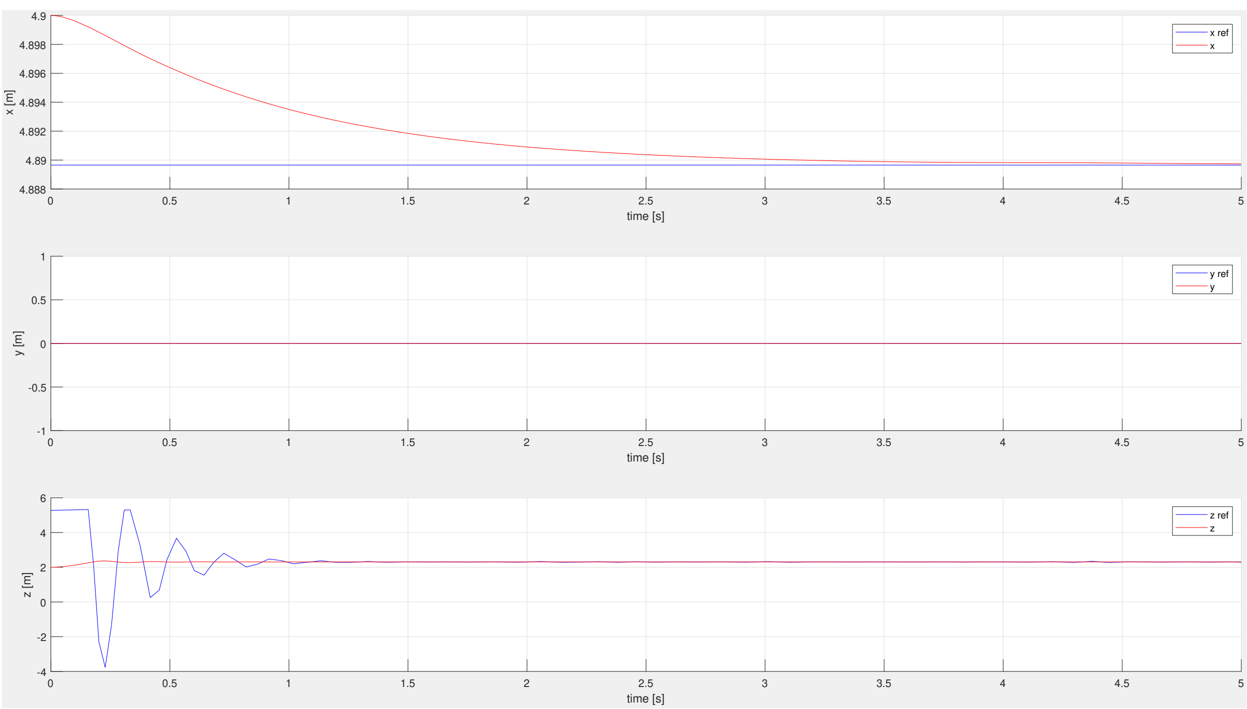

Figure 41: Parallel Force/Position Control - ee position

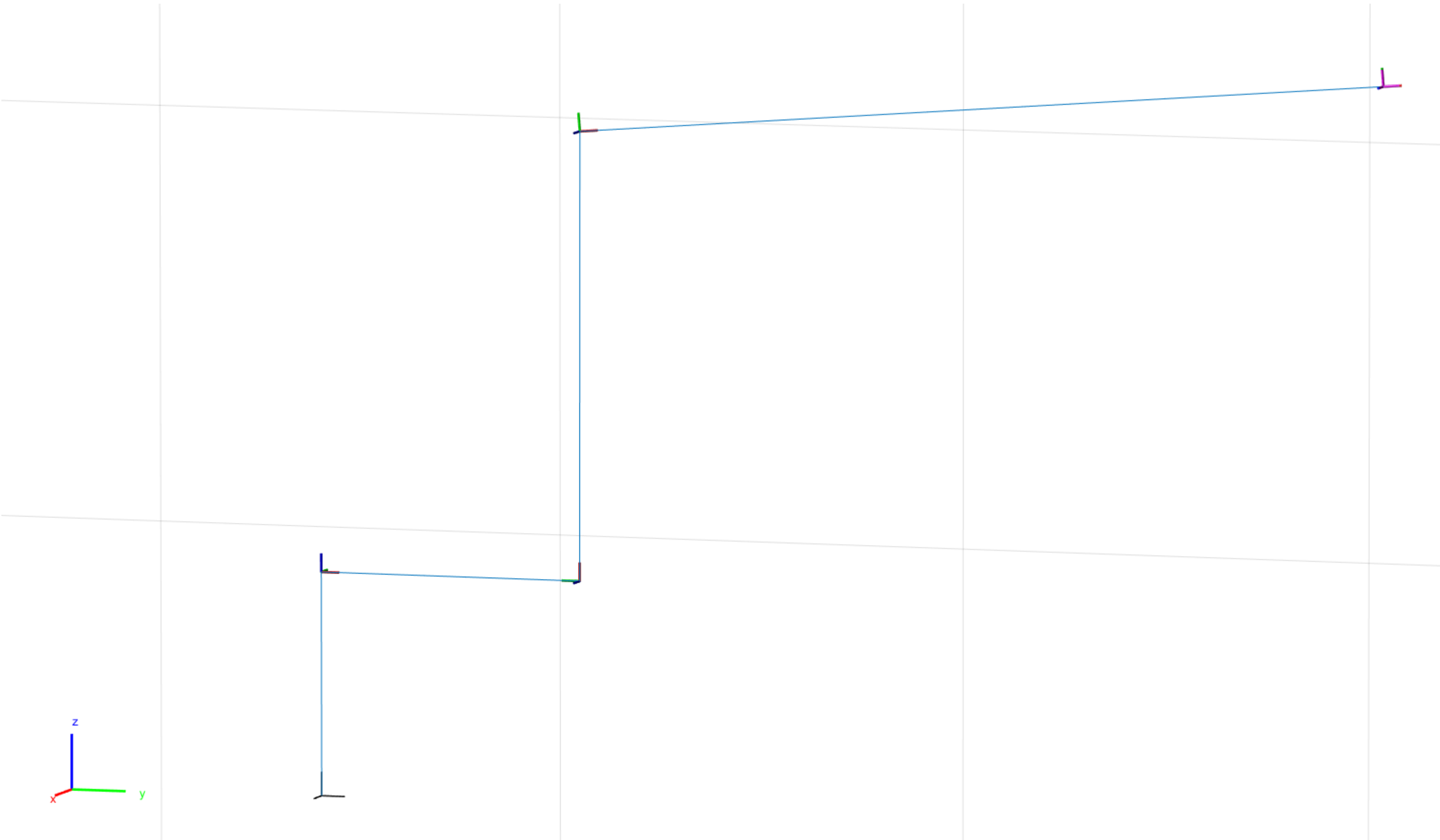

Figure 42: Parallel Force/Position Control - manipulator's final configuration

15 The S-function

15.0.1 State space representation

Assuming that $B(q)$ is not singular, the dynamical Lagrangian model of the manipulator can be rewritten in the state space representation $\dot{x} = f(x, u)$ by considering the state vector $x = [q, \dot{q}]$.
Since the manipulator has 3 joints, then $x \in \mathbb{R}^6$.
The matrices that appear in the lagrangian model, i.e. $B(q)$, $C(q, \dot{q})$ and $g(q)$, have been computed in the previous homeworks and stored into separate files that are called from the S-function.

15.0.2 Initial state

The initial state $x_0 = [q_0, \dot{q}_0]$ is set to 0, i.e. the manipulator starts in its home configuration and it's stationary.
The initial state can be changed if needed.

15.0.3 Goal

The goal of the S-function is to simulate the manipulator's continuos system state space representation.
The inputs are the current state x and the command torque u, while the output is x, i.e. the state.
In other words, we are asking: given the dynamical properties of the manipulator described by the $B(q)$, $C(q, \dot{q})$, and $g(q)$ matrices, given the current state x and input command u, what are the manipulator's joint positions and velocities at the next time instant?
però il sistema non è lineare, perchè dentro c'è g(q). quindi non abbiamo un sistema descritto da: dx = Ax + Bu Quindi con la S-function stiamo simulando un sistema non lineare?