Master's degree in Computer Engineering for Robotics and Smart Industry

Advanced control systems

Report on the assignments given during the 2022/2023 a.y.

Author: Lorenzo Busellato, VR472249
email: lorenzo.busellato_02@studenti.univr.it

UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Contents

1 Assignment 1 . . . **1**
1.1 Denavit-Hartenberg parameters . . . 1
1.2 Direct kinematics . . . 2
1.3 Inverse kinematics . . . 2
1.4 Analytical Jacobian . . . 3
1.5 Geometric Jacobian . . . 4
1.6 Relationship between JG and JA . . . 4

2 Assignment 2 . . . **5**
2.1 Compute the kinetic energy . . . 5
2.2 Compute the potential energy . . . 7

3 Assignment 3 . . . **8**
3.1 Equations of motion . . . 8

4 Assignment 4 . . . **10**
4.1 Compute the dynamic model using the recursive Newton-Euler formulation . . . 10

5 Assignment 5 . . . **11**
5.1 Compute the dynamic model in the operational space . . . 11

6 Assignment 6 . . . **12**
6.1 Design the joint space PD control law with gravity compensation . . . 12
6.2 What happens if g(q) is not taken into account? . . . 13
6.3 What happens if the gravity term is equal to g(q_d)? . . . 13
6.4 What happens if q_d is not constant? . . . 14

7 Assignment 7 . . . **15**
7.1 Design the joint space inverse dynamics control law . . . 15
7.2 Check the behaviour of the control law when the B, C and g values are different than the true ones . . . 17
7.3 What happens to the torque values when the settling time of the equivalent second order systems is chosen very small? . . . 18

8 Assignment 8 . . . **19**
8.1 Implement in Simulink the Adaptive Control law for the a 1-DoF link under gravity. . . . 19

9 Assignment 9 . . . **21**
9.1 Design the operational space PD control law with gravity compensation . . . 21

10 Assignment 10 . . . **22**
10.1 Design the operational space inverse dynamics control law . . . 22

11 Assignment 11 . . . **23**
11.1 Study the compliance control. Simulate the two “extreme” cases when the end-effector is interacting with the environment (considered as a planar surface) . . . 23
11.2 $K_{env} \ll K_P$ . . . 24
11.3 $K_{env} = K_P$ . . . 25
11.4 $K_{env} \gg K_P$ . . . 26

12 Assignment 12 . . . **27**
12.1 Implement the impedance control in the operational space . . . 27

13 Assignment 13 . . . **29**
13.1 Implement the admittance control in the operational space . . . 29
13.2 $K_{env} \ll K_P$ . . . 30
13.3 $K_{env} = K_P$ . . . 31
13.4 $K_{env} \gg K_P$ . . . 32

14 Assignment 14 . . . **33**
14.1 Implement the force control with inner position loop . . . 33

15 Assignment 15 . . . . **35**
15.1 Implement the parallel force/position control 35

1 Assignment 1

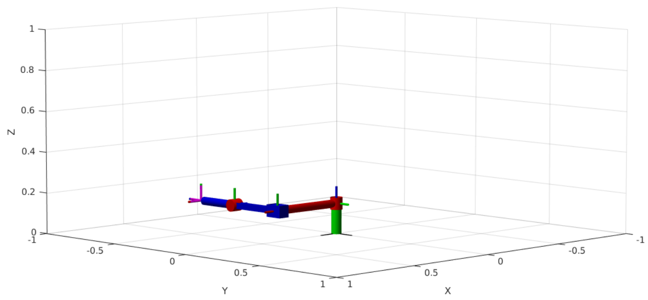


Figure 1: Visualization of the URDF of the PRP robot in the home configuration ($q_1 = 0, q_2 = 0, \theta_3 = 0$)

1.1 Denavit-Hartenberg parameters

First of all, reference frames were assigned to each joint according to the DH convention:

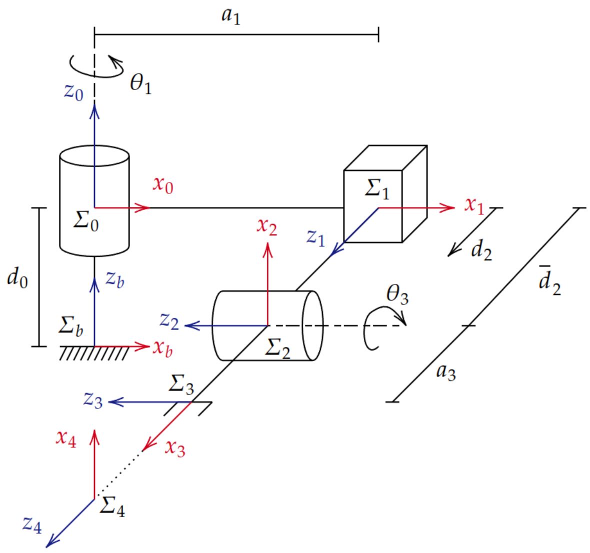


Figure 2: Reference frames assigned with the DH convention

Note that the DH frames do not correspond with the frames of the URDF model.
Then the DH parameter table was populated:

	a_i	α_i	d_i	θ_i
0	0	0	d_0	0
1	a_1	$\frac{\pi}{2}$	0	$q_1 = \theta_1$
2	0	$-\frac{\pi}{2}$	$(q_2 = d_2) + \overline{d}_2$	$\frac{\pi}{2}$
3	a_3	0	0	$(q_3 = \theta_3) - \frac{\pi}{2}$
4	0	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$

The first row is the fixed offset in the z_b direction between world and frame Σ_0, while the last row is the fixed rotation that aligns the z axis of the end-effector to the approach direction.

Using the DH parameters, the transformation matrices between the frames were then computed. The transformation matrix between frame i and frame $i-1$ is in the form:

$$T_{i-1}^{i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The computed transformation matrices are then:

$$T_b^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_0^1 = \begin{bmatrix} C_1 & 0 & S_1 & a_1C_1 \\ S_1 & 0 & -C_1 & a_1S_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_1^2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & q_2+\overline{d}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} C\left(q_3-\frac{\pi}{2}\right) & -S\left(q_3-\frac{\pi}{2}\right) & 0 & a_3C\left(q_3-\frac{\pi}{2}\right) \\ S\left(q_3-\frac{\pi}{2}\right) & C\left(q_3-\frac{\pi}{2}\right) & 0 & a_3S\left(q_3-\frac{\pi}{2}\right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_3 & C_3 & 0 & a_3S_3 \\ -C_3 & S_3 & 0 & -a_3C_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_3^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where C_i and S_i denote respectively $\cos(q_i)$ and $\sin(q_i)$.

1.2 Direct kinematics

The direct kinematics of the manipulator are obtained by chaining the above transformation matrices, thus obtaining a global transformation matrix from the end-effector to the base of the manipulator:

$$T_b^4 = T_b^0T_0^1T_1^2T_2^3T_3^4 = \underbrace{\begin{bmatrix} C_1 & 0 & S_1 & a_1C_1 \\ S_1 & 0 & -C_1 & a_1S_1 \\ 0 & 1 & 0 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{T_b^1} T_1^2T_2^3T_3^4 = \underbrace{\begin{bmatrix} 0 & -S_1 & -C_1 & (q_2+\overline{d}_2)S_1 + a_1C_1 \\ 0 & C_1 & -S_1 & -(q_2+\overline{d}_2)C_1 + a_1S_1 \\ 1 & 0 & 0 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{T_b^2} T_2^3T_3^4$$

$$= \underbrace{\begin{bmatrix} S_1C_3 & -S_1S_3 & -C_1 & (q_2+\overline{d}_2)S_1 + a_1C_1 + a_3S_1C_3 \\ -C_1C_3 & C_1S_3 & -S_1 & -(q_2+\overline{d}_2)C_1 + a_1S_1 - a_3C_1C_3 \\ S_3 & C_3 & 0 & d_0 + a_3S_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{T_b^3} T_3^4$$

$$= \begin{bmatrix} -S_1S_3 & -C_1 & S_1C_3 & (q_2+\overline{d}_2)S_1 + a_1C_1 + a_3S_1C_3 \\ C_1S_3 & -S_1 & -C_1C_3 & -(q_2+\overline{d}_2)C_1 + a_1S_1 - a_3C_1C_3 \\ C_3 & 0 & S_3 & d_0 + a_3S_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_b^4 & p_4 \\ \overline{0} & 1 \end{bmatrix}$$

1.3 Inverse kinematics

From the direct kinematics, the position of the end-effector is given by:

$$p_4 = \begin{pmatrix} p_{4x} \\ p_{4y} \\ p_{4z} \end{pmatrix} = \begin{pmatrix} (q_2+\overline{d}_2)S_1 + a_1C_1 + a_3S_1C_3 \\ -(q_2+\overline{d}_2)C_1 + a_1S_1 - a_3C_1C_3 \\ d_0 + a_3S_3 \end{pmatrix}$$

Therefore an expression for q_3 can immediately be derived:

$$p_{4z} = d_0 + a_3S_3 \implies S_3 = \frac{p_{4z}-d_0}{a_3}, C_3 = \pm\sqrt{1-S_3^2} = \pm\sqrt{\frac{a_3^2-(p_{4z}-d_0)^2}{a_3^2}} \implies q_3^{\pm} = \operatorname{atan2}(S_3, \pm C_3)$$

q_2 is determined by applying summing and squaring to the position of the origin of frame Σ_2:

$$p_2 = \begin{pmatrix} p_{2x} \\ p_{2y} \\ p_{2z} \end{pmatrix} = \begin{pmatrix} (q_2+\overline{d}_2)S_1 + a_1C_1 \\ -(q_2+\overline{d}_2)C_1 + a_1S_1 \\ d_0 \end{pmatrix}$$

$$
\begin{aligned}
p_{2x}^2 + p_{2y}^2 &= S_1^2(q_2+\bar{d}_2)^2 + a_1^2C_1^2 + \cancel{2a_1(q_2+\bar{d}_2)S_1C_1} + C_1^2(q_2+\bar{d}_2)^2 + a_1^2S_1^2 - \cancel{2a_1(q_2+\bar{d}_2)S_1C_1} = \\
&= (q_2+\bar{d}_2)^2 + a_1^2
\end{aligned}
$$

Therefore q_2 is given by the solution of the quadratic equation:

$$q_2^2 + 2\bar{d}_2 q_2 + a_2 - p_{2x}^2 - p_{2y}^2 = 0$$

which is:

$$q_{2,12} = -\frac{\cancel{2}\bar{d}_2 \pm \sqrt{\cancel{4}\bar{d}_2^{\cancel{2}} - \cancel{4}(a_1^2 + \cancel{\bar{d}_2} - p_{2x}^2 - p_{2y}^2)}}{\cancel{2}} = -\bar{d}_2 \pm \sqrt{p_{2x}^2 + p_{2y}^2 - a_1^2}$$

To choose the solution, both are computed and the correct one will be the one that satisfies joint limits.
To choose the correct sign for q_3, q_2 is recomputed by summing and squaring the x and y components of p_3:

$$
\begin{aligned}
p_{3x}^2 + p_{3y}^2 &= S_1^2(q_2+\bar{d}_2)^2 + a_1^2C_1^2 + a_3^2S_1^2C_3^2 + \cancel{2a_1(q_2+\bar{d}_2)S_1C_1} + 2a_3(q_2+\bar{d}_2)S_1^2C_3 + \cancel{2a_1a_3S_1C_1C_3} \\
&+ C_1^2(q_2+\bar{d}_2)^2 + a_1^2S_1^2 + a_3^2C_1^2C_3^2 - \cancel{2a_1(q_2+\bar{d}_2)S_1C_1} + 2a_3(q_2+\bar{d}_2)C_1^2C_3 - \cancel{2a_1a_3S_1C_1C_3} = \\
&= (q_2+\bar{d}_2)^2 + a_1^2 + a_3^2C_3^2 + 2a_3(q_2+\bar{d}_2)C_3
\end{aligned}
$$

and therefore:

$$q_{2,12} = -a_3C_3 - \bar{d}_2 \pm \sqrt{p_{4x}^2 + p_{4y}^2 - a_1^2}$$

By computing the four possible solutions and checking which one is equal to the one obtained previously, it is possible to determine the correct sign for C_3 and therefore the correct value for q_3.

Finally, q_1 is determined by solving the system of equations:

$$\begin{cases} p_{2x} = (q_2+\bar{d}_2)S_1 + a_1C_1 \\ p_{2y} = -(q_2+\bar{d}_2)C_1 + a_1S_1 \end{cases}$$

in the unknowns C_1 and S_1. The system yields:

$$C_1 = \frac{a_1p_{2x} - (q_2+\bar{d}_2)p_{2y}}{(q_2+\bar{d}_2)^2 + a_1^2}, S_1 = \frac{p_{2x} - a_1C_1}{(q_2+\bar{d}_2)} \implies q_1 = \operatorname{atan2}(S_1, C_1)$$

For the orientation o_4 of the end-effector, the angles α, β and γ can be derived by equating the rotation matrix R_b^4 to the rotation matrix that expresses a ZXZ Euler angle rotation:

$$\begin{bmatrix} -S_1S_3 & -C_1 & S_1C_3 \\ C_1S_3 & -S_1 & -C_1C_3 \\ C_3 & 0 & S_3 \end{bmatrix} = \begin{bmatrix} C_\alpha C_\gamma - C_\beta S_\alpha S_\gamma & -C_\alpha S_\gamma - C_\beta C_\gamma S_\alpha & S_\alpha S_\beta \\ S_\alpha C_\gamma + C_\beta C_\alpha S_\gamma & -S_\alpha S_\gamma - C_\beta C_\gamma C_\alpha & -C_\alpha S_\beta \\ S_\beta S_\gamma & C_\gamma S_\beta & C_\beta \end{bmatrix}$$

$$\begin{cases} S_\alpha S_\beta = S_1C_3, -C_\alpha S_\beta = -C_1C_3 \implies S_\alpha = S_1, C_\alpha = C_1 & \implies \alpha = q_1 \\ C_\beta = S_3 & \implies \beta = \frac{\pi}{2} - q_3 \\ S_\beta S_\gamma = C_3 \implies S_\gamma = 1 & \implies \gamma = \frac{\pi}{2} \end{cases} \implies o_4 = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} q_1 \\ \frac{\pi}{2} - q_3 \\ \frac{\pi}{2} \end{pmatrix}$$

1.4 Analytical Jacobian

The lines of the analytical Jacobian are the gradients of the pose (position and orientation) of the end-effector with respect to the joint variables:

$$
\begin{aligned}
\nabla p_x &= \begin{bmatrix} \frac{\partial p_x}{\partial q_1} & \frac{\partial p_x}{\partial q_2} & \frac{\partial p_x}{\partial q_3} \end{bmatrix} = \begin{bmatrix} (q_2+\bar{d}_2)C_1 - a_1S_1 + a_3C_3C_1 & S_1 & -a_3S_1S_3 \end{bmatrix} \\
\nabla p_y &= \begin{bmatrix} \frac{\partial p_y}{\partial q_1} & \frac{\partial p_y}{\partial q_2} & \frac{\partial p_y}{\partial q_3} \end{bmatrix} = \begin{bmatrix} (q_2+\bar{d}_2)S_1 + a_1C_1 + a_3S_1C_3 & -C_1 & a_3S_1S_3 \end{bmatrix} \\
\nabla p_z &= \begin{bmatrix} \frac{\partial p_z}{\partial q_1} & \frac{\partial p_z}{\partial q_2} & \frac{\partial p_z}{\partial q_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & a_3C_3 \end{bmatrix} \\
\nabla \alpha &= \begin{bmatrix} \frac{\partial \alpha}{\partial q_1} & \frac{\partial \alpha}{\partial q_2} & \frac{\partial \alpha}{\partial q_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\
\nabla \beta &= \begin{bmatrix} \frac{\partial \beta}{\partial q_1} & \frac{\partial \alpha}{\partial q_2} & \frac{\partial \beta}{\partial q_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} \\
\nabla \gamma &= \begin{bmatrix} \frac{\partial \gamma}{\partial q_1} & \frac{\partial \gamma}{\partial q_2} & \frac{\partial \gamma}{\partial q_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

Therefore the analytical Jacobian is:

$$J_A = \begin{bmatrix} \nabla p_x \\ \nabla p_y \\ \nabla p_z \\ \nabla \alpha \\ \nabla \beta \\ \nabla \gamma \end{bmatrix} = \begin{bmatrix} (q_2+\overline{d}_2)C_1 - a_1S_1 + a_3C_3C_1 & S_1 & -a_3S_1S_3 \\ (q_2+\overline{d}_2)S_1 + a_1C_1 + a_3S_1C_3 & -C_1 & a_3C_1S_3 \\ 0 & 0 & a_3C_3 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \implies \begin{bmatrix} \dot{p}_4 \\ \dot{o}_4 \end{bmatrix} = J_A\dot{q}$$

where $\dot{q} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^T$.

1.5 Geometric Jacobian

The i-th column of the geometric Jacobian ($i = 0, \ldots, n-1$) is constructed as follows:

$$J_{Gi} = \begin{bmatrix} z_i \times (p_4 - p_i) \\ z_i \end{bmatrix} \quad \text{(revolute joint)} \qquad J_{Gi} = \begin{bmatrix} z_i \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{(prismatic joint)}$$

where p_4 is the position of the end-effector, p_i is the position of frame i and z_i is the direction of the z axis of frame i, all with respect to the base frame.

So:

$$p_2 = \begin{bmatrix} S_1(q_2+\overline{d}_2) + a_1C_1 \\ -C_1(q_2+\overline{d}_2) + a_1S_1 \\ d_0 \end{bmatrix}, z_2 = \begin{bmatrix} -C_1 \\ -S_1 \\ 0 \end{bmatrix} \implies J_{G2} = \begin{bmatrix} -a_3S_1S_3 & a_3C_1C_3 & a_3C_3 & -C_1 & -S_1 & 0 \end{bmatrix}^T$$

$$p_1 = \begin{bmatrix} a_1C_1 \\ a_1S_1 \\ d_0 \end{bmatrix}, z_1 = \begin{bmatrix} S_1 \\ -C_1 \\ 0 \end{bmatrix} \implies J_{G1} = \begin{bmatrix} S_1 & -C_1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$p_0 = \begin{bmatrix} 0 \\ 0 \\ d_0 \end{bmatrix}, z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \implies J_{G0} = \begin{bmatrix} C_1(q_2+\overline{d}_2) - a_1S_1 + a_3C_1C_3 & S_1(q_2+\overline{d}_2) + a_1C_1 + a_3S_1C_3 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

Finally:

$$J_G = \begin{bmatrix} C_1(q_2+\overline{d}_2) - a_1S_1 + a_3C_1C_3 & S_1 & -a_3S_1S_3 \\ S_1(q_2+\overline{d}_2) + a_1C_1 + a_3S_1C_3 & -C_1 & a_3C_1C_3 \\ 0 & 0 & a_3C_3 \\ 0 & 0 & -C_1 \\ 0 & 0 & -S_1 \\ 1 & 0 & 0 \end{bmatrix} \implies \begin{bmatrix} \dot{p}_3 \\ \omega_3 \end{bmatrix} = J_G\dot{q}$$

As expected, the rows of the geometric Jacobian related to linear velocity are the same ones found in the analytical Jacobian.

1.6 Relationship between JG and JA

The geometric and analytical Jacobians are related by the following relationship:

$$J_G = T_A(\Phi)J_A = \begin{bmatrix} I & 0 \\ 0 & T(\Phi) \end{bmatrix} J_A$$

Therefore:

$$\begin{bmatrix} 0 & 0 & -C_1 \\ 0 & 0 & -S_1 \\ 1 & 0 & 0 \end{bmatrix} = T(\Phi) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \implies T(\Phi) = \begin{bmatrix} 0 & 0 & -C_1 \\ 0 & 0 & -S_1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}^\dagger = \begin{bmatrix} 0 & C_1 & 0 \\ 0 & S_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

where † denotes the Moore-Penrose pseudoinverse.

2 Assignment 2

To compute both the kinetic and potential energy, the positions of the centres of mass of each link are needed. With respect to each frame Σ_i attached to link i:

$$p^1_{l_1} = \begin{bmatrix} -\frac{h_1}{2} \\ 0 \\ 0 \end{bmatrix} \qquad p^2_{l_2} = \begin{bmatrix} 0 \\ \frac{a_2}{2} \\ 0 \end{bmatrix} \qquad p^3_{l_3} = \begin{bmatrix} -\frac{h_3}{2} \\ 0 \\ 0 \end{bmatrix}$$

Using the transformation matrices obtained with direct kinematics, the positions of the centres of mass with respect to the base frame Σ_b are:

$$p_{l_1} = R^1_b p^1_{l_1} + p_1 = \begin{bmatrix} \left(a_1 - \frac{h_1}{2}\right) C_1 \\ \left(a_1 - \frac{h_1}{2}\right) S_1 \\ d_0 \end{bmatrix} \qquad p_{l_2} = R^2_b p^2_{l_2} + p_2 = \begin{bmatrix} \left(q_2 + \overline{d}_2 - \frac{a_2}{2}\right) S_1 + a_1 C_1 \\ -\left(q_2 + \overline{d}_2 - \frac{a_2}{2}\right) C_1 + a_1 S_1 \\ d_0 \end{bmatrix}$$

$$p_{l_3} = R^3_b p^3_{l_3} + p_3 = \begin{bmatrix} (q_2 + \overline{d}_2) S_1 + a_1 C_1 + \left(a_3 - \frac{h_3}{2}\right) S_1 C_3 \\ -(q_2 + \overline{d}_2) C_1 + a_1 S_1 - \left(a_3 - \frac{h_3}{2}\right) C_1 C_3 \\ d_0 + \left(a_3 - \frac{h_3}{2}\right) S_3 \end{bmatrix}$$

2.1 Compute the kinetic energy

The total kinetic energy for an open-chain manipulator is:

$$\mathcal{T}(q, \dot{q}) = \frac{1}{2} \dot{q}^T B(q) \dot{q} \qquad B(q) = \sum_{i=1}^{n} (m_{l_i} (J_P^{l_i})^T J_P^{l_i} + (J_O^{l_i})^T R^i_b I^i_{l_i} R^{i\,T}_b J_O^{l_i})$$

where $B(q)$ is the inertia matrix, $I^i_{l_i}$ are the inertia tensors with respect to Σ_i, $J_P^{l_i}$ and $J_O^{l_i}$ are the linear and angular partial Jacobian matrices and R^i_b are the rotation matrices that bring frame Σ_i to frame Σ_b.

The inertia tensors are obtained with Steiner's theorem:

$$\begin{aligned}
I^1_{l_1} &= I^{C_1}_{l_1} + m_{l_1} S^T(p^1_{l_1}) S(p^1_{l_1}) \\
&= m_{l_1} \begin{bmatrix} \frac{1}{2}(r_1^2) & 0 & 0 \\ 0 & \frac{1}{2}(3r_1^2 + h_1^2) & 0 \\ 0 & 0 & \frac{1}{2}(3r_1^2 + h_1^2) \end{bmatrix} + m_{l_1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{h_1^2}{4} & 0 \\ 0 & 0 & \frac{h_1^2}{4} \end{bmatrix} \\
&= \frac{1}{2} m_{l_1} \begin{bmatrix} r_1^2 & 0 & 0 \\ 0 & 3r_1^2 + \frac{3h_1^2}{2} & 0 \\ 0 & 0 & 3r_1^2 + \frac{3h_1^2}{2} \end{bmatrix} \\
I^2_{l_2} &= I^{C_2}_{l_2} + m_{l_2} S^T(p^2_{l_2}) S(p^2_{l_2}) \\
&= m_{l_2} \begin{bmatrix} \frac{1}{12}(b_2^2 + c_2^2) & 0 & 0 \\ 0 & \frac{1}{12}(a_2^2 + c_2^2) & 0 \\ 0 & 0 & \frac{1}{12}(a_2^2 + b_2^2) \end{bmatrix} + m_{l_2} \begin{bmatrix} \frac{a_2^2}{4} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{a_2^2}{4} \end{bmatrix} \\
&= \frac{1}{12} m_{l_2} \begin{bmatrix} 3a_2^2 + b_2^2 + c_2^2 & 0 & 0 \\ 0 & a_2^2 + c_2^2 & 0 \\ 0 & 0 & 4a_2^2 + b_2^2 \end{bmatrix} \\
I^3_{l_3} &= I^{C_3}_{l_3} + m_{l_3} S^T(p^3_{l_3}) S(p^3_{l_3}) \\
&= m_{l_3} \begin{bmatrix} \frac{1}{2}(r_3^2) & 0 & 0 \\ 0 & \frac{1}{2}(3r_3^2 + h_3^2) & 0 \\ 0 & 0 & \frac{1}{2}(3r_3^2 + h_3^2) \end{bmatrix} + m_{l_3} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{h_3^2}{4} & 0 \\ 0 & 0 & \frac{h_3^2}{4} \end{bmatrix} \\
&= \frac{1}{2} m_{l_3} \begin{bmatrix} r_3^2 & 0 & 0 \\ 0 & 3r_3^2 + \frac{3}{2}h_3^2 & 0 \\ 0 & 0 & 3r_3^2 + \frac{3}{2}h_3^2 \end{bmatrix}
\end{aligned}$$

The links are assumed to be made up of an homogeneous material, specifically aluminium. Therefore the mass m_{l_i} of link i is $m_{l_i} = \rho V_{l_i}$, where $\rho = 2710\ kg/m^3$ is the density of aluminium and V_{l_i} is the volume of link i.

The partial Jacobian matrices are constructed as follows:

$$J_P^{l_i} = \begin{bmatrix} j_{P1}^{l_i} & \dots & j_{Pj}^{l_i} & 0 & \dots & 0 \end{bmatrix} \text{ where } j_{Pj}^{l_i} = \begin{cases} z_{j-1} & \text{prismatic joint} \\ z_{j-1} \times (p_{l_i} - p_{j-1}) & \text{revolute joint} \end{cases}$$

$$J_O^{l_i} = \begin{bmatrix} j_{O1}^{l_i} & \dots & j_{Oj}^{l_i} & 0 & \dots & 0 \end{bmatrix} \text{ where } j_{Oj}^{l_i} = \begin{cases} 0 & \text{prismatic joint} \\ z_{j-1} & \text{revolute joint} \end{cases}$$

where p_{j-1} is the position vector of the origin of frame Σ_{j-1} and z_{j-1} is the unit vector of axis z of frame Σ_{j-1}, all with respect of Σ_b.

$$J_P^{l_1} = \begin{bmatrix} j_{P1}^{l_1} & 0 & 0 \end{bmatrix} = \begin{bmatrix} -S_1\left(a_1 - \frac{h_1}{2}\right) & 0 & 0 \\ C_1\left(a_1 - \frac{h_1}{2}\right) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$j_{P1}^{l_1} = z_0 \times (p_{l_1} - p_0) = \begin{bmatrix} -S_1\left(a_1 - \frac{h_1}{2}\right) & C_1\left(a_1 - \frac{h_1}{2}\right) & 0 \end{bmatrix}^T$$

$$J_O^{l_1} = \begin{bmatrix} j_{O1}^{l_1} & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$j_{O1}^{l_1} = z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$J_P^{l_2} = \begin{bmatrix} j_{P1}^{l_2} & j_{P2}^{l_2} & 0 \end{bmatrix} = \begin{bmatrix} \left(d_2 + q_2 - \frac{a_2}{2}\right) C_1 & S_1 & 0 \\ \left(d_2 + q_2 - \frac{a_2}{2}\right) S_1 & -C_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$j_{P1}^{l_2} = z_0 \times (p_{l_2} - p_0) = \begin{bmatrix} \left(d_2 + q_2 - \frac{a_2}{2}\right) C_1 & \left(d_2 + q_2 - \frac{a_2}{2}\right) S_1 & 0 \end{bmatrix}^T$$

$$j_{P2}^{l_2} = z_1 = \begin{bmatrix} S_1 & -C_1 & 0 \end{bmatrix}^T$$

$$J_O^{l_2} = \begin{bmatrix} j_{O1}^{l_2} & j_{O2}^{l_2} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$j_{O1}^{l_2} = z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$j_{O2}^{l_2} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

$$J_P^{l_3} = \begin{bmatrix} j_{P1}^{l_3} & j_{P2}^{l_3} & j_{P3}^{l_3} \end{bmatrix} = \begin{bmatrix} C_1 C_3\left(a_3 - \frac{h_3}{2}\right) & S_1 & S_1 S_3\left(a_3 - \frac{h_3}{2}\right) \\ S_1 C_3\left(a_3 - \frac{h_3}{2}\right) & -C_1 & -C_1 S_3\left(a_3 - \frac{h_3}{2}\right) \\ 0 & 0 & C_3\left(a_3 - \frac{h_3}{2}\right) \end{bmatrix}$$

$$j_{P1}^{l_3} = z_0 \times (p_{l_3} - p_0) = \begin{bmatrix} C_1 C_3\left(a_3 - \frac{h_3}{2}\right) & S_1 C_3\left(a_3 - \frac{h_3}{2}\right) & 0 \end{bmatrix}^T$$

$$j_{P2}^{l_3} = z_1 = \begin{bmatrix} S_1 & -C_1 & 0 \end{bmatrix}^T$$

$$j_{P3}^{l_3} = z_2 \times (p_{l_3} - p_2) = \begin{bmatrix} S_1 S_3\left(a_3 - \frac{h_3}{2}\right) & -C_1 S_3\left(a_3 - \frac{h_3}{2}\right) & C_3\left(a_3 - \frac{h_3}{2}\right) \end{bmatrix}^T$$

$$J_O^{l_3} = \begin{bmatrix} j_{O1}^{l_3} & j_{O2}^{l_3} & j_{O3}^{l_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -C_1 \\ 0 & 0 & -S_1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$j_{O1}^{l_3} = z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$j_{O2}^{l_3} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

$$j_{O3}^{l_3} = z_2 = \begin{bmatrix} -C_1 & -S_1 & 0 \end{bmatrix}^T$$

So the inertial matrices of each joint are:

$$\begin{aligned} B_1(q) &= m_{l_1} (J_P^{l_1})^T J_P^{l_1} + (J_O^{l_1})^T R_b^1 I_{l_1}^1 {R_b^1}^T J_O^{l_1} \\ &= m_{l_1} \begin{bmatrix} \frac{1}{2}((a_1 - h_1)^2 + r_1^2) & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} \\ B_2(q) &= m_{l_2} (J_P^{l_2})^T J_P^{l_2} + (J_O^{l_2})^T R_b^2 I_{l_2}^2 {R_b^2}^T J_O^{l_2} \\ &= m_{l_2} \begin{bmatrix} a_1^2 + \left(d_2 + q_2 - \frac{1}{2} a_2\right)^2 + \frac{1}{12}\left(a_2^2 + c_2^2\right) & -a_1 & 0 \\ * & 1 & 0 \\ * & * & 0 \end{bmatrix} \\ B_3(q) &= m_{l_3} (J_P^{l_3})^T J_P^{l_3} + (J_O^{l_3})^T R_b^3 I_{l_3}^3 {R_b^3}^T J_O^{l_3} \\ &= m_{l_3} \begin{bmatrix} a_1^2 + \left(d_2 + q_2 - \left(\frac{1}{2} h_3 - a_3\right) C_3\right)^2 + \frac{1}{12}(h_3^2 + r_3^2) & -a_1 & \frac{1}{2} a_1 (2a_3 - h_3) S_3 \\ * & 1 & -\frac{1}{2}(2a_3 - h_3) S_3 \\ * & * & a_3^2 - a_3 h_3 + h_3^2 + \frac{1}{2} r_3^2 \end{bmatrix} \end{aligned}$$

So the overall inertia matrix is:

$$
\begin{aligned}
B(q) &= B_1(q) + B_2(q) + B_3(q) \\
&= \begin{bmatrix} K & -a_1(m_{l_2}+m_{l_3}) & \frac{1}{2}m_{l_3}a_1(2a_3-h_3)S_3 \\ * & m_{l_2}+m_{l_3} & -\frac{1}{2}m_{l_3}(2a_3-h_3)S_3 \\ * & * & a_3^2 - a_3h_3 + h_3^2 + \frac{1}{2}r_3^2 \end{bmatrix} \\
&= \begin{bmatrix} 0.4904q_2 + 0.05885C_3 + 0.0474C_3^2 + 0.1962q_2C_3 + 0.8173q_2^2 + 0.4796 & -0.6196 & 0.03923S_3 \\ * & 1.549 & -0.09808S_3 \\ * & * & 0.04757 \end{bmatrix}
\end{aligned}
$$

with :

$$
\begin{aligned}
K = \frac{1}{2}m_{l_1}\left((a_1-h_1)^2 + r_1^2\right) + m_{l_2}\left(a_1^2 + \left(d_2 + q_2 - \frac{1}{2}a_2\right)^2 + \frac{1}{12}\left(a_2^2 + c_2^2\right)\right) \\
+ m_{l_3}\left(a_1^2 + \left(d_2 + q_2 - \left(\frac{1}{2}h_3 - a_3\right)C_3\right)^2 + \frac{1}{12}(h_3^2 + r_3^2)\right)
\end{aligned}
$$

Finally, the kinetic energy is given by:

$$
\begin{aligned}
\mathcal{T}(q,\dot{q}) = \frac{1}{2}\dot{q}^T B(q)\dot{q} &= 0.0237\dot{q}_1^2C_3^2 - 0.6196\dot{q}_1\dot{q}_2 + 0.3549\dot{q}_1^2q_2 + 0.248\dot{q}_1^2 + 0.7745\dot{q}_2^2 + 0.02378\dot{q}_3^2 \\
&+ 0.7745\dot{q}_1^2q_2^2 + 0.02942\dot{q}_1^2C_3 + 0.03923\dot{q}_1\dot{q}_3S_3 - 0.09808\dot{q}_2\dot{q}_3S_3 + 0.09808\dot{q}_1^2q_2C_3
\end{aligned}
$$

The symbolic expression is not reported for space reasons.

2.2 Compute the potential energy

The potential energy is given by:

$$
\mathcal{U}(q) = -\sum_{i=1}^{n} m_{l_i} g_0^T p_{l_i}
$$

where $g_0 = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$ is the gravity acceleration vector in the base frame Σ_b.
So:

$$
\begin{aligned}
\mathcal{U}_1 &= -m_{l_1} g d_0 \\
\mathcal{U}_2 &= -m_{l_2} g d_0 \\
\mathcal{U}_3(q) &= -m_{l_3} g \left(d_0 + \left(a_3 - \frac{h_3}{2}\right)S_3\right)
\end{aligned}
$$

Finally:

$$
\mathcal{U}(q) = -(\mathcal{U}_1 + \mathcal{U}_2 + \mathcal{U}_3(q)) = \left[(m_{l_1} + m_{l_2} + m_{l_3})d_0 + m_{l_3}\left(a_3 - \frac{h_3}{2}\right)S_3\right]g = 0.9621S_3 + 4.033
$$

3 Assignment 3

3.1 Equations of motion

The equations of motion for an open chain robotic manipulator are:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + F_s sign(\dot{q}) + g(q) = \tau - J^T(q)h_e$$

Ignoring the contributions related to frictions and the external wrench (F_v, F_s, he), the equations reduce to:

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

where τ is the command torque, $C(q,\dot{q})$ is the Coriolis matrix and $g(q)$ is the gravity term, which is given by:

$$g_i(q) = -\sum_{j=1}^{n} m_{l_i} g_0^T j_{P_i}^{l_j}(q) \rightarrow g(q) = \begin{bmatrix} 0 \\ 0 \\ m_{l_3} g\left(a_3 - \frac{1}{2}h_3\right) C_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -0.9621 C_3 \end{bmatrix}$$

The c_{ij} elements of $C(q,\dot{q})$ are:

$$c_{ij} = \sum_{k=1}^{n} \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k$$

where b_{ij}, b_{ik} and b_{jk} are the elements of the inertial matrix $B(q)$. The derivatives of the $B(q)$ matrix are:

$$\frac{\partial B}{\partial q_1} = \begin{bmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix}$$

$$\frac{\partial B}{\partial q_2} = \begin{bmatrix} 3.098q_2 + 0.1962C_3 + 0.7099 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix}$$

$$\frac{\partial B}{\partial q_3} = \begin{bmatrix} -0.0948S_3C_3 - 0.05885S_3 - 0.1962q_2S_3 & 0 & 0.03923C_3 \\ * & 0 & -0.09808C_3 \\ * & * & 0 \end{bmatrix}$$

So the c_{ij} components are:

$$\begin{aligned}
c_{11} &= \frac{1}{2}\left(\frac{\partial b_{11}}{\partial q_1} + \frac{\partial b_{11}}{\partial q_1} - \frac{\partial b_{11}}{\partial q_1}\right)\dot{q}_1 + \frac{1}{2}\left(\frac{\partial b_{11}}{\partial q_2} + \frac{\partial b_{12}}{\partial q_1} - \frac{\partial b_{12}}{\partial q_1}\right)\dot{q}_2 + \frac{1}{2}\left(\frac{\partial b_{11}}{\partial q_3} + \frac{\partial b_{13}}{\partial q_1} - \frac{\partial b_{13}}{\partial q_1}\right)\dot{q}_3 \\
&= \frac{1}{2}m_{l_2}(2d_2 + 2q_2 - a_2)\dot{q}_2 + m_{l_3}S_3C_3(a_3h_3 - a_3^2 - h_3^2 - r_3^2)\dot{q}_3 \\
c_{12} &= \frac{1}{2}\left(\frac{\partial b_{12}}{\partial q_1} + \frac{\partial b_{11}}{\partial q_2} - \frac{\partial b_{21}}{\partial q_1}\right)\dot{q}_1 + \frac{1}{2}\left(\frac{\partial b_{12}}{\partial q_2} + \frac{\partial b_{12}}{\partial q_2} - \frac{\partial b_{22}}{\partial q_1}\right)\dot{q}_2 + \frac{1}{2}\left(\frac{\partial b_{12}}{\partial q_3} + \frac{\partial b_{13}}{\partial q_2} - \frac{\partial b_{23}}{\partial q_1}\right)\dot{q}_3 \\
&= 0.5\dot{q}_1\dot{q}_2(3.098q_2 + 0.1962C_3 + 0.7099) \\
c_{13} &= \frac{1}{2}\left(\frac{\partial b_{13}}{\partial q_1} + \frac{\partial b_{11}}{\partial q_3} - \frac{\partial b_{31}}{\partial q_1}\right)\dot{q}_1 + \frac{1}{2}\left(\frac{\partial b_{13}}{\partial q_2} + \frac{\partial b_{12}}{\partial q_3} - \frac{\partial b_{32}}{\partial q_1}\right)\dot{q}_2 + \frac{1}{2}\left(\frac{\partial b_{13}}{\partial q_3} + \frac{\partial b_{13}}{\partial q_3} - \frac{\partial b_{33}}{\partial q_1}\right)\dot{q}_3 \\
&= 0.03923C_3\dot{q}_3^2 - 0.5\dot{q}_1(0.0948S_3C_3 + 0.05885S_3 + 0.1962q_2S_3)\dot{q}_3 \\
c_{22} &= \frac{1}{2}\left(\frac{\partial b_{22}}{\partial q_1} + \frac{\partial b_{21}}{\partial q_2} - \frac{\partial b_{21}}{\partial q_2}\right)\dot{q}_1 + \frac{1}{2}\left(\frac{\partial b_{22}}{\partial q_2} + \frac{\partial b_{22}}{\partial q_2} - \frac{\partial b_{22}}{\partial q_2}\right)\dot{q}_2 + \frac{1}{2}\left(\frac{\partial b_{22}}{\partial q_3} + \frac{\partial b_{23}}{\partial q_2} - \frac{\partial b_{23}}{\partial q_2}\right)\dot{q}_3 \\
&= 0 \\
c_{23} &= \frac{1}{2}\left(\frac{\partial b_{23}}{\partial q_1} + \frac{\partial b_{21}}{\partial q_3} - \frac{\partial b_{31}}{\partial q_2}\right)\dot{q}_1 + \frac{1}{2}\left(\frac{\partial b_{23}}{\partial q_2} + \frac{\partial b_{22}}{\partial q_3} - \frac{\partial b_{32}}{\partial q_2}\right)\dot{q}_2 + \frac{1}{2}\left(\frac{\partial b_{23}}{\partial q_3} + \frac{\partial b_{23}}{\partial q_3} - \frac{\partial b_{33}}{\partial q_2}\right)\dot{q}_3 \\
&= 0 \\
c_{33} &= \frac{1}{2}\left(\frac{\partial b_{33}}{\partial q_1} + \frac{\partial b_{31}}{\partial q_3} - \frac{\partial b_{31}}{\partial q_3}\right)\dot{q}_1 + \frac{1}{2}\left(\frac{\partial b_{33}}{\partial q_2} + \frac{\partial b_{32}}{\partial q_3} - \frac{\partial b_{32}}{\partial q_3}\right)\dot{q}_2 + \frac{1}{2}\left(\frac{\partial b_{33}}{\partial q_3} + \frac{\partial b_{33}}{\partial q_3} - \frac{\partial b_{33}}{\partial q_3}\right)\dot{q}_3 \\
&= 0
\end{aligned}$$

So the equations of motion are:

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

$$\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\end{bmatrix} = \begin{bmatrix} 0.496\ddot{q}_1 - 0.6196\ddot{q}_2 + 0.7099\ddot{q}_1 q_2 + 0.7099\dot{q}_1\dot{q}_2^2 + 1.549\ddot{q}_1 q_2^2 + 0.05885\ddot{q}_1 C_3 + 0.03923d\dot{q}_3 S_3 + 0.0474\ddot{q}_1 C_3^2 + 0.03923\dot{q}_3^3 C_3 + 0.1962\ddot{q}_1 q_2 C_3 - 0.0474\dot{q}_1\dot{q}_3^2 sin(2.0q_3) + 0.1962\dot{q}_1\dot{q}_2^2 C_3 - 0.05885\dot{q}_1\dot{q}_3^2 S_3 + 3.098\dot{q}_1\dot{q}_2^2 q_2 - 0.1962\dot{q}_1\dot{q}_3^2 q_2 S_3 \\ 0.5\dot{q}_2(3.098q_2 + 0.1962C_3 + 0.7099)\dot{q}_1^2 - 0.6196\ddot{q}_1 + 1.549\ddot{q}_2 - 0.09808d\dot{q}_3 S_3 \\ 0.04757d\dot{q}_3 + 0.9621C_3 + \dot{q}_1(0.03923C_3\dot{q}_3^2 - 0.5\dot{q}_1(0.0948S_3C_3 + 0.05885S_3 + 0.1962q_2S_3)\dot{q}_3) + 0.03923\ddot{q}_1 S_3 - 0.09808\ddot{q}_2 S_3 \end{bmatrix}$$

where $q = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T, \dot{q} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^T, \ddot{q} = \begin{bmatrix} \ddot{q}_1 & \ddot{q}_2 & \ddot{q}_3 \end{bmatrix}^T$ and:

$$\begin{aligned} B(q) &= B_1(q) + B_2(q) + B_3(q) \\ &= \begin{bmatrix} K & -a_1(m_{l_2} + m_{l_3}) & \frac{1}{2}m_{l_3}a_1(2a_3 - h_3)S_3 \\ * & m_{l_2} + m_{l_3} & -\frac{1}{2}m_{l_3}(2a_3 - h_3)S_3 \\ * & * & a_3^2 - a_3h_3 + h_3^2 + \frac{1}{2}r_3^2 \end{bmatrix} \\ K &= \frac{1}{2}m_{l_1}\left((a_1 - h_1)^2 + r_1^2\right) + m_{l_2}\left(a_1^2 + \left(d_2 + q_2 - \frac{1}{2}a_2\right)^2 + \frac{1}{12}\left(a_2^2 + c_2^2\right)\right) \\ &\quad + m_{l_3}\left(a_1^2 + \left(d_2 + q_2 - \left(\frac{1}{2}h_3 - a_3\right)C_3\right)^2 + \frac{1}{12}(h_3^2 + r_3^2)\right) \end{aligned}$$

$$C(q,\dot{q})\dot{q} = \begin{bmatrix} 0.5\dot{q}_2^2(3.098q_2 + 0.1962C_3 + 0.7099) - 0.5\dot{q}_3^2(0.0958S_3C_3 + 0.05885S_3 + 0.1962q_2S_3) & 0.5\dot{q}_1\dot{q}_2(3.098q_2 + 0.1962C_3 + 0.7099) & 0.03923C_3\dot{q}_3^2 - 0.5\dot{q}_1(0.0948S_3C_3 + 0.05885S_3 + 0.1962q_2S_3)\dot{q}_3 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} 0 \\ 0 \\ -m_{l_3}g\left(a_3 - \frac{1}{2}h_3\right)C_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -0.9621C_3 \end{bmatrix}$$

4 Assignment 4

4.1 Compute the dynamic model using the recursive Newton-Euler formulation

The Newton-Euler formulation is a recursive algorithm used to compute the dynamic model.

/* Initial Conditions */
$\omega_0,\ \ddot{p}_0 - g_0,\ \dot{\omega}_0$
for $i = 1$ to n **do**
  Given current $q_i, \dot{q}_i, \ddot{q}_i$ (i.e. $\vartheta_i, \dot{\vartheta}_i, \ddot{\vartheta}_i$ or $d_i, \dot{d}_i, \ddot{d}_i$)
  /* if revolute joint add, if prismatic joint add */
  $R_i^{i-1} = R_i^{i-1}(\vartheta_i)$ or $R_i^{i-1} = R_i^{i-1}(d_i)$
  $\omega_i^i = (R_i^{i-1})^T \omega_{i-1}^{i-1} + (R_i^{i-1})^T \dot{\vartheta}_i z_0$
  $\dot{\omega}_i^i = (R_i^{i-1})^T \dot{\omega}_{i-1}^{i-1} + (R_i^{i-1})^T (\ddot{\vartheta}_i z_0 + \dot{\vartheta}_i \omega_{i-1}^{i-1} \times z_0)$
  $\ddot{p}_i = (R_i^{i-1})^T \ddot{p}_{i-1}^{i-1} + \dot{\omega}_i^i \times r_{i-1,i}^i + \omega_i^i \times (\omega_i^i \times r_{i-1,i}^i) + (R_i^{i-1})^T \ddot{d}_i z_0 + 2\dot{d}_i \omega_i^i \times ((R_i^{i-1})^T z_0)$
  $\ddot{p}_{C_i} = \ddot{p}_i^i + \dot{\omega}_i^i \times r_{i,C_i}^i + \omega_i^i \times (\omega_i^i \times r_{i,C_i}^i)$
  $\dot{\omega}_{m_i}^{i-1} = \dot{\omega}_{i-1}^{i-1} + k_{ri} \ddot{q}_i z_{m_i}^{i-1} + k_{ri} \dot{q}_i \omega_{i-1}^{i-1} \times z_{m_i}^{i-1}$
end for

Algorithm 1: Forward equations

/* Initial Conditions */
$h_e = \begin{bmatrix} f_{n+1} \\ \mu_{n+1} \end{bmatrix}$
$f_{n+1}^{n+1} = f_{n+1},\ \mu_{n+1}^{n+1} = \mu_{n+1}$
for $i = n$ to 1 **do**
  Given current $\omega_i^i, \dot{\omega}_i^i, \ddot{p}_i^i, \ddot{p}_{C_i}^i, \dot{\omega}_{m_i}^i$
  $R_{i+1}^i = R_{i+1}^i(\vartheta_{i+1})$ or $R_{i+1}^i = R_{i+1}^i(d_{i+1})$
  $f_i^i = R_{i+1}^i f_{i+1}^{i+1} + m_i \ddot{p}_{C_i}^i$
  $\mu_i^i = -f_i^i \times (r_{i-1,i}^i + r_{i,C_i}^i) + R_{i+1}^i \mu_{i+1}^{i+1} + R_{i+1}^i f_{i+1}^{i+1} \times r_{i,C_i}^i + \bar{I}_i^i \dot{\omega}_i^i + \omega_i^i \times (\bar{I}_i^i \omega_i^i) + k_{r,i+1} \ddot{q}_{i+1} I_{m_{i+1}} z_{m_{i+1}}^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \omega_i^i \times z_{m_{i+1}}^i$
  $\tau_i = \begin{cases} (f_i^i)^T (R_i^{i-1})^T z_0 + k_{ri} I_{m_i} (\dot{\omega}_{m_i}^{i-1})^T z_{m_i}^{i-1} + F_{vi} \dot{d}_i + F_{si} \mathrm{sign}(\dot{d}_i) \\ (\mu_i^i)^T (R_i^{i-1})^T z_0 + k_{ri} I_{m_i} (\dot{\omega}_{m_i}^{i-1})^T z_{m_i}^{i-1} + F_{vi} \dot{\vartheta}_i + F_{si} \mathrm{sign}(\dot{\vartheta}_i) \end{cases}$
end for

Algorithm 2: Backward equations

To check the results, the resulting B_{RNE}, C_{RNE} e g_{RNE} were compared to the corresponding quantities computed with the Lagrangian method, in a random robot configuration and joint velocities:

$$B_{RNE} = \begin{bmatrix} 0.4372 & -0.61966 & 0.03208 \\ -0.6196 & 1.549 & -0.06925 \\ 0.03208 & -0.069256 & 0.04708 \end{bmatrix} \quad B_L = \begin{bmatrix} 0.4372 & -0.61966 & 0.03208 \\ -0.6196 & 1.549 & -0.06925 \\ 0.03208 & -0.069256 & 0.04708 \end{bmatrix}$$

$$C_{RNE} = \begin{bmatrix} -0.001341 \\ 0.0005088 \\ -4.893e-5 \end{bmatrix} \quad C_L = \begin{bmatrix} -0.001341 \\ -0.001073 \\ 0.0002748 \end{bmatrix}$$

$$g_{RNE} = \begin{bmatrix} 0 \\ 0 \\ -0.5539 \end{bmatrix} \quad g_L = \begin{bmatrix} 0 \\ 0 \\ -0.5539 \end{bmatrix}$$

5 Assignment 5

5.1 Compute the dynamic model in the operational space

The dynamic model in the operational space is described as follows:

$$B_A(x)\ddot{x} + C_A(x,\dot{x})\dot{x} + g_A(x) = u - u_e$$

where:

$$\begin{aligned} B_A(x) &= J_A^{-T} B J_A^{-1} \\ C_A\dot{x} &= (J_A^{-T} C - B_A \dot{J}_A)\dot{q} \\ g_A(x) &= J_A^{-T} g \\ u &= T_A^T h \\ u_e &= T_A^T h_e \end{aligned}$$

with:

$$T_A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_1 & 0 \\ 0 & 0 & 0 & 0 & S_1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\dot{J}_A = \begin{bmatrix} C_1\ddot{q}_2 - S_1\dot{q}_1(d_2 + q2) - a_1C_1\dot{q}_1 - a_3C_3S_1\dot{q}_1 - a_3C_1S_3\ddot{q}_3 & C_1\dot{q}_1 & -a_3C_1S_3\dot{q}_1 - a_3C_3S_1\ddot{q}_3 \\ S_1\ddot{q}_2 + C_1\dot{q}_1(d_2 + q2) - a_1S_1\dot{q}_1 + a_3C_1C_3\dot{q}_1 - a_3S_1S_3\ddot{q}_3 & S_1\dot{q}_1 & a_3C_1C_3\ddot{q}_3 - a_3S_1S_3\dot{q}_1 \\ 0 & 0 & -a_3S_3\ddot{q}_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The resulting matrices are not reported for space reasons.

6 Assignment 6

6.1 Design the joint space PD control law with gravity compensation

The goal is to implement the following architecture:

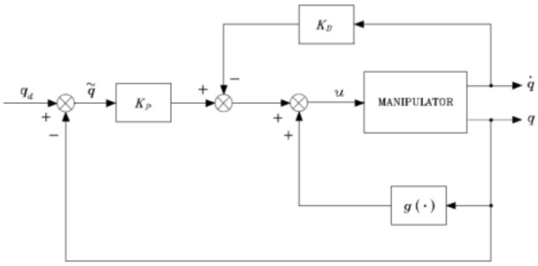


Figure 3: Joint space PD control law with gravity compensation architecture

The architecture follows the control law:

$$\tau = g(q) + K_P(q_d - q) + K_D(\dot{q}_d - q_d) \qquad \text{with } K_P = 50, K_D = 10$$

The architecture was implemented in SIMULINK as follows:

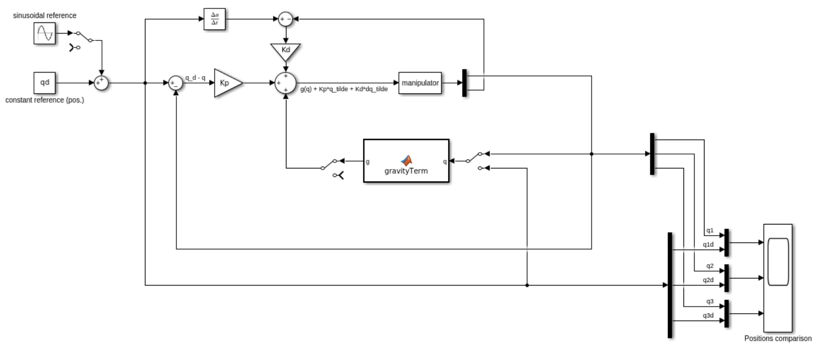


Figure 4: Joint space PD control law with gravity compensation SIMULINK model

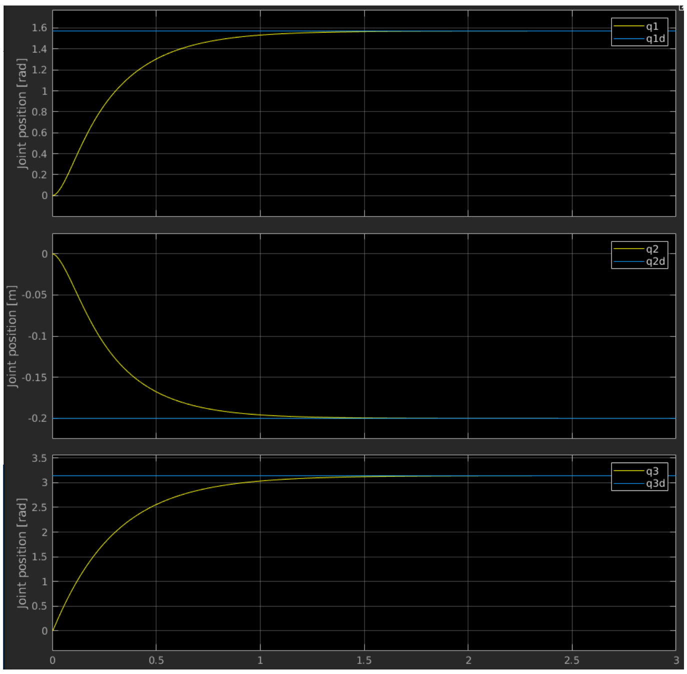


Figure 5: Joint positions - constant reference $q_d = \begin{bmatrix} \pi/2 & -0.2 & \pi \end{bmatrix}^T$

6.2 What happens if g(q) is not taken into account?

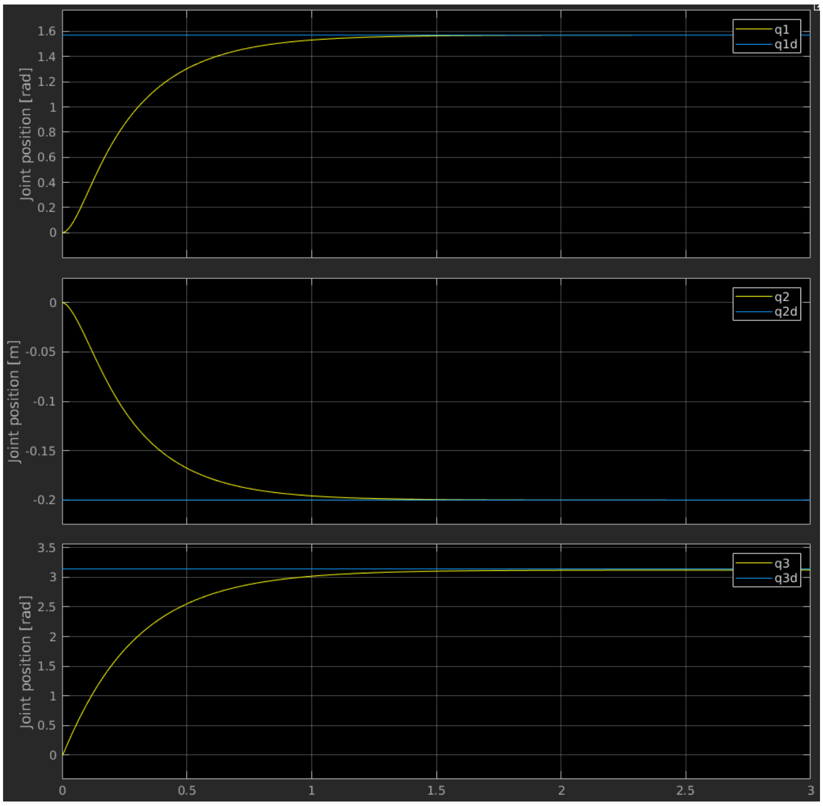


Figure 6: Joint positions - $g(q) = 0$

6.3 What happens if the gravity term is equal to g(q_d)?

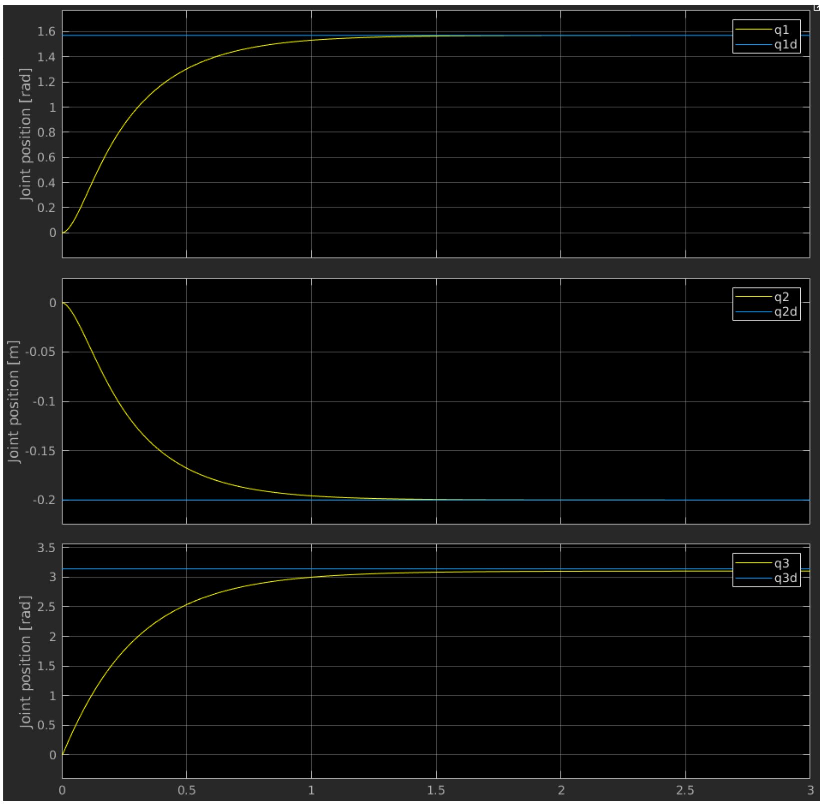


Figure 7: Joint positions - $g(q) = g(q_d)$

6.4 What happens if q_d is not constant?

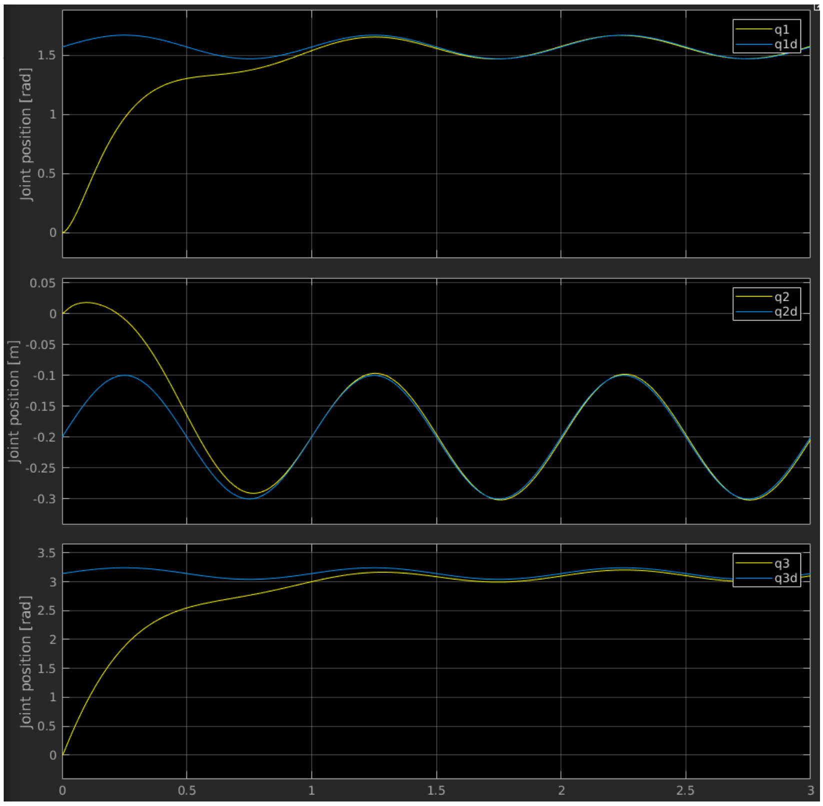


Figure 8: Joint positions - sinusoidal reference $q_d = \begin{bmatrix}\pi/2 & -0.2 & \pi\end{bmatrix}^T + 0.1sin(2\pi)$

7 Assignment 7

7.1 Design the joint space inverse dynamics control law

The goal is to implement the following architecture:

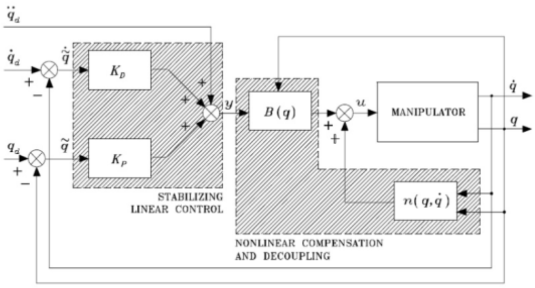


Figure 9: Joint space PD control law with gravity compensation architecture

The architecture works by linearizing non-linear dynamics and by decoupling each joint variable. It does so by implementing an inner feedback loop:

$$\tau = B(q)y + n(q, \dot{q})$$

where $n(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q)$ and y is controlled by the outer feedback loop:

$$y = \ddot{q}_d + K_D(\dot{q}_d - q) + K_P(q_d - q) \quad \text{with } K_P = 50, K_D = 10$$

The architecture was implemented in SIMULINK as follows:

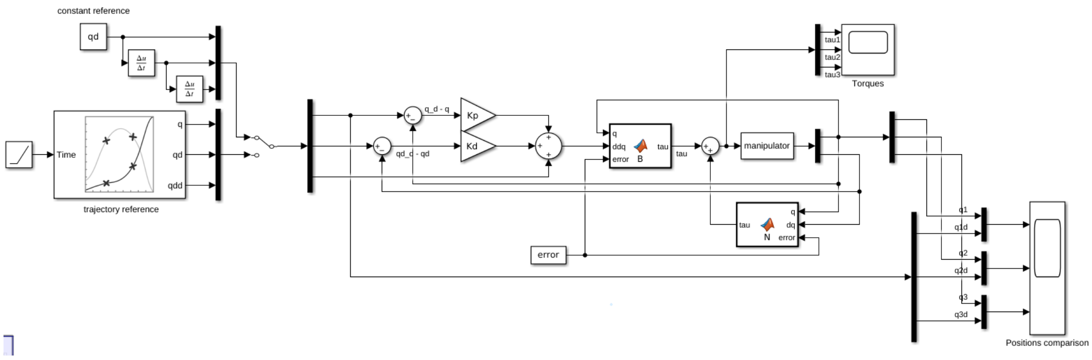


Figure 10: Joint space inverse control law SIMULINK model

The architecture is tested with a quintic polynomial trajectory that passes through the waypoints:

$$\begin{aligned}
q_d \mid_{t=0.5} &= \begin{bmatrix} \pi/3 & -0.2 & \pi/3 \end{bmatrix} \\
q_d \mid_{t=1} &= \begin{bmatrix} -\pi/3 & -0.1 & \pi/4 \end{bmatrix} \\
q_d \mid_{t=1.5} &= \begin{bmatrix} 0 & -0.2 & 0 \end{bmatrix} \\
q_d \mid_{t=2} &= \begin{bmatrix} \pi/2 & 0 & -\pi/4 \end{bmatrix} \\
q_d \mid_{t=2.5} &= \begin{bmatrix} \pi/3 & -0.1 & \pi/4 \end{bmatrix} \\
q_d \mid_{t=3} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}
\end{aligned}$$

The boundary conditions for the generation of the quintic trajectory are that the velocity and acceleration are null in each of the waypoints.

To show the effect of decoupling, the architecture was also tested against a constant reference $q_d = \begin{bmatrix} \pi/1 & -0.2 & \pi \end{bmatrix}$.

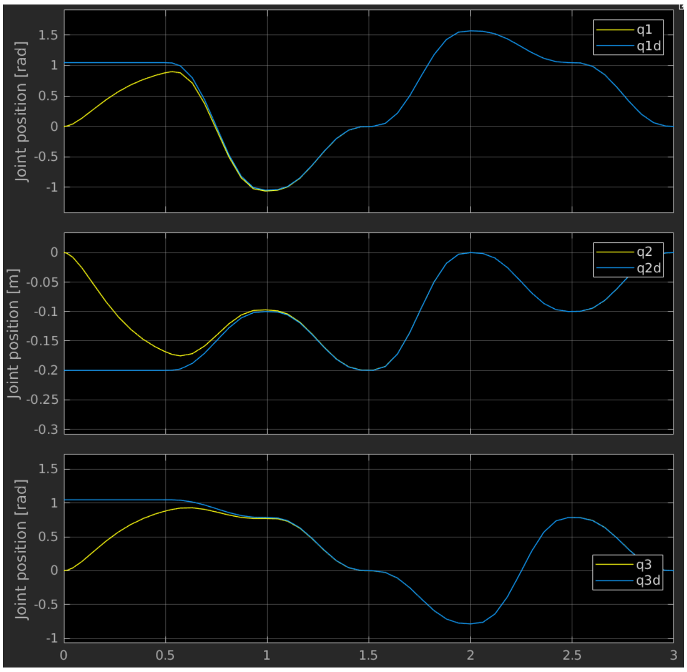


Figure 11: Joint positions

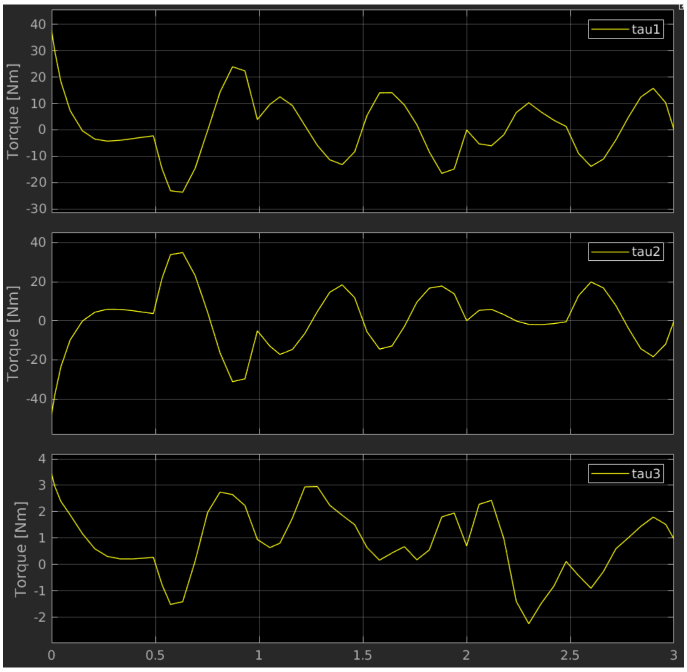


Figure 12: Joint torques

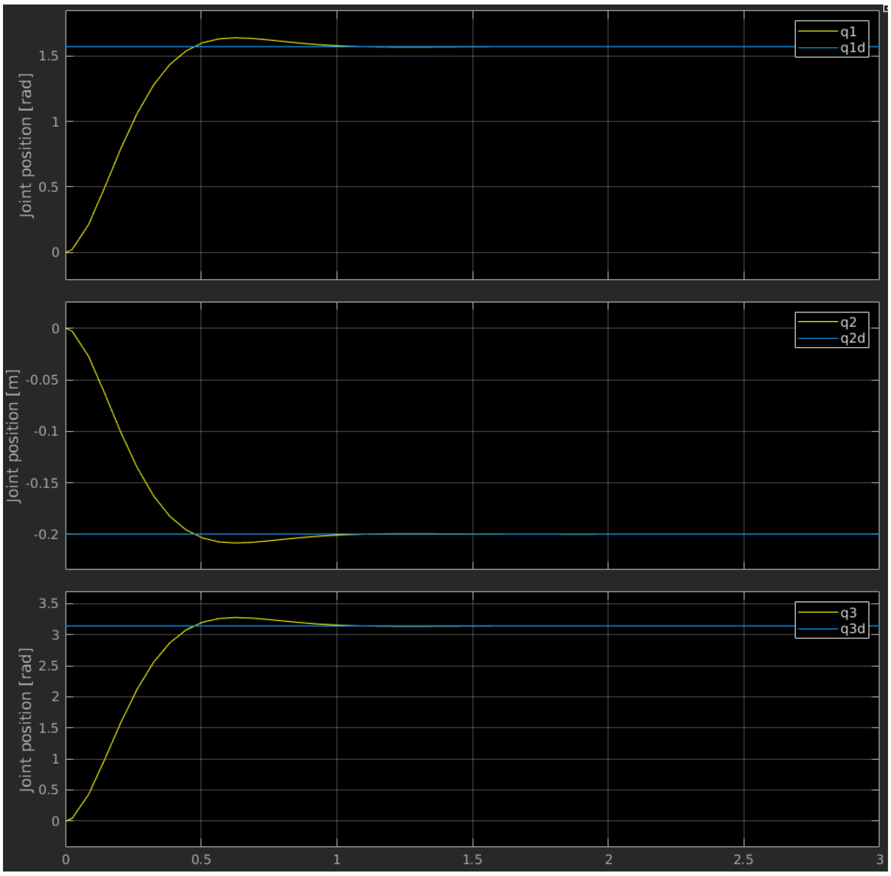


Figure 13: Joint positions - constant reference, decoupling

7.2 Check the behaviour of the control law when the B, C and g values are different than the true ones

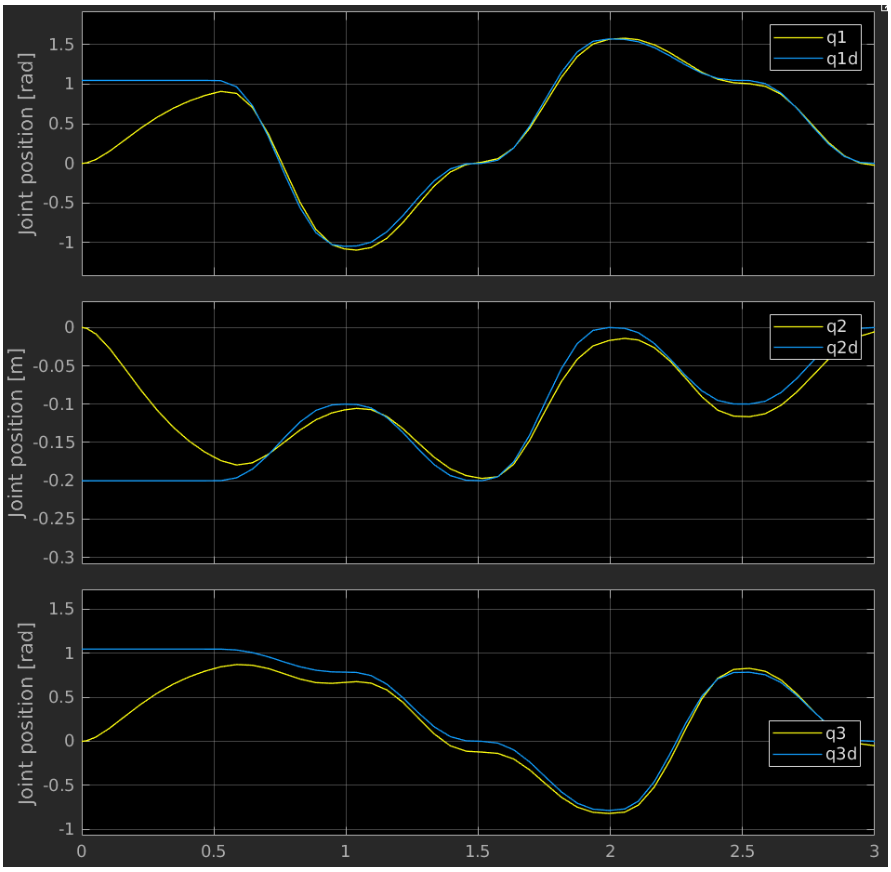


Figure 14: Joint positions - changed values of the link masses

7.3 What happens to the torque values when the settling time of the equivalent second order systems is chosen very small?

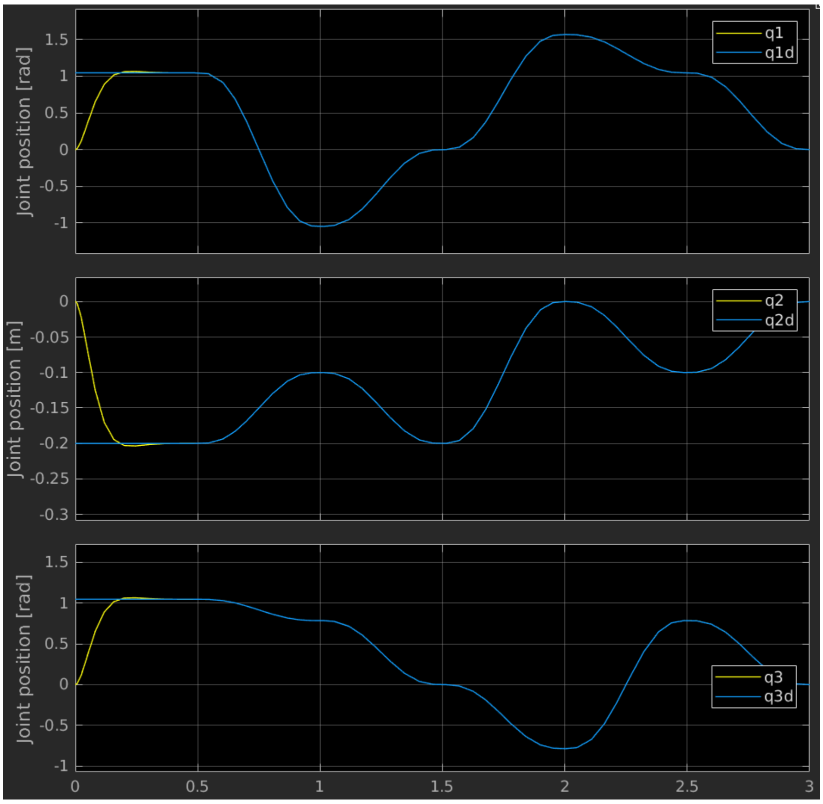


Figure 15: Joint positions - $K_P = 500, K_D = 35$

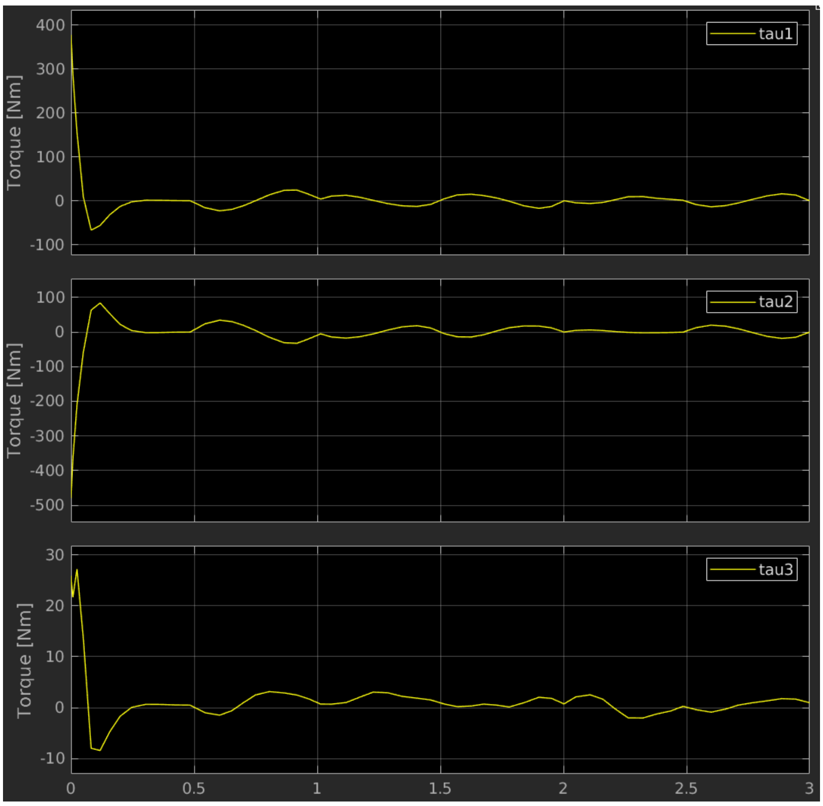


Figure 16: Joint torques - $K_P = 500, K_D = 35$

8 Assignment 8

8.1 Implement in Simulink the Adaptive Control law for the a 1-DoF link under gravity.

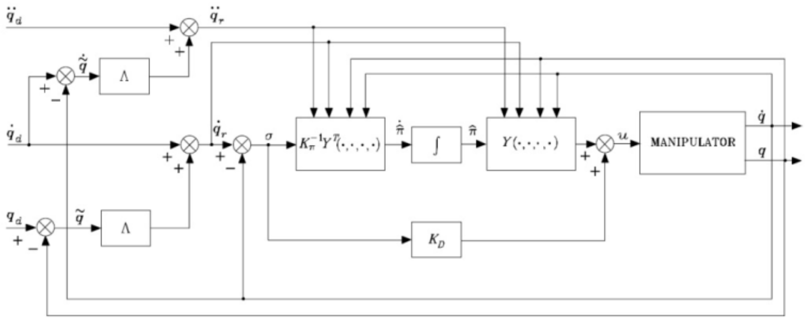


Figure 17: Adaptive control architecture

The adaptive control law tackles the problem of model uncertainty by implementing an on-line estimator of the robot's dynamic parameters.

The plant is modelled with three dynamic parameters (inertia, friction and gravity term):

$$I\ddot{q} + F\dot{q} + G\sin q = \tau \implies \tau = \begin{bmatrix}\ddot{q} & \dot{q} & \sin q\end{bmatrix}\begin{bmatrix}I\\F\\G\end{bmatrix} = Y(q,\dot{q},\ddot{q})\Theta$$

The estimation of the dynamic parameters is as follows:

$$\hat{\Theta} = \begin{bmatrix}\hat{I}\\\hat{F}\\\hat{G}\end{bmatrix} = \begin{bmatrix}\gamma_1 & 0 & 0\\0 & \gamma_2 & 0\\0 & 0 & \gamma_3\end{bmatrix}\begin{bmatrix}\ddot{q}\\\dot{q}\\\sin q\end{bmatrix}(\dot{q}_r - \dot{q})$$

where the reference velocity is computed as:

$$\dot{q}_r = \dot{q}_d + \lambda(q_d - q) = \dot{q}_d + \frac{K_P}{K_D}(q_d - q)$$

The architecture is implemented in SIMULINK as follows:

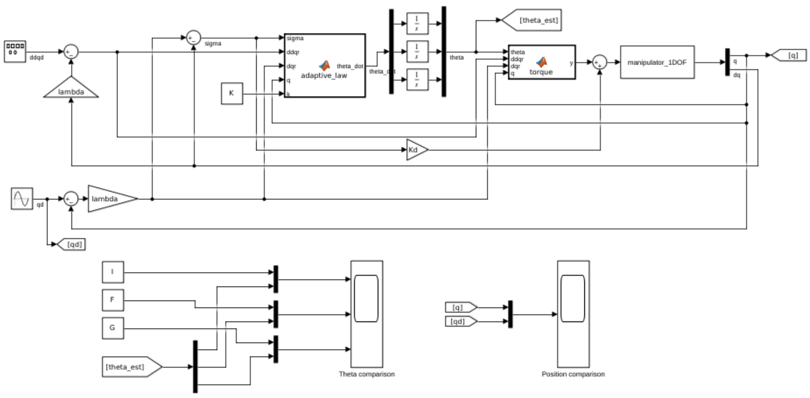


Figure 18: Adaptive control SIMULINK model

The model is tested with a sinusoidal reference trajectory $q_d = A\sin(\omega t)$ and a periodic square wave acceleration reference $\ddot{q}_d = square(\pm A)$, with $A = 1$ and $\omega = 2\pi$.

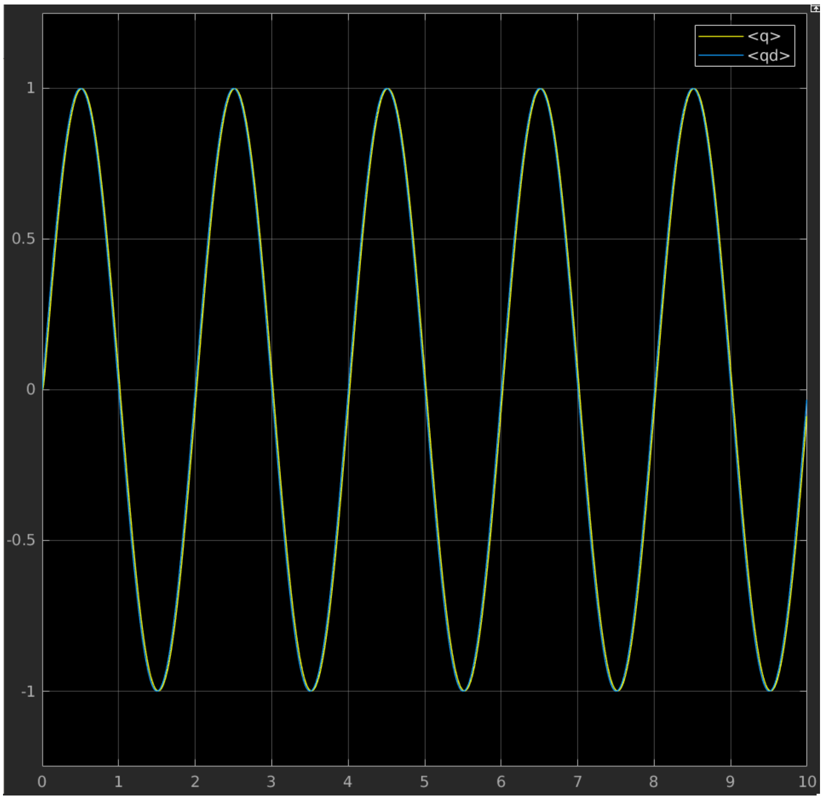


Figure 19: Adaptive control - position reference tracking

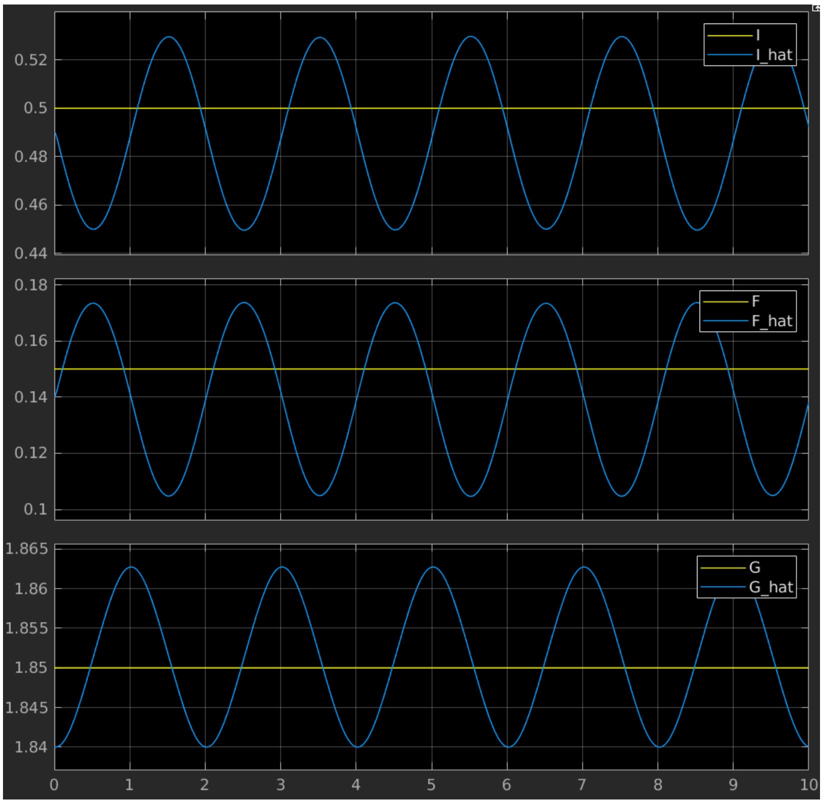


Figure 20: Adaptive control - parameter estimation

9 Assignment 9

9.1 Design the operational space PD control law with gravity compensation

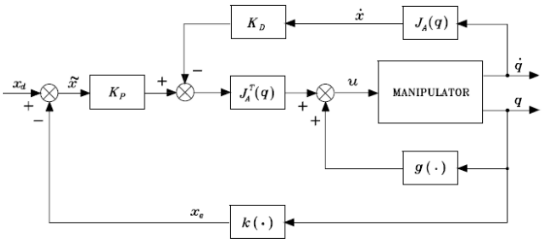


Figure 21: Operational space PD control law with gravity compensation architecture

The operational space PD control law with gravity compensation works exactly the same as the joint space one, except for the transformation of the joint space quantities into the corresponding operational space ones with direct kinematics (i.e. with the use of the transformation matrix T and of the analytical Jacobian JA). The control law parameters are $K_P = 50, K_D = 10$. The architecture was tested with a constant pose reference obtained by applying direct kinematics to the joint configuration $q = \begin{bmatrix} -\pi/3 & -0.1 & \pi/2 \end{bmatrix}$.

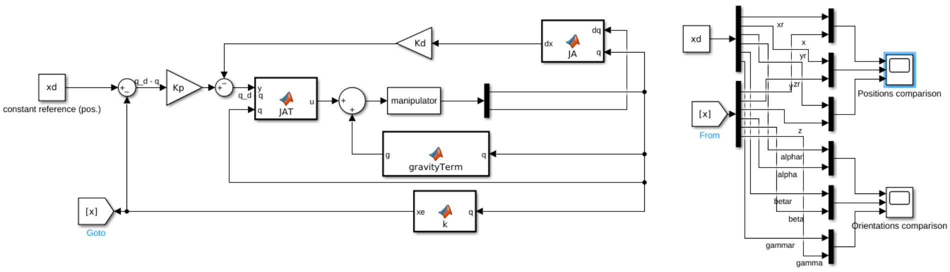


Figure 22: Operational space PD control law with gravity compensation SIMULINK model

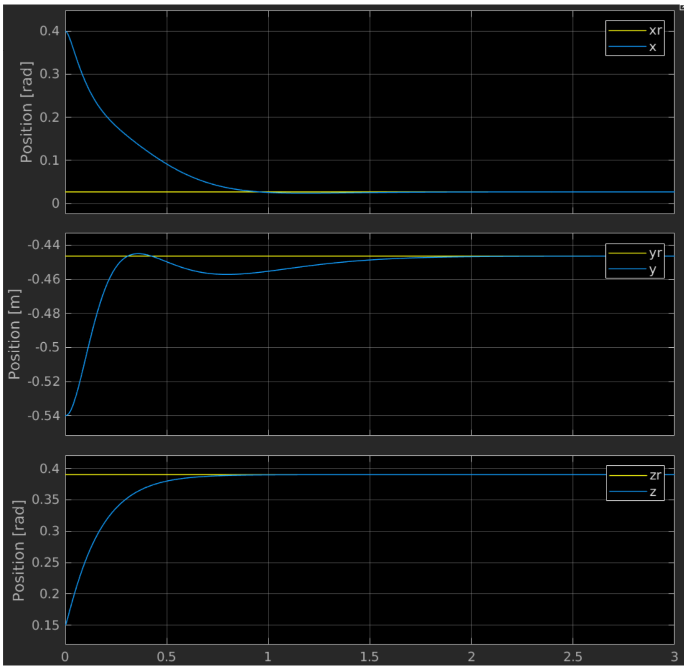


Figure 23: Pose tracking - end-effector xyz coordinates

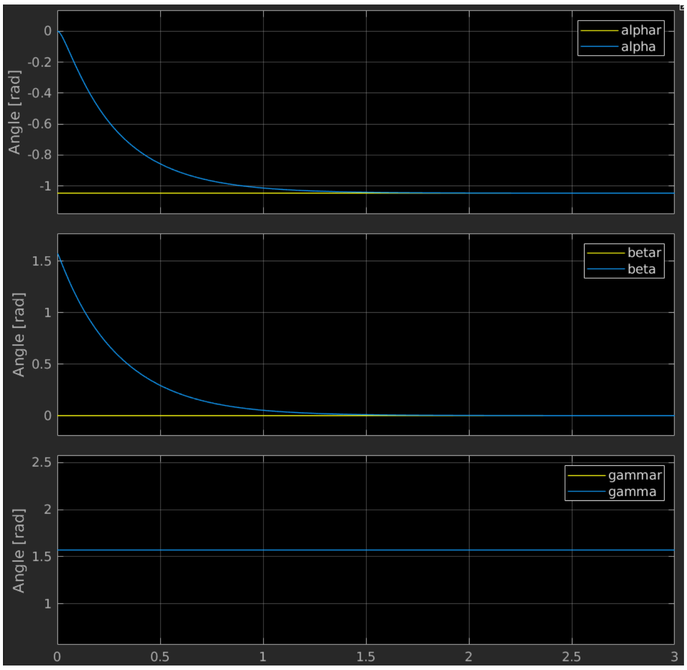


Figure 24: Pose tracking - end-effector $\alpha\beta\gamma$ coordinates

10 Assignment 10

10.1 Design the operational space inverse dynamics control law

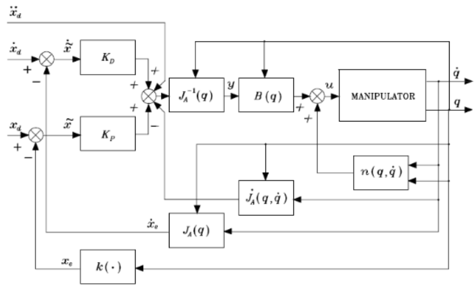


Figure 25: Operational space inverse dynamics control law architecture

The operational space inverse dynamics control law works exactly the same as the joint space one, except for the transformation of the joint space quantities into the corresponding operational space ones with direct kinematics (i.e. with the use of the transformation matrix T, the analytical Jacobian JA and its inverse JA^{-1} and derivative $\dot{J}A$). The control law parameters are $K_P = 100, K_D = 15$. The architecture was tested with a constant pose reference obtained by applying direct kinematics to the same waypoints seen in the joint space case.

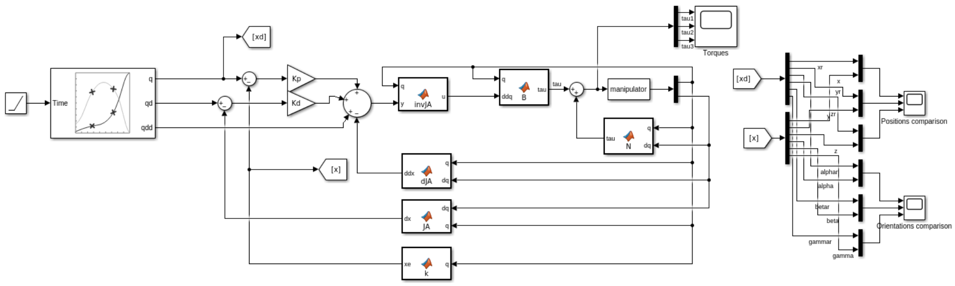
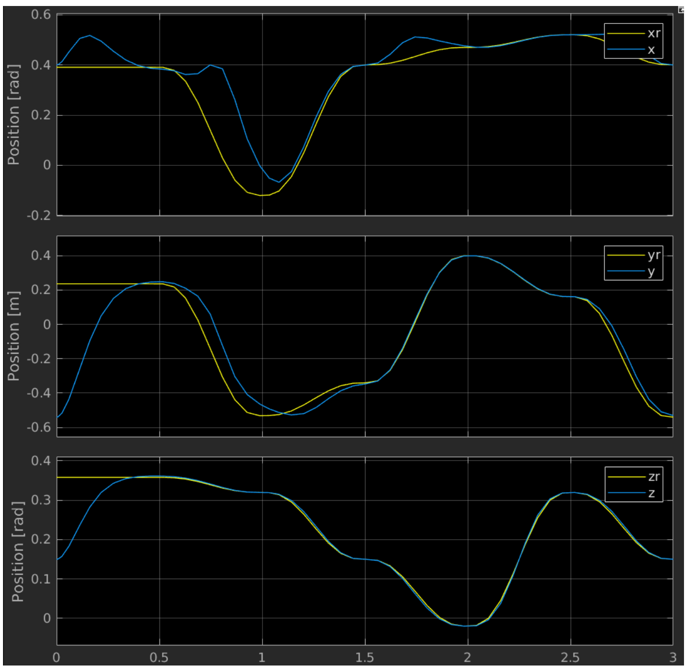


Figure 26: Operational space inverse dynamics control lawSIMULINK model



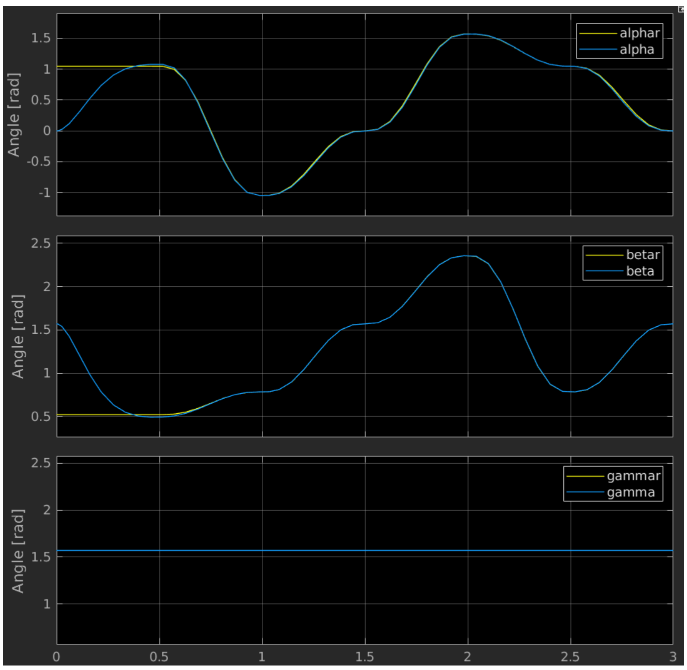


Figure 27: Pose tracking - end-effector xyz coordinates

Figure 28: Pose tracking - end-effector $\alpha\beta\gamma$ coordinates

11 Assignment 11

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

Compliance control is a form of indirect force control, i.e. a motion control without the explicit closure of a force feedback loop. Such an architecture allows the handling of non-null external wrenches h_e:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau - J^T(q)h_e$$

By looking at the above expression at the equilibrium we can relate the external wrench to the position error $\tilde{x}$ in operational space, through the use of the vector of external forces in operational space h_A and a compliance matrix K^{-1}:

$$\tilde{x} = K^{-1}h_A$$

The new PD control law becomes:

$$\tau = g(q) + J_{Ad}^T(q, \tilde{x})(K_P\tilde{x} - K_D J_{Ad}(q, \tilde{x})\dot{q}) \quad \text{with } K_P = 50, K_D = 10$$

The architecture is modelled in SIMULINK as follows:

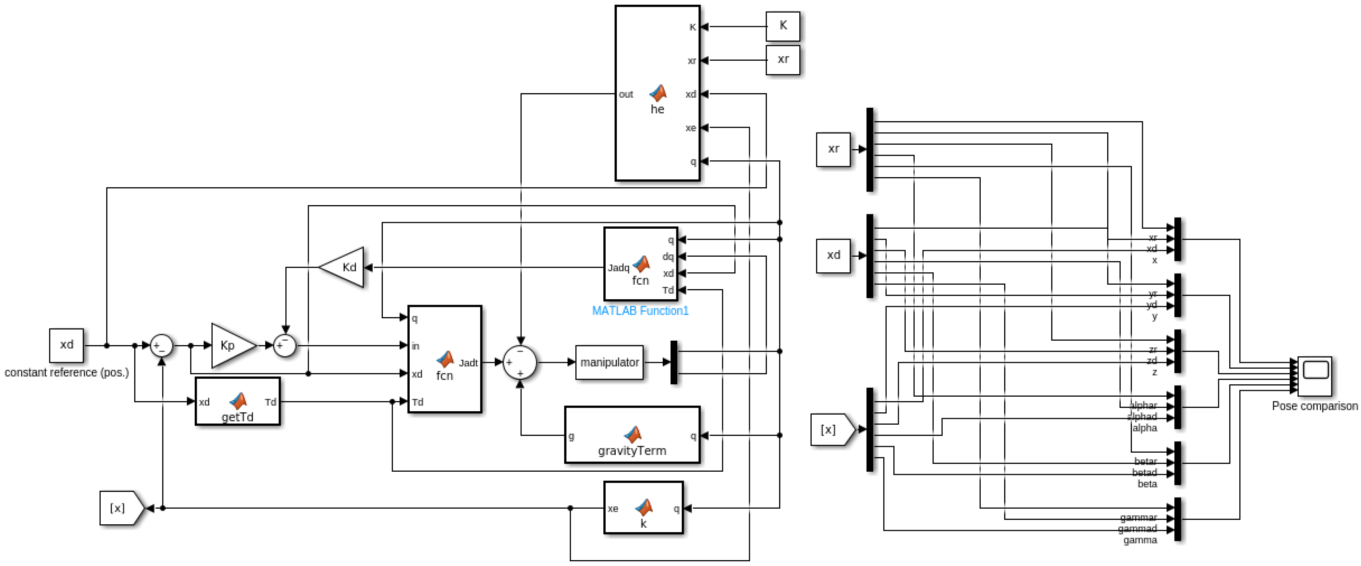


Figure 29: Compliance control in operational space SIMULINK model

To test the architecture, the manipulator was moved to $x_0 = k(\begin{bmatrix} 0 & -0.3 & 0 \end{bmatrix})$, with a desired position of $x_d = k(\begin{bmatrix} 0 & -0.2 & 0 \end{bmatrix})$ and the environment placed at $x_e = k(\begin{bmatrix} 0 & -0.1 & 0 \end{bmatrix})$. This was done to simplify the architecture, allowing for contact only in the y direction.

11.2 $K_{env} \ll K_P$

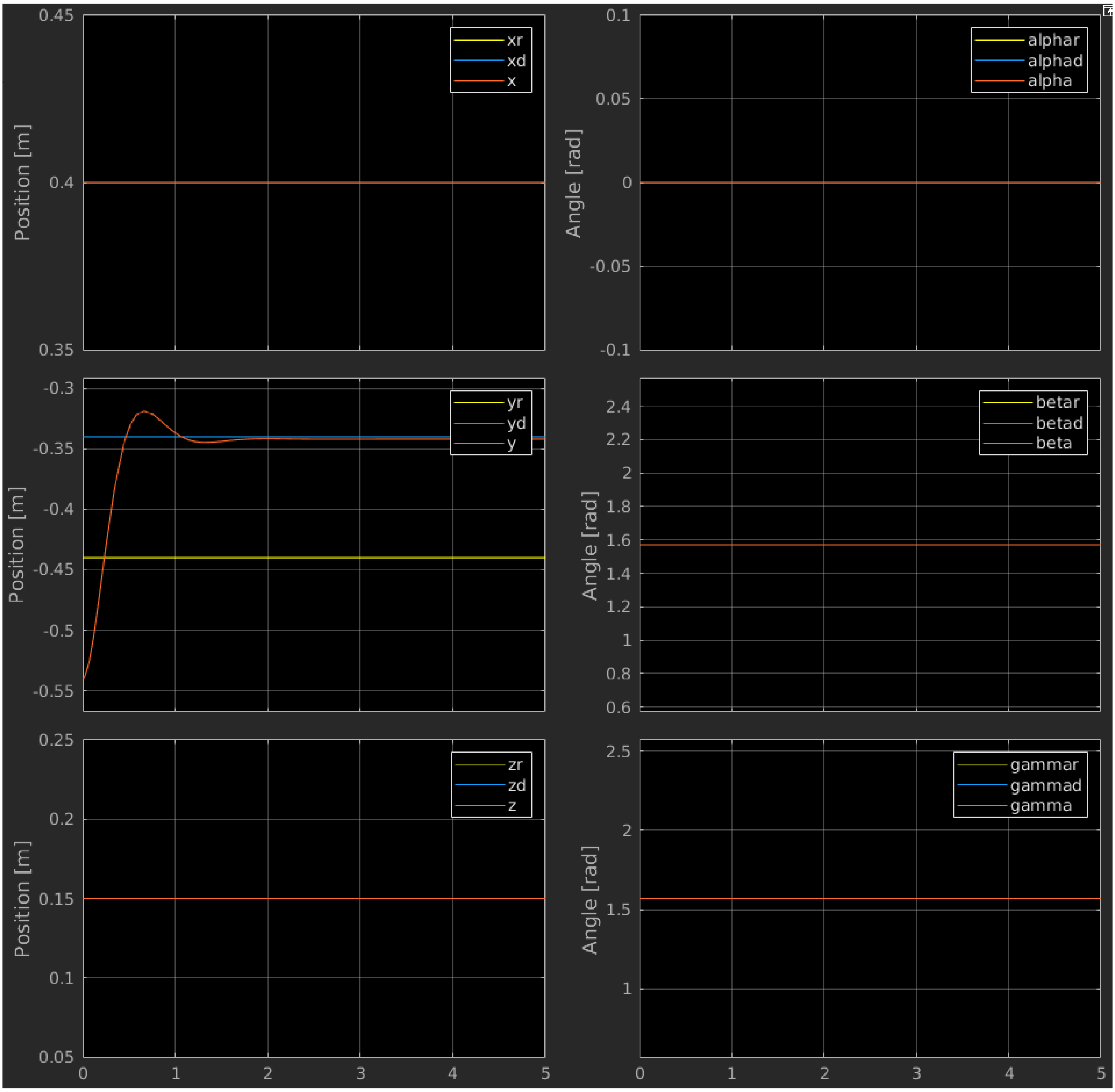


Figure 30: Compliance control - $K = 1$

11.3 $K_{env} = K_P$

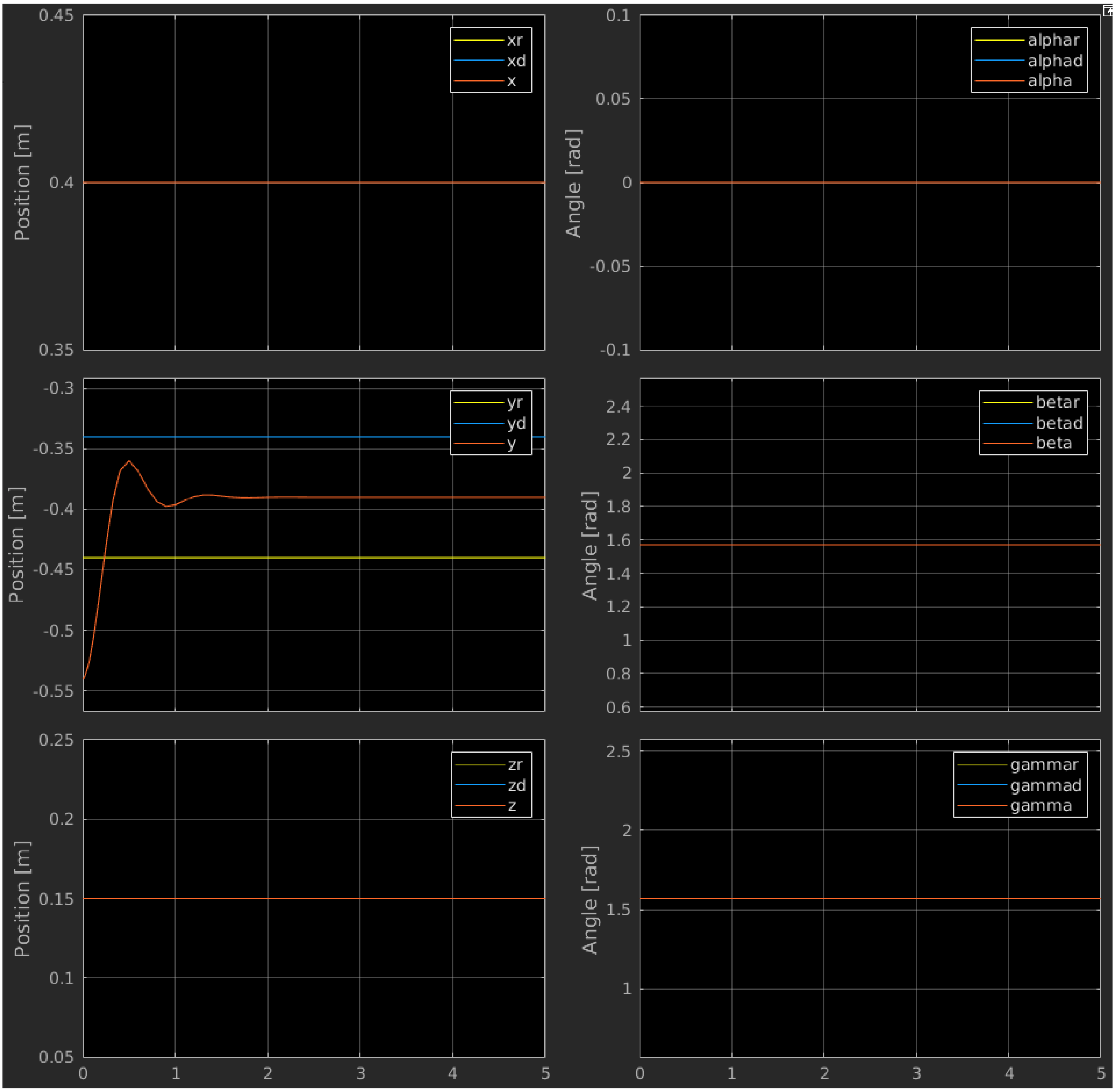


Figure 31: Compliance control - $K = 50$

11.4 $K_{env} \gg K_P$

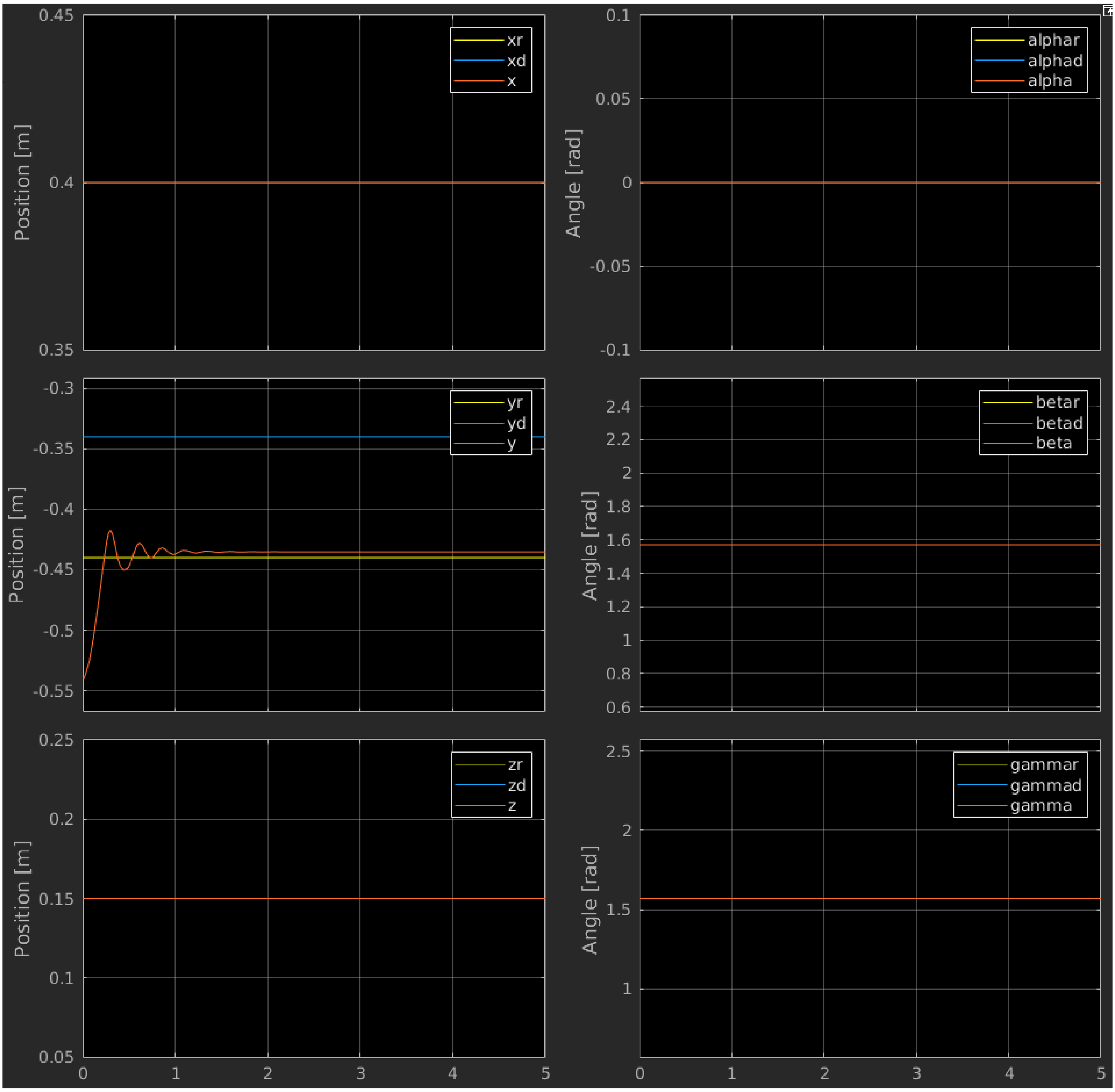


Figure 32: Compliance control - $K = 1000$

12 Assignment 12

12.1 Implement the impedance control in the operational space

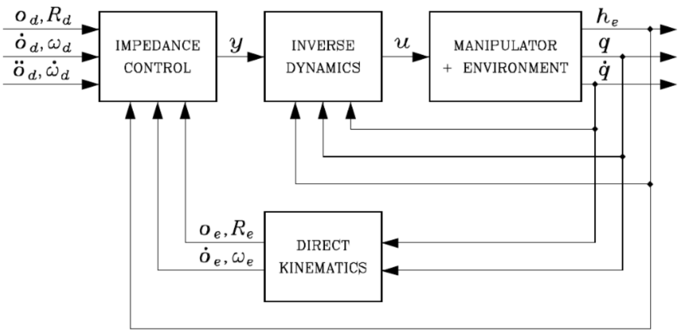


Figure 33: Impedance control in operational space architecture

The architecture implements the following stabilizing linear control law:

$$y = J_A^{-1}(q)M_d^{-1}(K_D\dot{\tilde{x}} + K_P\tilde{x} - M_d\dot{J}_A(q,\dot{q})\dot{q} - M_d b(\tilde{x}, R_d, \dot{o}_d, \omega_d) - h_e^d)$$

The architecture is based on the definition of a mechanical impedance defined by an equivalent mass matrix M_d, an equivalent damping matrix K_D and an equivalent stiffness matrix K_P. The mechanical impedance is independent from the configuration of the robot.

The architecture is modelled in SIMULINK as follows:

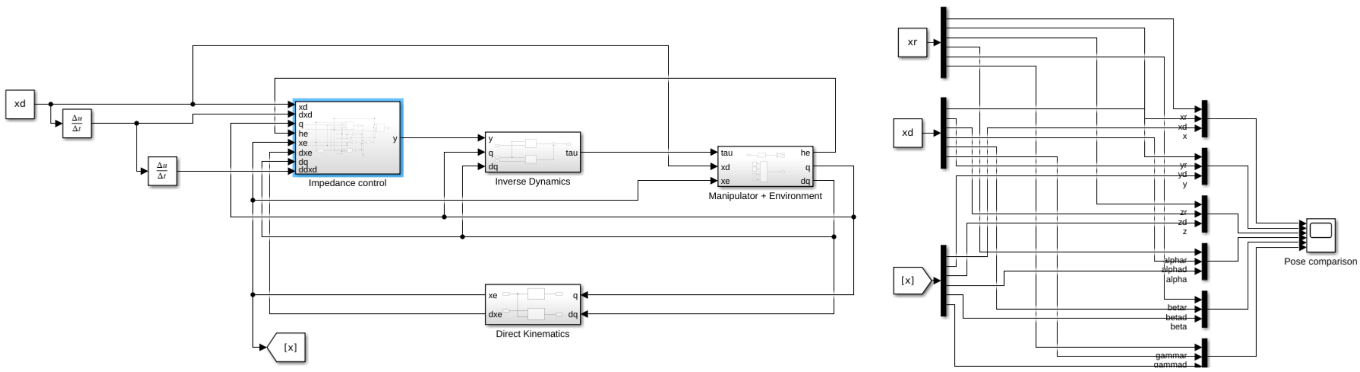


Figure 34: Impedance control in operational space SIMULINK model

To test the architecture, the manipulator was moved to $x_0 = k(\begin{bmatrix}0 & -0.3 & 0\end{bmatrix})$, with a desired position of $x_d = k(\begin{bmatrix}0 & -0.2 & 0\end{bmatrix})$ and the environment placed at $x_e = k(\begin{bmatrix}0 & -0.1 & 0\end{bmatrix})$. This was done to simplify the architecture, allowing for contact only in the y direction. The mechanical impedance was defined with $M_d = \begin{bmatrix}0.3 & 1 & 0.3 & 0.3 & 0.3 & 0.3\end{bmatrix}$, $K_D = \begin{bmatrix}30 & 25 & 30 & 30 & 30 & 30\end{bmatrix}$ and $K_P = \begin{bmatrix}50 & 75 & 50 & 50 & 50 & 50\end{bmatrix}$.

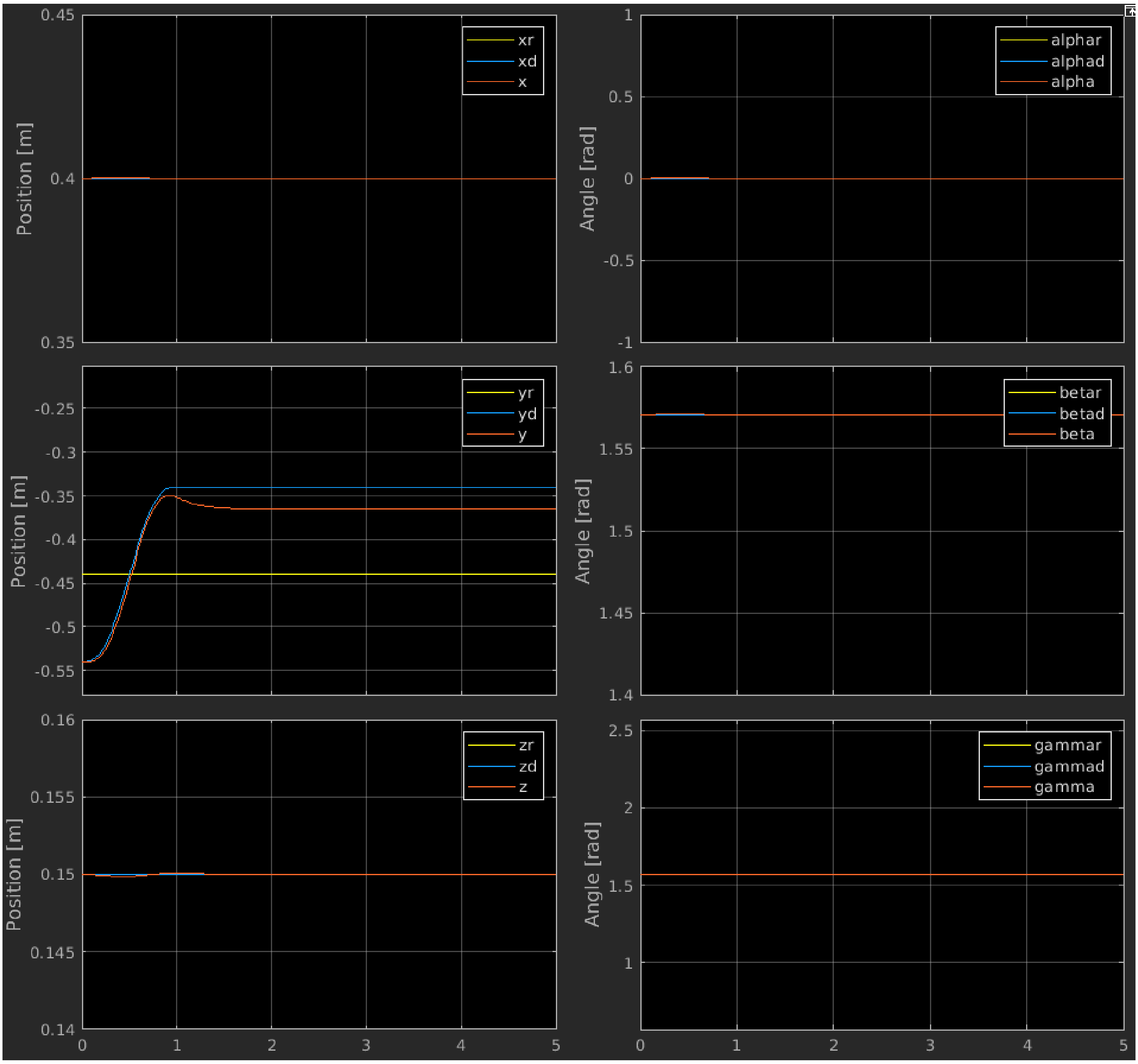


Figure 35: Impedance control

13 Assignment 13

13.1 Implement the admittance control in the operational space

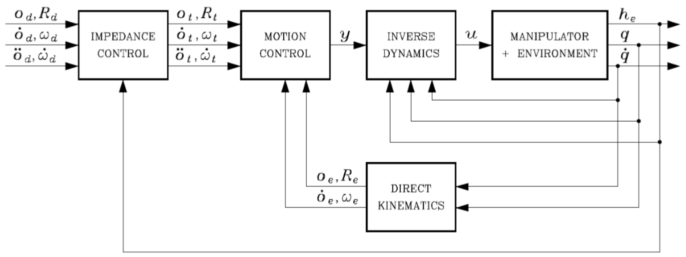


Figure 36: Admittance control in operational space architecture

The relationship between the external interaction force h_e^d and the operational space error $\tilde{z}$ is defined through an admittance:

$$M_t \ddot{\tilde{z}} + K_D \dot{\tilde{z}} + K_P \tilde{z} = h_e^d$$

The operational space error is between the desired frame and a compliant frame, introduce to force a compliant behaviour of the end-effector. The manipulator then always follows the compliant trajectory, which is equal to the desired one only in absence of external disturbances.

The architecture is modelled in SIMULINK as follows:

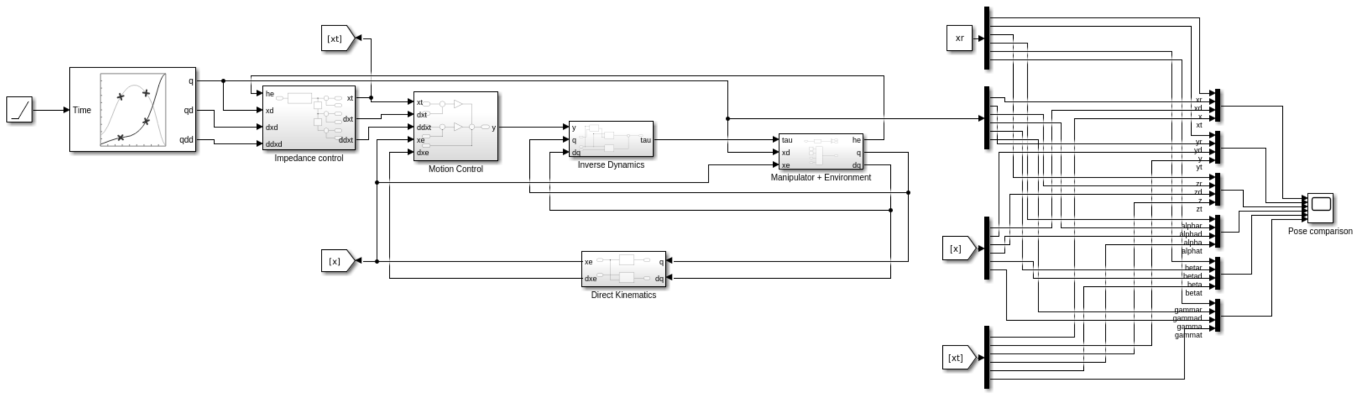


Figure 37: Admittance control in operational space SIMULINK model

To test the architecture, the manipulator was moved to $x_0 = k(\begin{bmatrix} 0 & -0.3 & 0 \end{bmatrix})$, with a desired position of $x_d = k(\begin{bmatrix} 0 & -0.25 & 0 \end{bmatrix})$ and the environment placed at $x_e = k(\begin{bmatrix} 0 & -0.2 & 0 \end{bmatrix})$. This was done to simplify the architecture, allowing for contact only in the y direction. The mechanical impedance was defined with $M_t = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$, $K_{Dt} = \begin{bmatrix} 1 & 10 & 1 & 1 & 1 & 1 \end{bmatrix}$ and $K_{Pt} = \begin{bmatrix} 1 & 20 & 1 & 1 & 1 & 1 \end{bmatrix}$, while the control law is defined by $K_D = \begin{bmatrix} 10 & 10 & 10 & 10 & 10$ and $K_P = \begin{bmatrix} 50 & 50 & 50 & 50 & 50 & 50 \end{bmatrix}$.

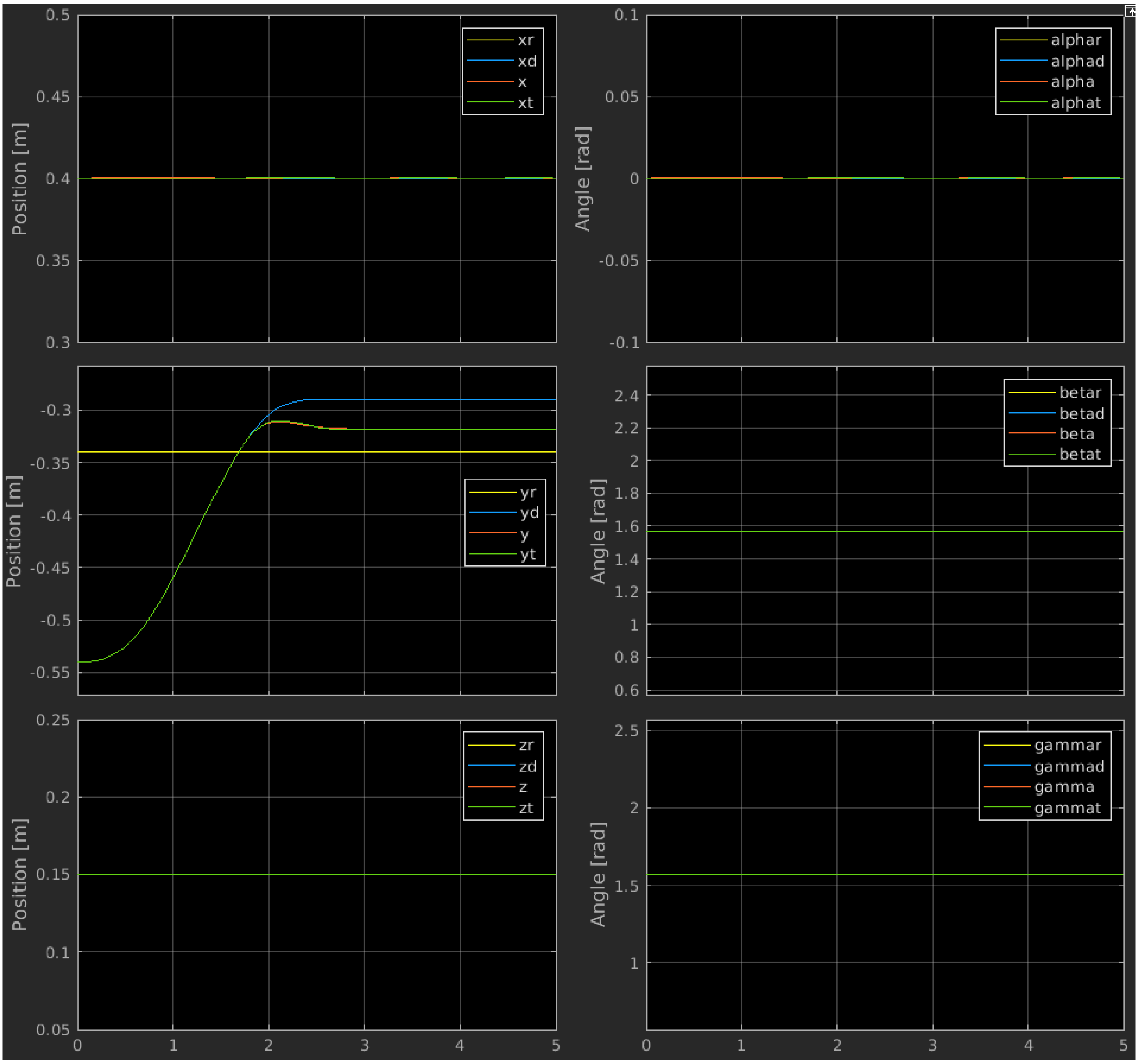


Figure 38: Admittance control

14 Assignment 14

14.1 Implement the force control with inner position loop

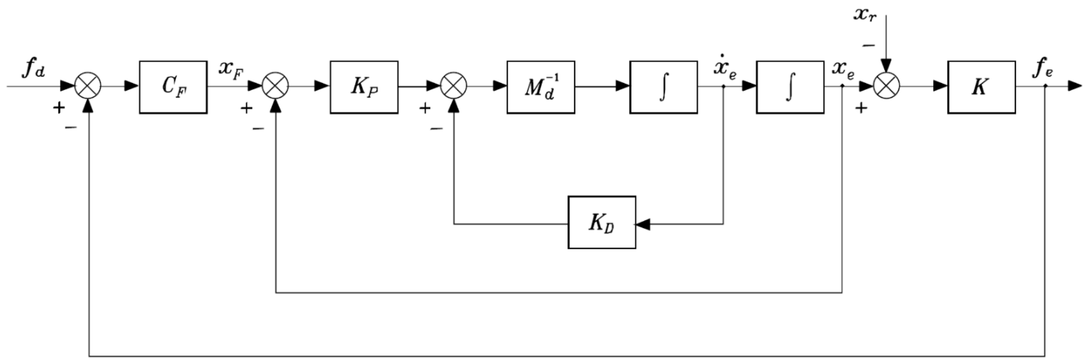


Figure 39: Force control with inner position loop architecture.

The architecture is based on an inverse dynamic position control surrounded by a force feedback loop. Given a desired constant force reference f_d we define a diagonal matrix C_F that acts like a compliant matrix, mapping force into position:

$$x_F = C_F(f_d - f_e)$$

where f_e is the measured interaction force.

The shape of the controller C_F is important. If C_F is a PI controller, i.e. $C_F = K_F + \frac{1}{s}K_I$, then the steady state error is null.

The architecture is modelled in SIMULINK as follows:

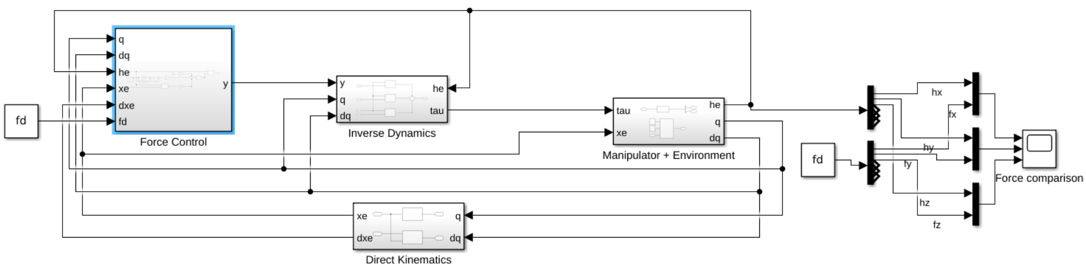


Figure 40: Force control with inner position loop SIMULINK model.

The architecture was tested with $f_d = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$, with an environment stiffness $K = 5\mathbb{I}_6$, PI gains for the force loop $K_I = 5\mathbb{I}_6$ and $K_F = 10\mathbb{I}_6$.

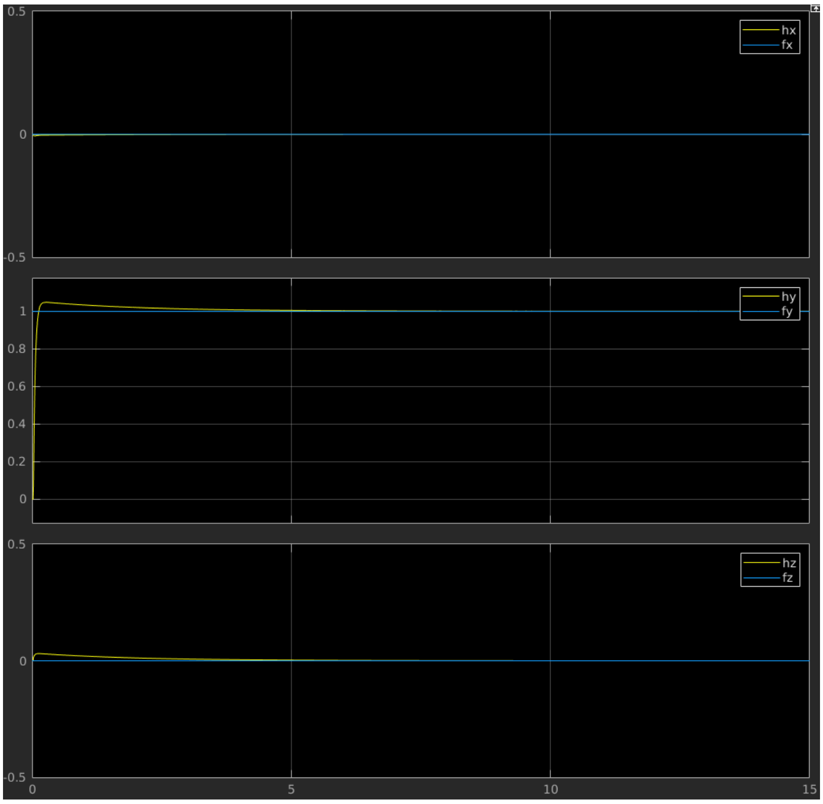


Figure 41: Force control with inner position loop - $C_F = K_F + \frac{1}{s}K_I$

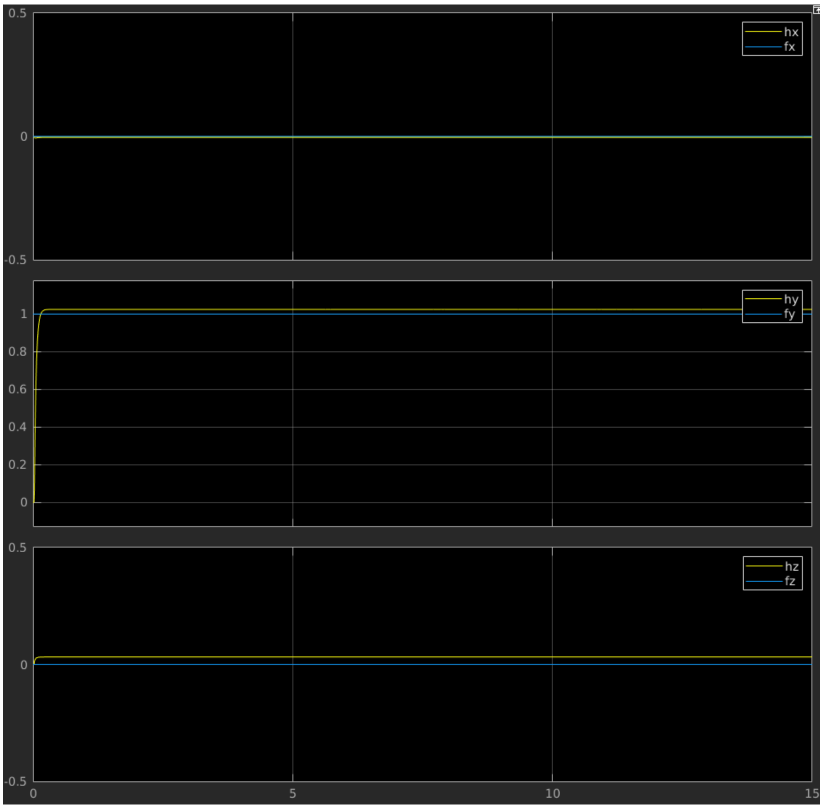


Figure 42: Force control with inner position loop - $C_F = K_F$

15 Assignment 15

15.1 Implement the parallel force/position control

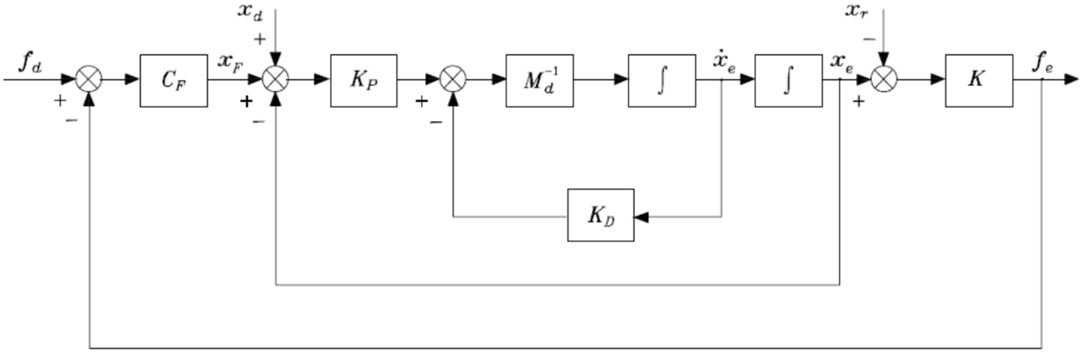


Figure 43: Parallel force/position control architectures.

The architecture is essentially the same as the force control with inner position loop. The main difference is that we actually introduce the desired end-effector pose x_d into the position loop.

The architecture is implemented in SIMULINK as follows:

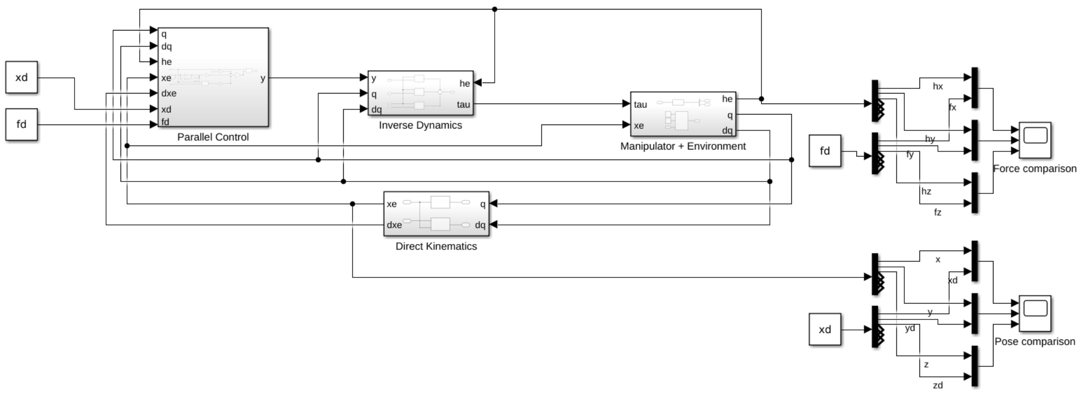


Figure 44: Parallel force/position control SIMULINK model.

The architecture was tested with $f_d = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, with an environment stiffness $K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, PI gains for the force loop $K_I = 15\mathbb{I}_6$ and $K_F = 15\mathbb{I}_6$. The desired pose is $k(\begin{bmatrix} -\pi/2 & -0.1 & -\pi/6 \end{bmatrix})$ while the environment is placed at $k(\begin{bmatrix} -\pi/2 & -0.2 & -\pi/6 \end{bmatrix})$.

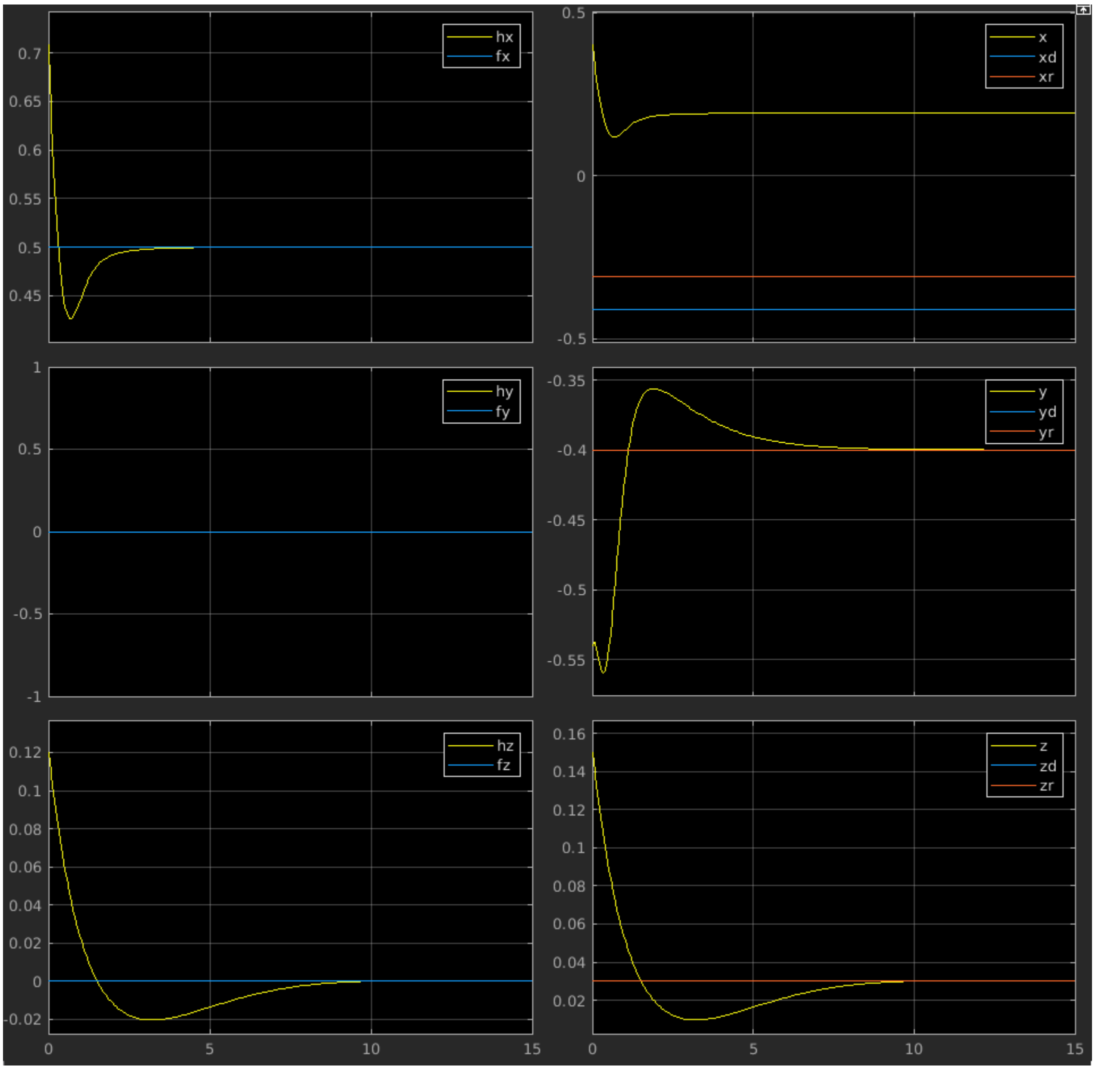


Figure 45: Parallel force/position control