UNIVERSITÀ DEGLI STUDI DI VERONA

Advanced Control Systems Report

COMPUTER ENGINEERING FOR ROBOTICS AND SMART INDUSTRY

Nicola Marchiotto - VR462317

2022

Contents

1 Homework 1

Description: Compute the DH Table, direct and inverse Kinematics, differential Kinematics of the assigned robot.

1.1 Robot Model

The following image represent the model of my PPP robot where the frames according to the Denavit Hartenberg convention are shown. The c parameters, all expressed in meters, are respectively $c_0 = 0.4$, $c_1 = 0.2$, $c_2 = 0.1498$, $c_3 = 0.2502$.

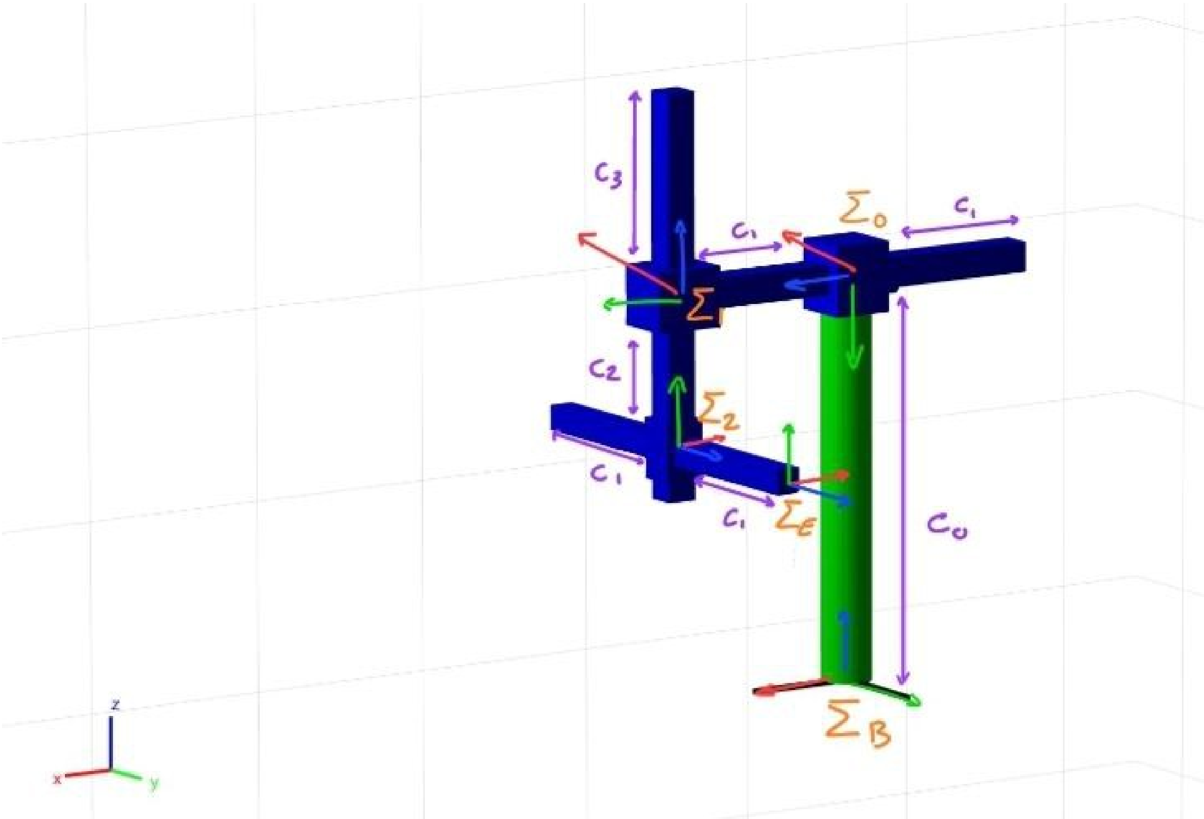


Figure 1: Robot Model

Each joint its characterized by its variable, $d1, d2, d3$, and they can assume the following range of values:

$$d1 \in [-0.2, 0.2] \quad d2 \in [-0.2502, 0.1498] \quad d3 \in [-0.2, 0.2]$$

1.2 DH Table

	$\boldsymbol{\theta}$	$\boldsymbol{\alpha}$	**a**	**d**
B-0	$-\pi/2$	$-\pi/2$	0	$0,4$
0-1	0	$\pi/2$	0	$d1+0.2$
1-2	$-\pi/2$	$\pi/2$	0	$d2-0.1498$
2-E	0	0	0	$d3+0.2$

Table 1: DH Table

1.3 Direct Kinematics

The following rotational matrices are the ones needed describe the change of coordinates from the frames Σ_0, Σ_1, Σ_2, Σ_E to the base frame Σ_B.

$$^BT_0 = \begin{pmatrix} 0 & 1 & 0 & d_1+0.2 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^BT_1 = \begin{pmatrix} 0 & 1 & 0 & d_1+0.2 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^BT_2 = \begin{pmatrix} -1 & 0 & 0 & d_1+0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & d_2+0.2502 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^BT_E = \begin{pmatrix} -1 & 0 & 0 & d_1+0.2 \\ 0 & 0 & 1 & d_3+0.2 \\ 0 & 1 & 0 & d_2+0.2502 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1.4 Inverse Kinematics

$$InvKin = \begin{pmatrix} x-0.2 \\ y-0.2 \\ z-0.2502 \\ -pi/2 \\ -pi/2 \\ -pi/2 \end{pmatrix}$$

1.5 Differential Kinematics

For the matrix Ta the zyz euler angles were assumed.

$$J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad Ja = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad T_a = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{pmatrix}$$

2 Homework 2

Description: Compute the Kinetic and Potential energy of the assigned robot considering only the links contribution and leaving out the motor one.

2.1 Kinetic energy

Since no information about the mass of each links was given, I assumed every link composed by the same material distributed in an homogeneous way. I expressed their masses in function of their Volumes and of a parametric density ρ. Here are reported the volumes of each link $[m^3]$:

$$V_1 = 0.3600 * 10^{-3} \quad V_2 = 0.3600 * 10^{-3} \quad V_3 = 0.2500 * 10^{-3}$$

Choosing arbitrary the density as $\rho = 8000\ kg/m^3$, we get the following masses for the links $[kg]$:

$$m_1 = 2.88 \quad m_2 = 2.88 \quad m_3 = 2$$

The following vectors represent the position of the center of mass of each link with respect to the base reference frame Σ_B in homogeneous coordinates.

$$p_{l1} = \begin{pmatrix} d_1 \\ 0 \\ 0.4 \\ 1 \end{pmatrix} \quad p_{l2} = \begin{pmatrix} d_1 + 0.2 \\ 0 \\ d_2 + 0.4 \\ 1 \end{pmatrix} \quad p_{l3} = \begin{pmatrix} d_1 + 0.2 \\ d_3 \\ d_2 + 0.2502 \\ 1 \end{pmatrix}$$

The following matrices represent the partial Jacobians, needed for the computation of the Kinetic energy. Since my robot is not affected by rotational motion, I considered only the first 3 rows of the Jacobian related to the linear velocities.

$$J_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad J_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad J_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Here is reported the formula of the kinetic energy in case of no rotational motion

$$T(q, \dot{q}) = \frac{1}{2}\dot{q}^T \left[\sum_{i=1}^{n} (m_{li}(J_i)^T(J_i)) \right] \dot{q} = \frac{1}{2}\dot{q}^T B(q)\dot{q}$$

Where B resulted as the following term

$$B = \begin{pmatrix} 7.7600 & 0 & 0 \\ 0 & 4.8800 & 0 \\ 0 & 0 & 2.0000 \end{pmatrix}$$

Finally the kinetic energy of the robot is expressed by the following equation:

$$T(q, \dot{q}) = 3.88\dot{d1}^2 + 2.44\dot{d2}^2 + \dot{d3}^2$$

2.2 Potential energy

Here are reported the formulas needed for the computation of the potential energy:

$$U_{li} = -m_{li} g_0^T p_{li} \quad U = \sum_{i=1}^{n} U_{li} \quad g_0 = \begin{pmatrix} 0 \\ 0 \\ -9.81 \\ 1 \end{pmatrix}$$

The potential energy of the robot is expressed by the following equation:

$$47.873d2 + 27.5113$$

3 Homework 3

Description: Compute the equation of motion following the Lagrangian formulation.

3.1 Theory recap

The Lagrangian is given by

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) = \frac{1}{2}\dot{q}^T B(q)\dot{q} + \sum_{i=1}^{n} m_{li} g_0^T p_{li}$$

The generalized forces τ performing work on the generalized coordinates q are given by solving the following set of differential equations

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right)^T - \left(\frac{\partial L}{\partial q}\right)^T = \tau$$

Using the Christoffel symbols of the first type we end up with the following equation for computing the τ_i term

$$\sum_{j=1}^{n} b_{ij(q)}\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} + \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j - \sum_{i=1}^{n} m_{li} g_0^T \frac{\partial p_{li}}{\partial q_i} = \tau_i$$

3.2 Lagrangian formulation

Here are reported the expression of the B, C, g and τ for my PPP serial link manipulator computed using the Lagrangian formulation.

$$B = \begin{pmatrix} 7.7600 & 0 & 0 \\ 0 & 4.8800 & 0 \\ 0 & 0 & 2.0000 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad g = \begin{pmatrix} 0 \\ 47.8728 \\ 0 \end{pmatrix}$$

$$\tau = \begin{pmatrix} 7.7600\,\ddot{d}_1 \\ 4.8800\,\ddot{d}_2 + 47.8728 \\ 2\,\ddot{d}_3 \end{pmatrix}$$

4 Homework 4

Description: Compute the RNE formulation.

4.1 Theory recap

Here are reported the forward and backward equations in case of a prismatic joint, without considering friction forces and the inertia of the motors

Forward equations

- Initial conditions: $w_0 = [0, 0, 0]^T, \dot{w}_0 = [0, 0, 0]^T, \ddot{p}_0 = [0, -9.81, 0]^T$
- $R_i^i = R_i^i(d_i)$
- $w_i^i = (R_i^{i-1})^T w_{i-1}^{i-1}$
- $\dot{w}_i^i = (R_i^{i-1})^T \dot{w}_{i-1}^{i-1}$
- $\ddot{p}_i^i = (R_i^{i-1})^T \ddot{p}_{i-1}^{i-1} + \dot{w}_i^i \times r_{i-1,i}^i + w_i^i \times (w_i^i \times r_{i-1,i}^i) + (R_i^{i-1})^T \ddot{d}_i z_0 + 2\dot{d}_i w_i^i \times ((R_i^{i-1})^T z_0)$
- $\ddot{p}_{c_i}^i = \ddot{p}_i^i + \dot{w}_i^i \times r_{i,c_i}^i + w_i^i \times (w_i^i \times r_{i,c_i}^i)$

Backward equations

- Initial conditions: $h_e = [f_{n+1}, \mu_{n+1}]^T, f_{n+1}^{n+1} = f_{n+1}, \mu_{n+1}^{n+1} = \mu_{n+1}$
- $R_{i+1}^i = R_{i+1}^i(d_{i+1})$
- $f_i^i = R_{i+1}^i f_{i+1}^{i+1} + m_i \ddot{p}_{c_i}^i$
- $\mu_i^i = -f_i^i \times (r_{i-1,i}^i + r_{i,c_i}^i) + R_{i+1}^i \mu_{i+1}^{i+1} + R_{i+1}^i \mu_{i+1}^{i+1} \times r_{i,c_i}^i$
- $\tau = (f_i^i)^T (R_i^{i-1})^T z_0$

4.2 RNE formulation

Here are reported the expression of the B, Cdq, g and τ for my PPP serial link manipulator using the RNE formulation.

$$B = \begin{pmatrix} 7.7600 & 0 & 0 \\ 0 & 4.8800 & 0 \\ 0 & 0 & 2.0000 \end{pmatrix} \quad C * dq = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad g = \begin{pmatrix} 0 \\ 47.8728 \\ 0 \end{pmatrix}$$

$$\tau = \begin{pmatrix} 7.7600\, \ddot{d}_1 \\ 4.8800\, \ddot{d}_2 + 47.8728 \\ 2\, \ddot{d}_3 \end{pmatrix}$$

5 Homework 5

Description: Compute the dynamic model in the operational space.

5.1 Theory recap

Equations for the operational space dynamic model

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

Equations for the non -redundant manipulator in a non singular configuration

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

5.2 Operational Space Dynamic Model

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

Here are reported the expression of the $B_A, C_A\dot{x}$ and g_A for my PPP serial link manipulator.

$$B_A = \begin{pmatrix} 7.7600 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.8800 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad C_A = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad g_A = \begin{pmatrix} 0 \\ 0 \\ 47.8728 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$u = \begin{pmatrix} 7.7600\ddot{x}1 + u_{e1} \\ 2\ddot{x}2 + u_{e2} \\ 4.8800\ddot{x}3 + 47.8728 + u_{e3} \\ u_{e4} \\ u_{e5} \\ u_{e6} \end{pmatrix}$$

6 Homework 6

6.1 Joint Space PD Control with Gravity Compensation

Description: Design the Joint Space PD control law with gravity compensation

The control law is reported in the following equation

$$\tau = g(q) + K_p(q_d - q) - K_d\dot{q}$$

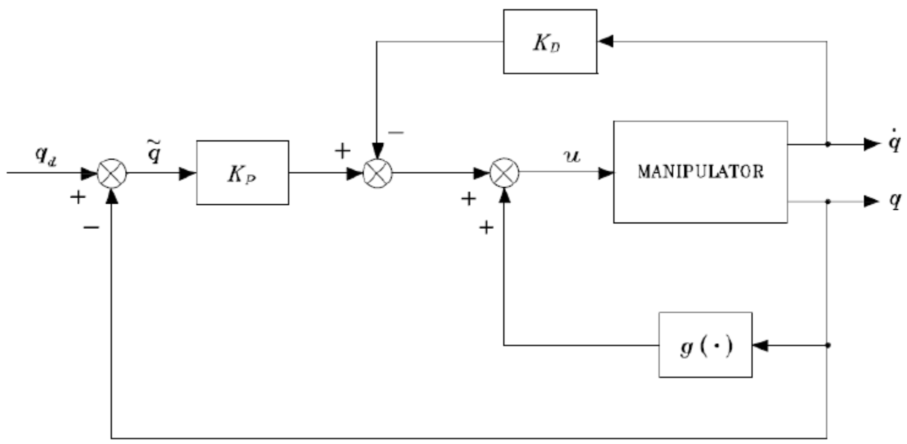


Figure 2: Scheme for the Joint Space PD control law with gravity compensation

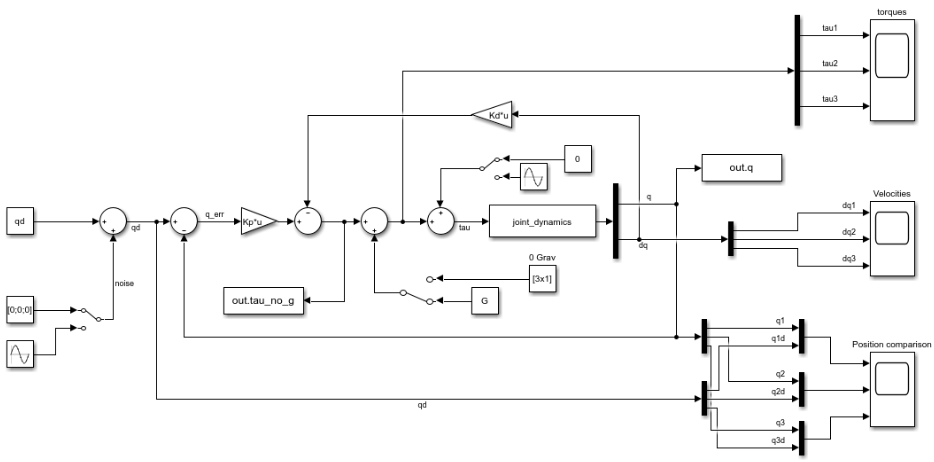


Figure 3: Implementation of the Joint Space PD control law with gravity compensation

The following scopes show the trajectories of and the torques provided to the various joints, the start position is $[0; 0; 0]$ and the final is set to $[0.1; 0.1; 0.15]$

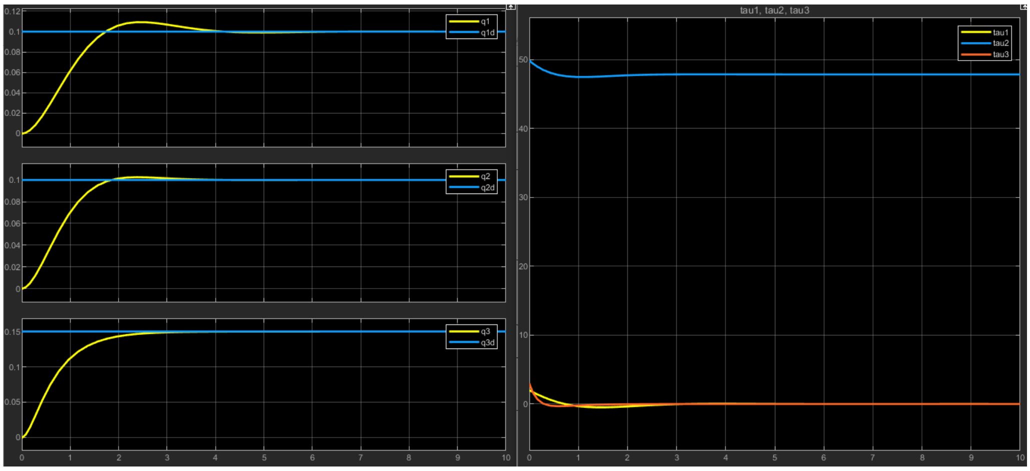


Figure 4: Position and torque, gravity compensation active

As can be seen, the desired pose is reached and the torque necessary to compensate gravity is provided just to the second joint, the only one to be affected by it.

If the gravity compensation is not taken into account, the torque provided to the second joint will be less than the required to overcame the gravity force, consecutively the link E.E. will ”fall”.

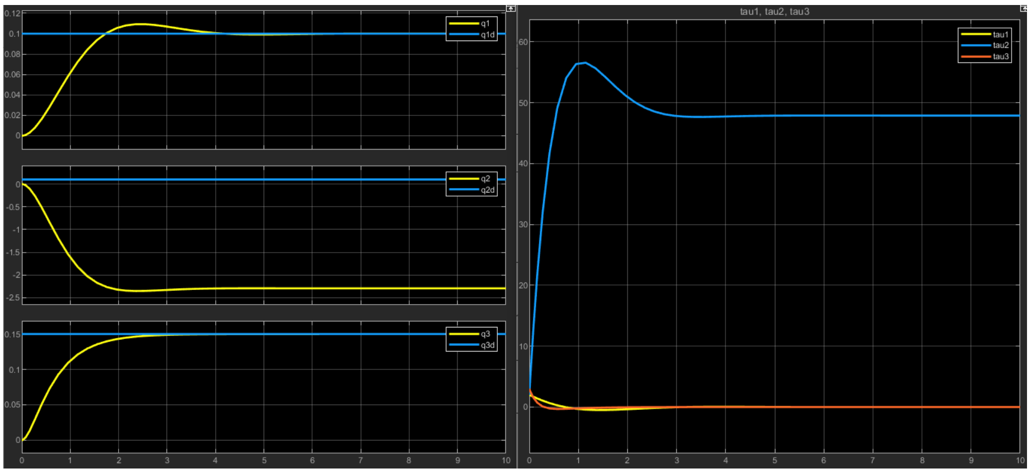


Figure 5: Position and torque, gravity compensation not active

It is worth noticing that, since my robot is of the PPP type, the gravity term does not dependent on the configuration but it is constant: $G = [0; 47.8728; 0]$

If the desired position is not constant, i.e. $q_d = sin(wt)$, we get the following behaviour

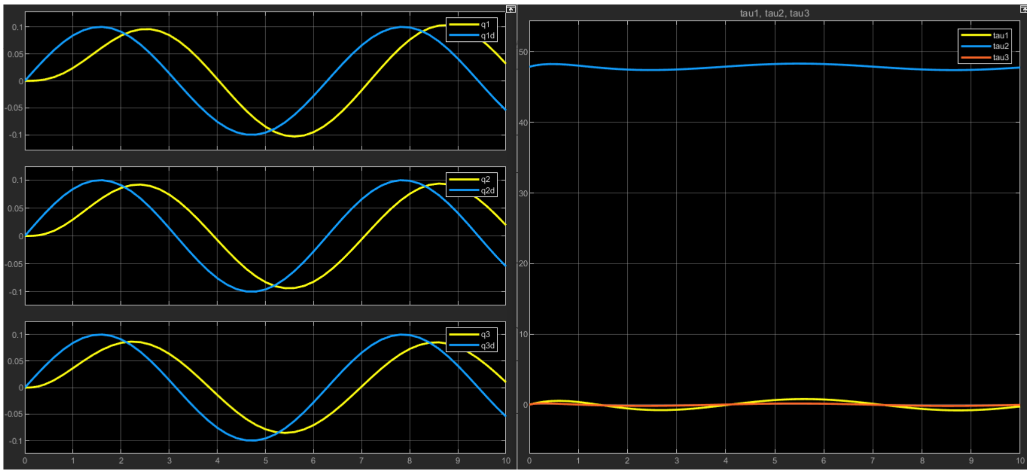


Figure 6: Position and torque, position not constant

7 Homework 7

7.1 Joint Space Inverse Dynamics Control Law

Description: Design the Joint Space Inverse Dynamics Control Law

The control law is reported in the following equations

$$\tau = B(q)[\ddot{q} + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)] + n(q, \dot{q}) \qquad n(q, \dot{q}) = C(q, \dot{q})\dot{q} + F\dot{q} + g(q)$$

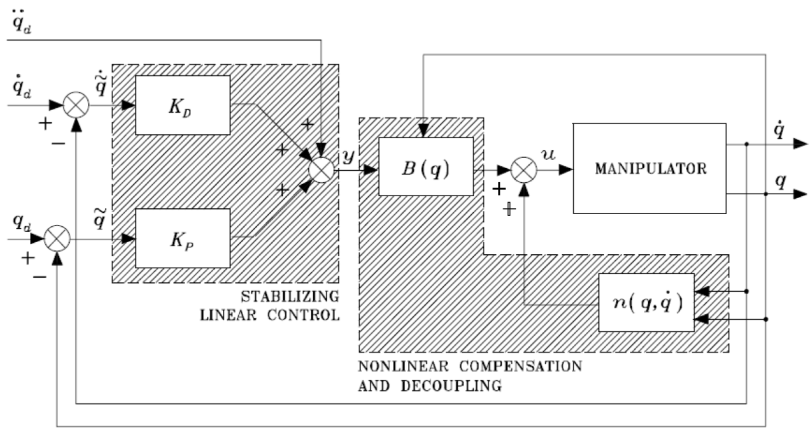


Figure 7: Scheme for the Joint Space Inverse Dynamics Control Law

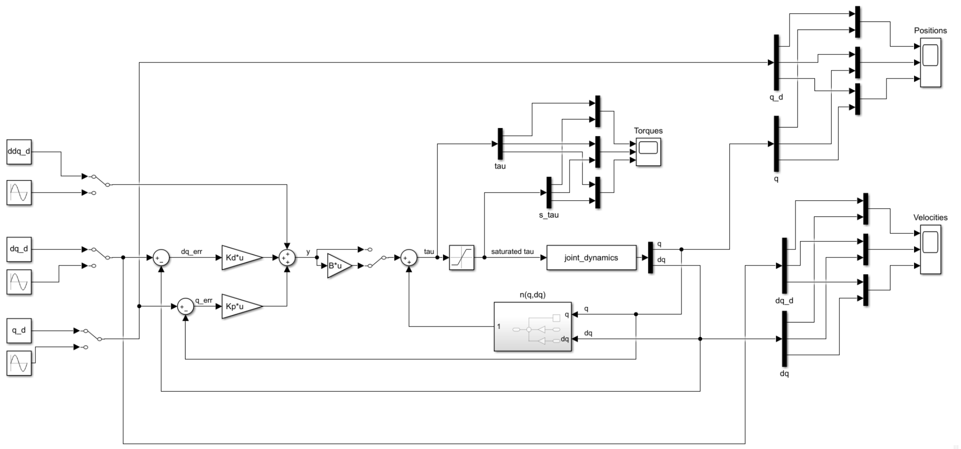


Figure 8: Implementation of the Joint Space Inverse Dynamics Control Law

The following scopes represent position and velocities of the various joints, providing as reference input a multi point trajectory designed in the joint space joint by joint using cubic splines, The way points for each joints where set respectively to:

$$qk_1 = [0, 0.1, 0.2] \quad qk_2 = [0, 0.1, 0.2] \quad qk_3 = [0, -0.1, -0.2]$$

Torques above $50N * m$ and below $-50N * m$ were saturated in the simulation

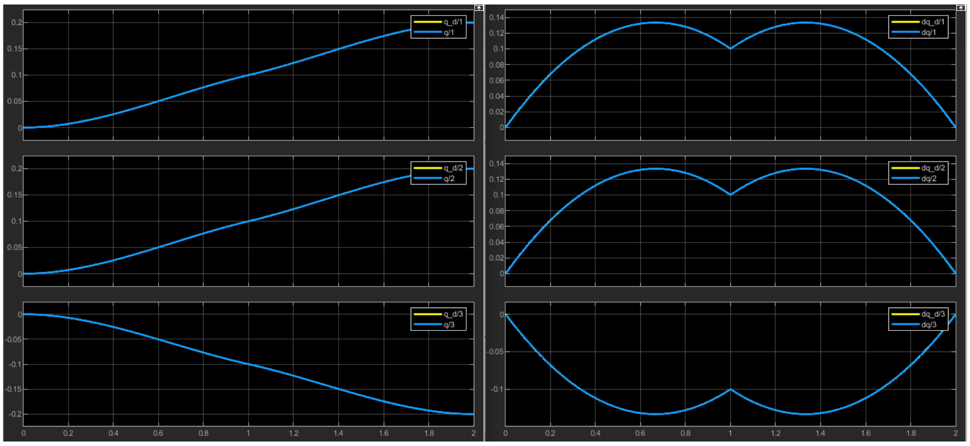


Figure 9: Inverse Dynamics, positions and velocities

As can be seen, a perfect match was successfully achieved

If different values for the "true" B, C and g are chosen in the control law, these latter computed by changing the value of the masses for the links, we get the following behaviour:

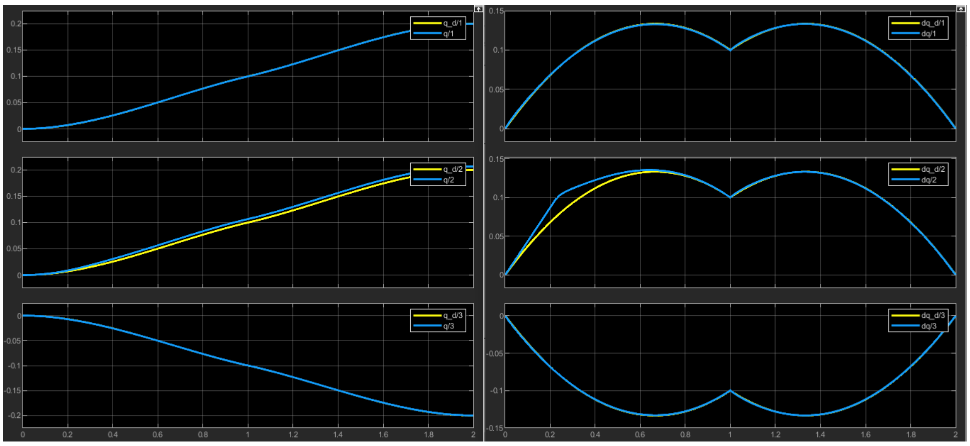


Figure 10: Inverse Dynamics, positions and velocities with not "true" B, C and g in the control law

The second joint, the only one affect by gravity, is not perfectly able to follow the reference.

8 Homework 8

8.1 Adaptive Control

Description: Implement the Adaptive Control law for the 1-DoF link under gravity

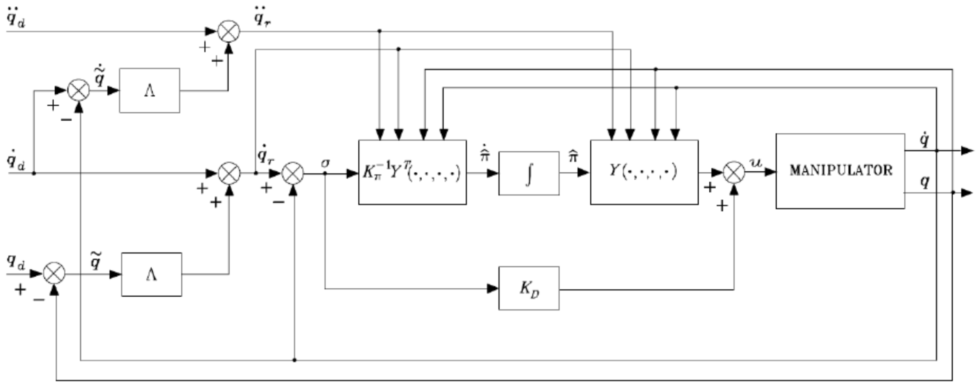


Figure 11: Joint Space Adaptive Control Law Scheme

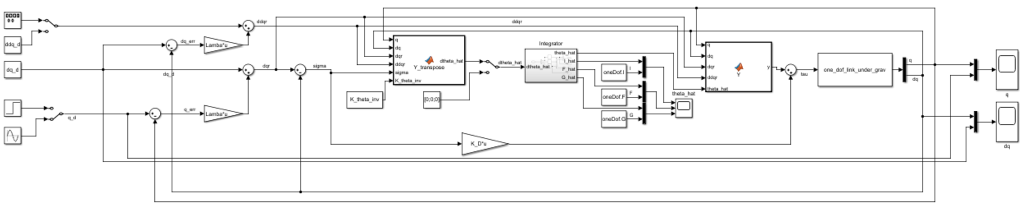


Figure 12: Implementation of Joint Space Adaptive Control Law Scheme

8.2 Theory Recap

- Model plant

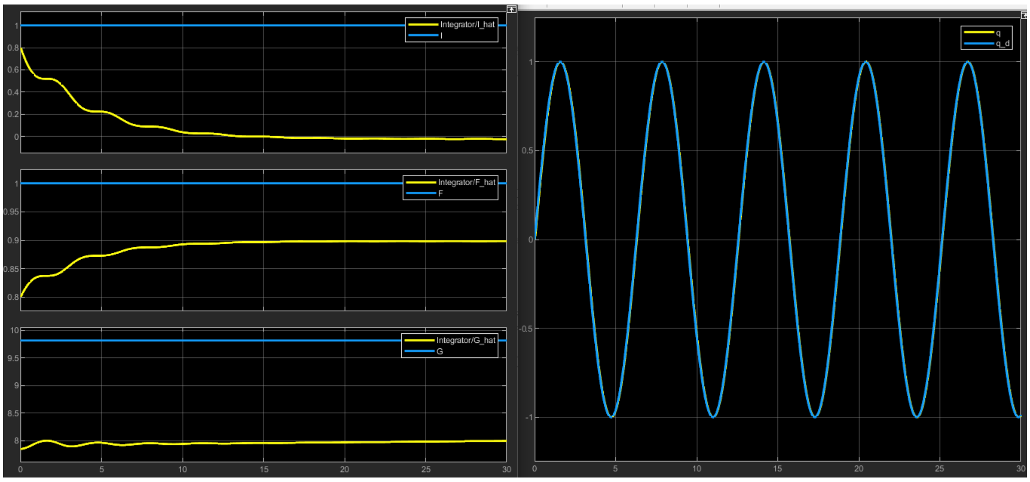
$$I\ddot{q} + F\dot{q} + Gsin(q) = \tau$$

- Linear parameterization

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

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

The following scopes represent the parameter estimation $\hat{\theta}$ and the position of the one d.o.f. link, the provided reference position was a sinusoidal wave of amplitude 1 and the initial estimation of the parameters was set as 0.8% of the real ones



9 Homework 9

9.1 Operational Space PD Control with gravity compensation

Description: Design the Operational Space PD control law with gravity compensation

The control law is reported in the following equations

$$\tau = g(q) + J_A^T(q)K_P(x_d - d) - J_A^T(q)K_D J_A(q)\dot{q}$$

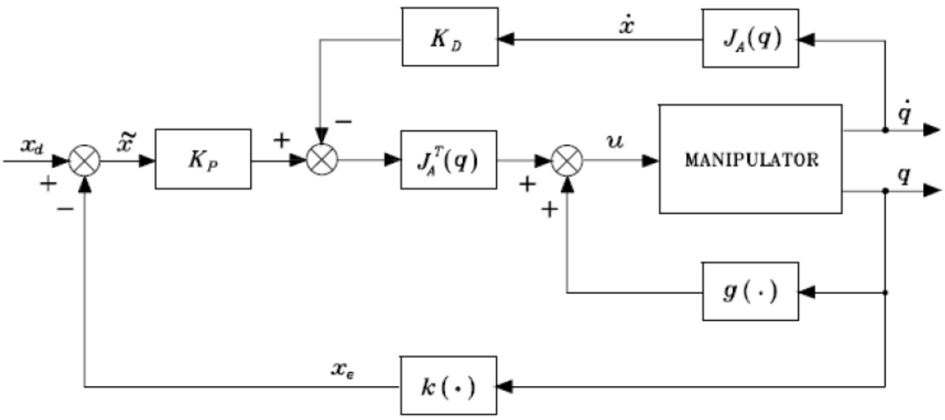


Figure 13: Scheme for the Operational Space PD control law with gravity compensation

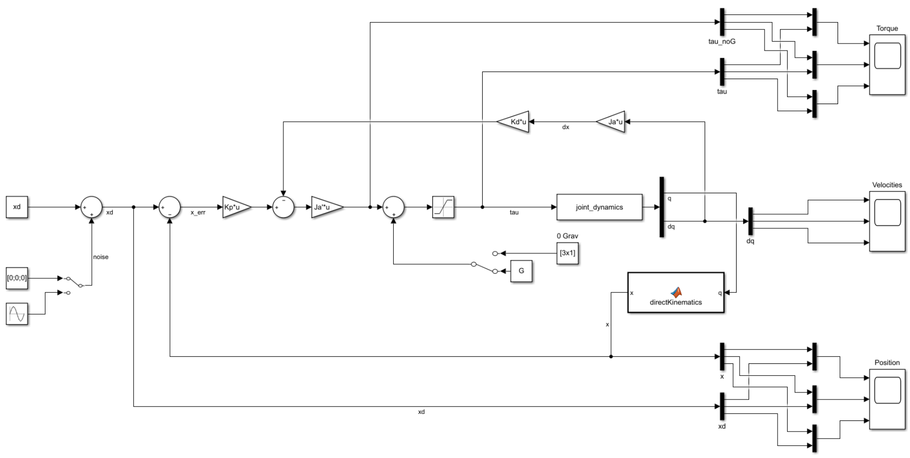
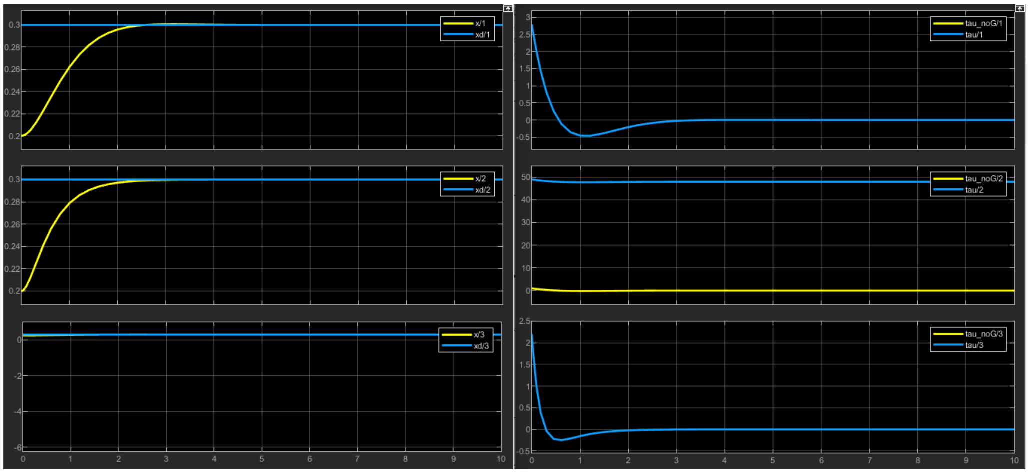


Figure 14: Implementation of the Operational Space PD control law with gravity compensation

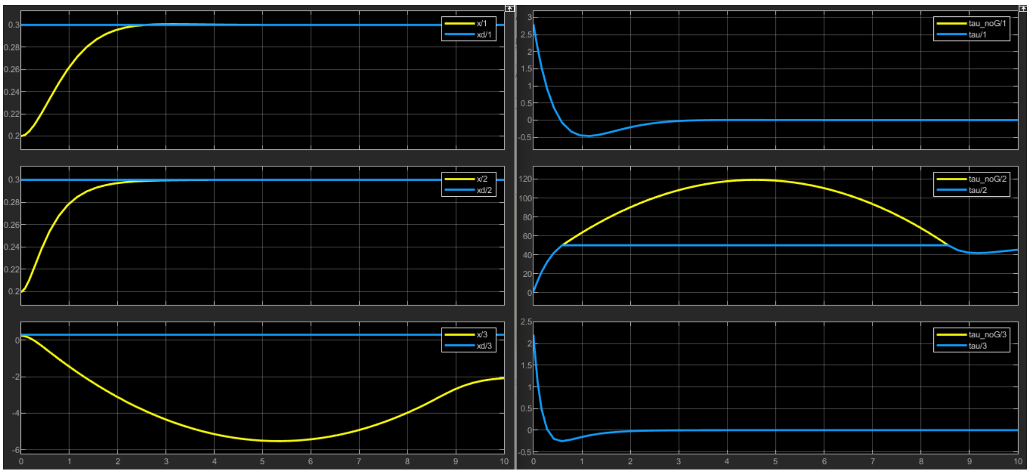
The operational space E.E. starting position is $[0.2; 0.2; 0.2502]$, the following scopes represent position and torques applied to the joint during the motion, the desired position was set to $[0.3; 0.3; 0.3]$.

Torques above $50N * m$ and below $-50N * m$ were saturated in the simulation



As can be seen, the desired pose is reached and the torque necessary to compensate gravity is provided just to the second one, the only to be affected by it.

If the gravity compensation is not taken into account, the torque provided to the second joint will be less than the required to overcame the gravity force, consecutively the link E.E. will "fall", as can be seen from the third plot of the position graph representing the z axis



10 Homework 10

10.1 Operational Space Inverse Dynamic Control law

Description: Design the Operational Space Inverse Dynamic Control law

The control law is reported in the following equations

$$\tau = B(q)[J_a^{-1}(q)(\ddot{x}_d + K_D\dot{\tilde{x}}_d + K_P\tilde{x} - \dot{J}_A(q,\dot{q})\dot{q})] + n(q,\dot{q}) \qquad n(q,\dot{q}) = C(q,\dot{q})\dot{q} + F\dot{q} + g(q)$$

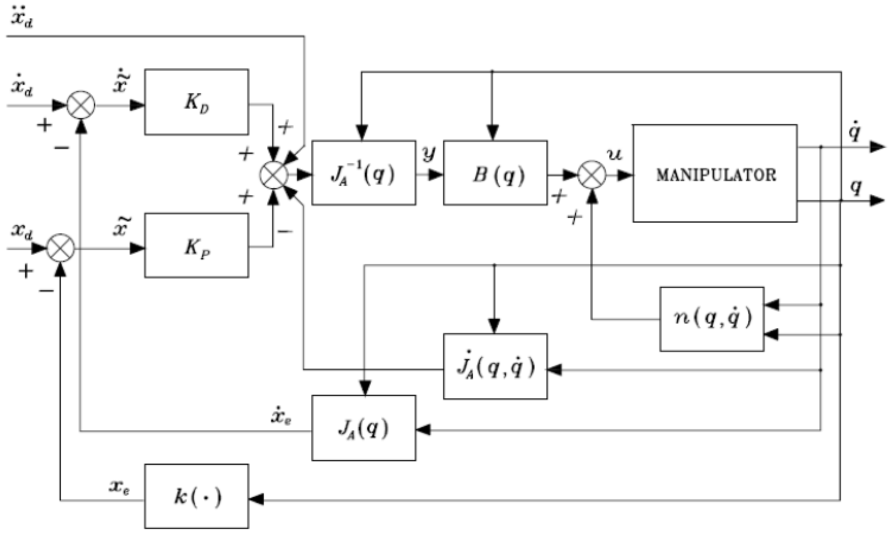


Figure 15: Scheme for the Operational Space Inverse Dynamic Control law

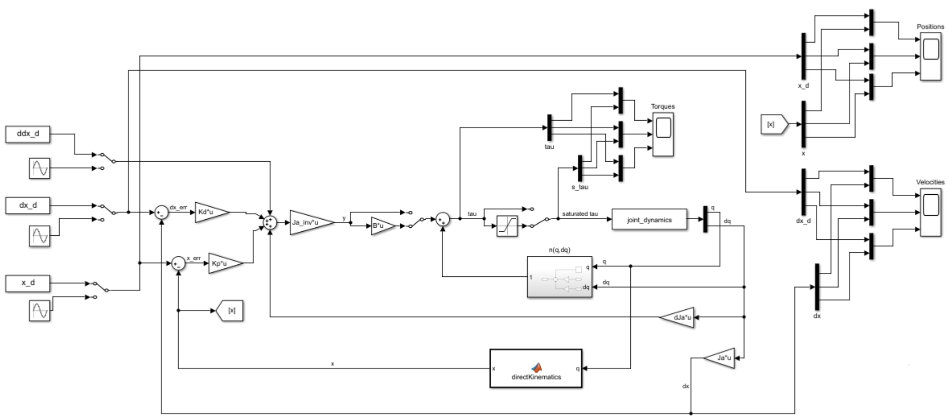
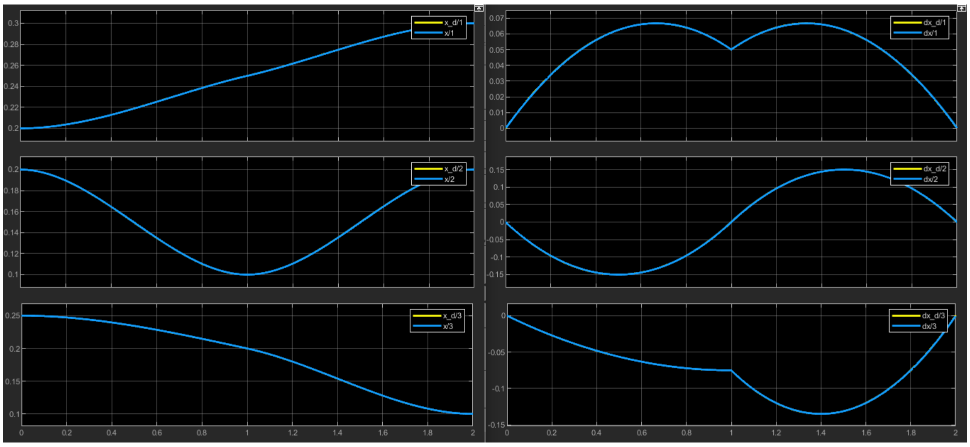


Figure 16: Implementation of the Operational Space Inverse Dynamic Control law

The operational space E.E. starting position is $[0.2; 0.2; 0.2502]$, the following scopes represent position and velocities of along the x, y, z during the motion, the desired trajectory was designed using multi point trajectories composed by cubic splines, considering a single d.o.f. each time, the way points for each dimension were:

- $x = [0.2, 0.25, 0.3]$
- $y = [0.2, 0.1, 0.2]$
- $z = [0.2502, 0.2, 0.1]$

Since my robot is a PPP, the orientation d.o.f. are omitted. Initial and final velocity for each spline were set to 0



11 Homework 11

11.1 Compliance Control

Description: Study the Compliance Control

The control law is reported in the following equations

$$\tau = g(q) + J_{A_d}^T(q, \tilde{x})(K_P \tilde{x} - K_D(q, \tilde{x})\dot{q})$$

$$\tilde{x} = x_d - x_e = -\begin{bmatrix} o_{d,e}^d \\ \phi_{d,e} \end{bmatrix} \qquad J_{A_d} = T_a^{-1}(\phi_{d,e}) \begin{bmatrix} R_d^T & 0 \\ 0 & R_d^T \end{bmatrix} J(q)$$

The plane of contact was considered as an elastically compliant frame. The external wrench was computed only in the direction of contact, y for my robot, by considering the displacement between the environmental rest position and the end effector position

$$h_e = K x_{r,e}$$

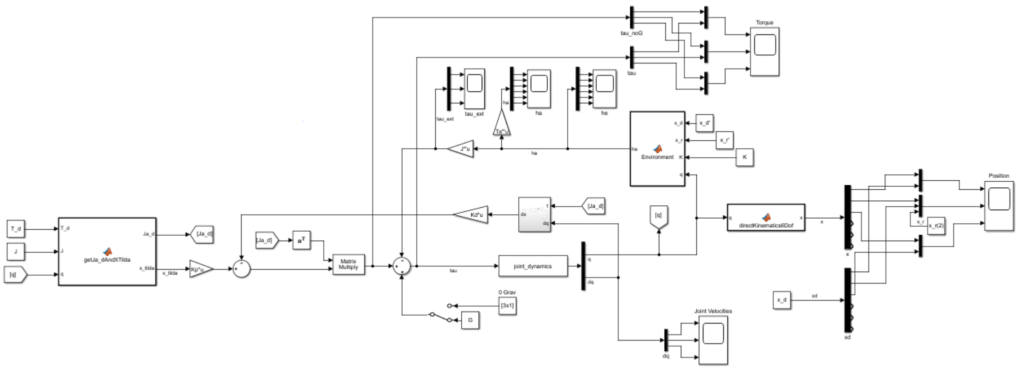


Figure 17: Implementation of the Compliance Control

The operational space E.E. starting position is $[0.2; 0.2; 0.2502]$, the desired position was set to $xdr = [0.3; 0.5; 0.3]$. The following scopes represent the position of the end effector, in the first one the robot does not touch the environment, in the second one it does touch the environment.

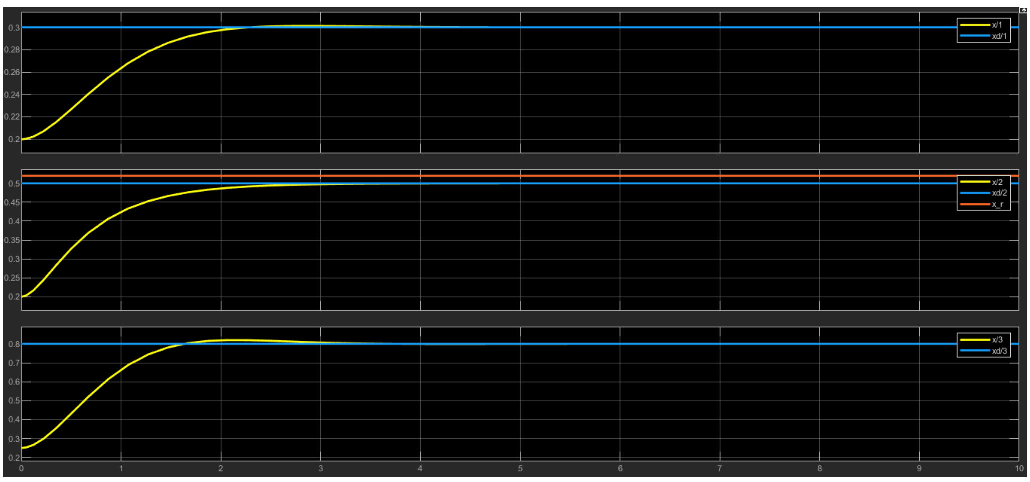


Figure 18: E.E. position, rest position set to $x_r = 0.51$ along the y direction

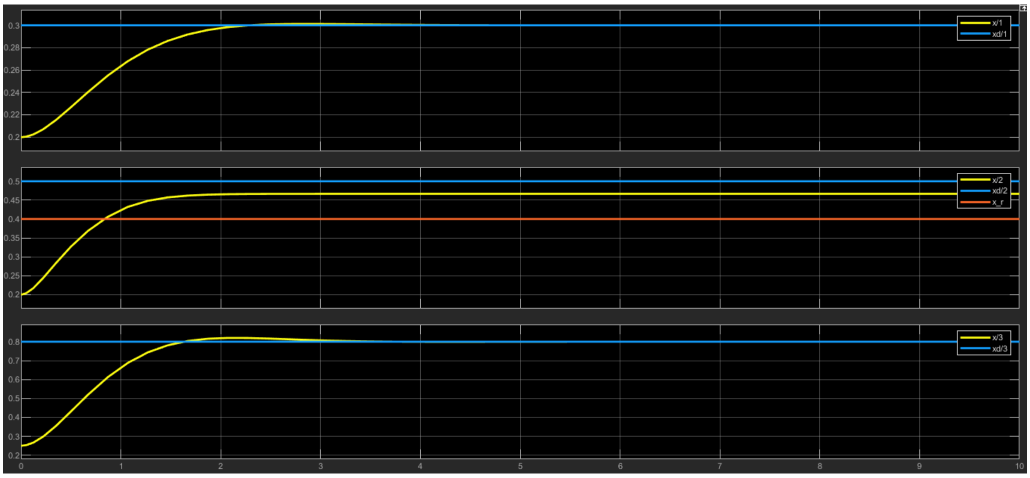


Figure 19: E.E. position, rest position set to $x_r = 0.4$ along the y direction

12 Homework 12

12.1 Impedance Control

Description: Implement the impedance control in the operational space

The control law is described by the following equations

$$\tau = B(q)y + n(q, \dot{q}) + J^T(q)h_e$$

$$y = J_{A_d}^{-1} M_d^{-1}(K_D\dot{\tilde{x}} + K_P\tilde{x} - M_d\dot{J}_{A_d}(q,\dot{q})\dot{q} - M_d\dot{b}(\tilde{x}, R_d, \dot{o}_d, w_d) - h_e^d)$$

The plane of contact was considered as an elastically compliant frame. The external wrench was computed only in the direction of contact, y for my robot, by considering the displacement between the environmental rest position and the end effector position

$$h_e = Kx_{r,e}$$

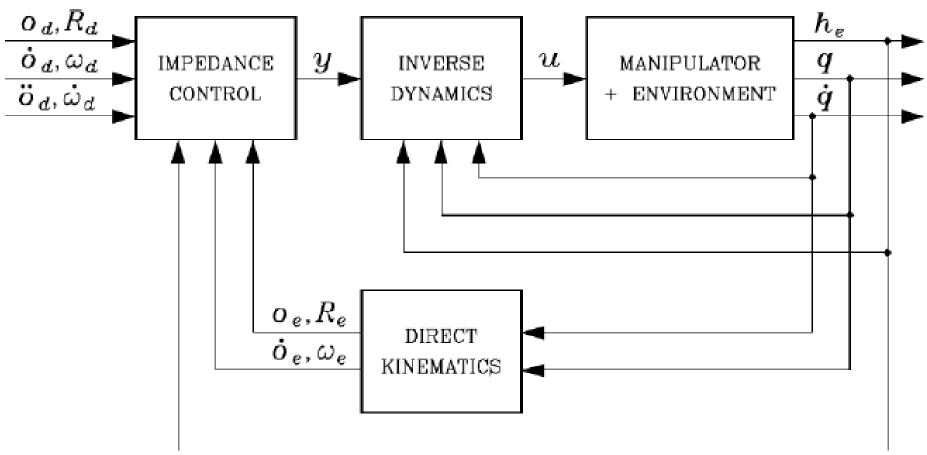


Figure 20: Scheme for the Impedance Control

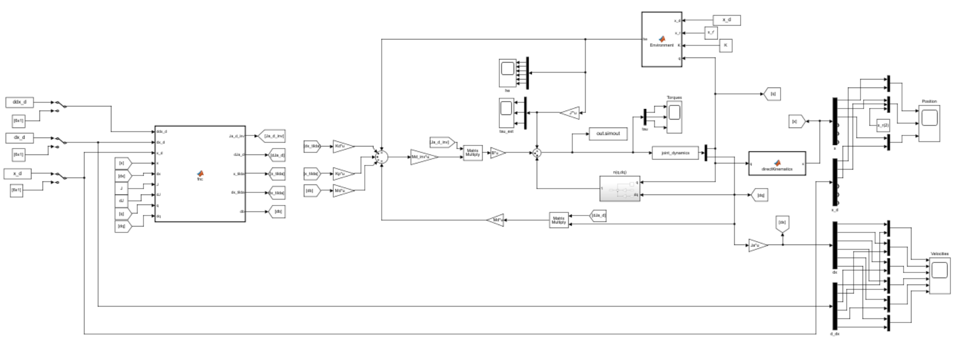


Figure 21: Implementation of the Impedance Control

The operational space E.E. starting position is $[0.2; 0.2; 0.2502]$. The following scopes represent the position of the end effector. In the first one the environmental rest position was set at $x_r = [0.2; 0.26; 0.2502]$ and the robot does not touch the environment, in the second one the environmental rest position was set at $x_r = [0.2; 0.23; 0.2502]$ and the robot does touch the environment.

The desired trajectory was designed using multi point trajectories composed by cubic splines, considering a single d.o.f. each time, the way points for each dimension were:

- $x = [0.2, 0.25, 0.3, 0.3, 0.3, 0.3]$
- $y = [0.2, 0.22, 0.25, 0.25, 0.25, 0.25]$
- $z = [0.2502, 0.22, 0.3, 0.3, 0.3, 0.3]$

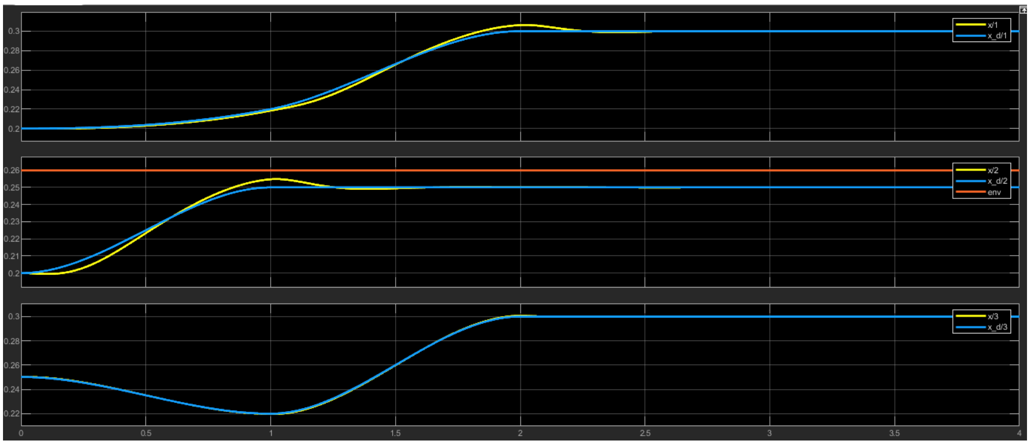


Figure 22: E.E. position, environmental rest position set to $x_r = [0.2; 0.26; 0.2502]$

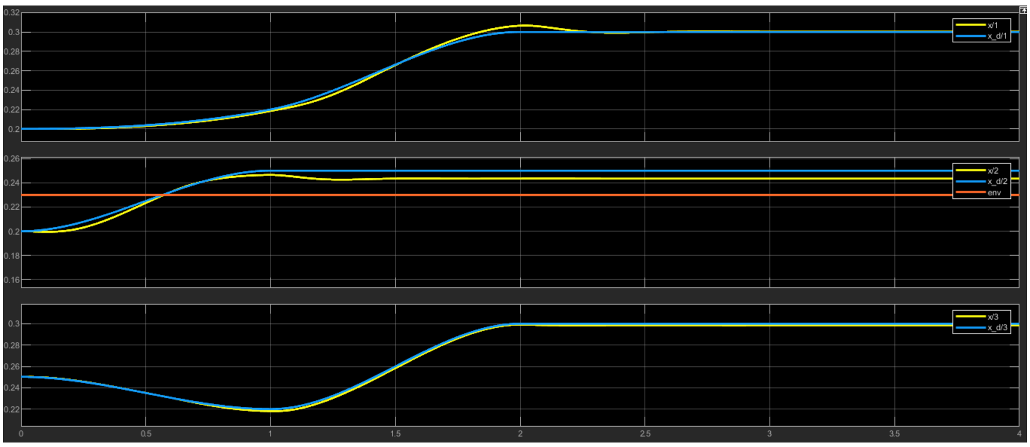


Figure 23: E.E. position, environmental rest position set to $x_r = [0.2; 0.23; 0.2502]$

13 Homework 13

13.1 Admittance Control

Description: Implement the admittance control in the operational space

The control law is described by the following equations

$$\tau = B(q)y + C(q,\dot{q})\dot{q} + g(q) + J(q)^T h_e$$
$$y = J_a^{-1}(q)(\ddot{x}_t + K_D\dot{\tilde{x}}_t + K_P\tilde{x} - \dot{J}_A(q,\dot{q})\dot{q})$$
$$M_t\ddot{\tilde{z}} + K_{Dt}\dot{\tilde{z}} + K_{Pt}\tilde{z} = h_e \quad \tilde{z} = x_d - x_t = -\begin{bmatrix} o_{d,t}^d \\ \phi_{d,t} \end{bmatrix}$$

where M_t, K_{Dt}, K_{Pt} are the parameters of the mechanical impedance, h_e^d is the measured interaction force and $\tilde{z}$ is the operational space error between the desired frame Σ_d and the compliant frame Σ_t

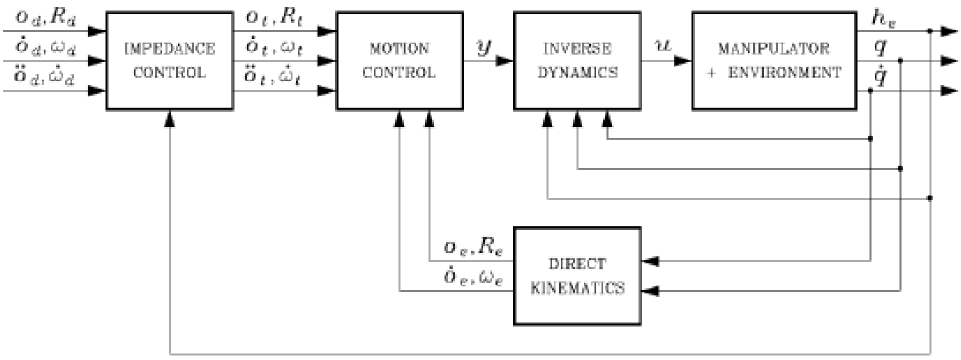


Figure 24: Scheme for the Admittance Control

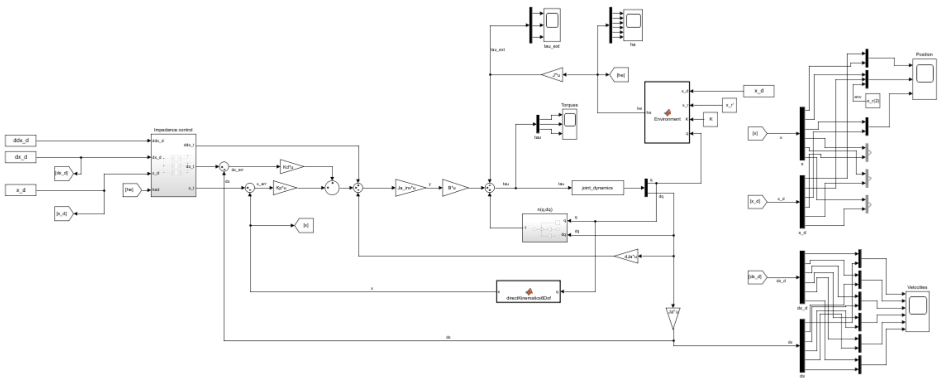


Figure 25: Implementation of the Admittance Control

The operational space E.E. starting position is $[0.2; 0.2; 0.2502]$. The following scopes represent the position of the end effector, in the first one the environmental rest position was set at $x_r = [0.3; 0.31; 0.3]$ and the robot does not touch the environment, in the second one the environmental rest position was set at $x_r = [0.3; 0.26; 0.3]$ and the robot does touch the environment.

The desired trajectory was designed using multi point trajectories composed by cubic splines, considering a single d.o.f. each time, the way points for each dimension were:

- $x = [0.2, 0.25, 0.3, 0.3, 0.3, 0.3]$
- $y = [0.2, 0.22, 0.3, 0.3, 0.3, 0.3]$
- $z = [0.2502, 0.22, 0.3, 0.3, 0.3, 0.3]$

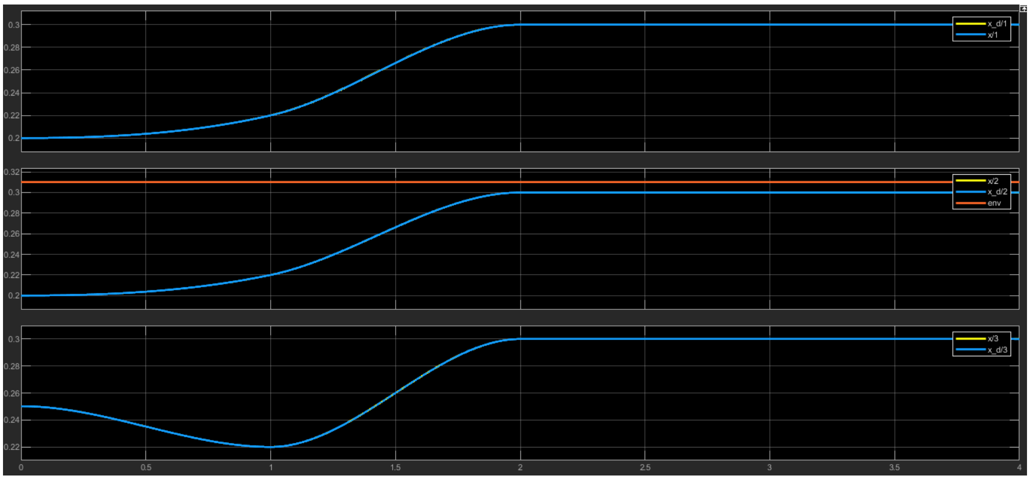


Figure 26: E.E. position, environmental rest position set to $x_r = [0.3; 0.31; 0.3]$

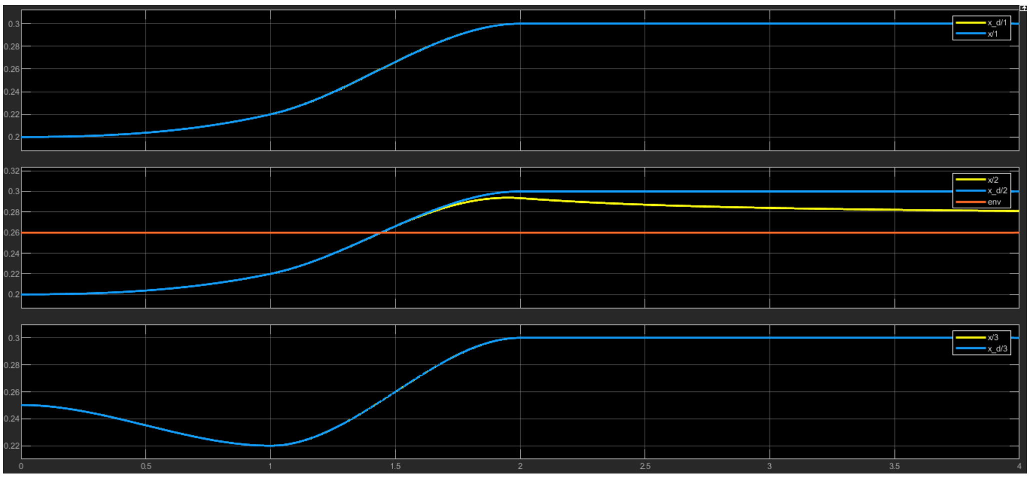


Figure 27: E.E. position, environmental rest position set to $x_r = [0.3; 0.26; 0.3]$

14 Homework 14

14.1 Force Control

Description: Implement the force control with the inner position loop

The control law is described by the following equations

$$\tau = B(q)y + C(q,\dot{q})\dot{q} + g(q) + J(q)^T h_e$$
$$y = J^{-1}(q)M_d^{-1}(-K_D\dot{x}_e + K_P(x_F - x_e) - M_d\dot{J}(q,\dot{q})\dot{q})$$
$$x_F = C_F(f_d - f_e) \quad f_e = K(x_e - x_r) \quad C_F = K_F + \frac{K_I}{s}$$

x_F is a suitable reference position to be related to the force error, f_d is the desired constant force and f_e is the measured interaction force. The reported schema presents a C_F with both proportional and integral actions, in this way at steady state $f_e = f_d$

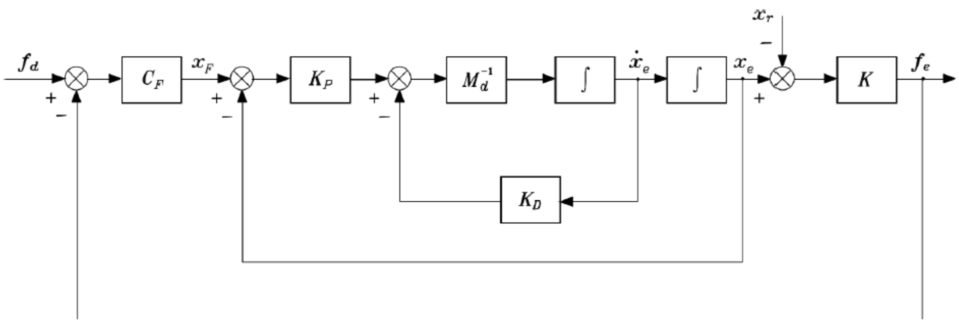


Figure 28: Scheme for the Force Control with inner position loop

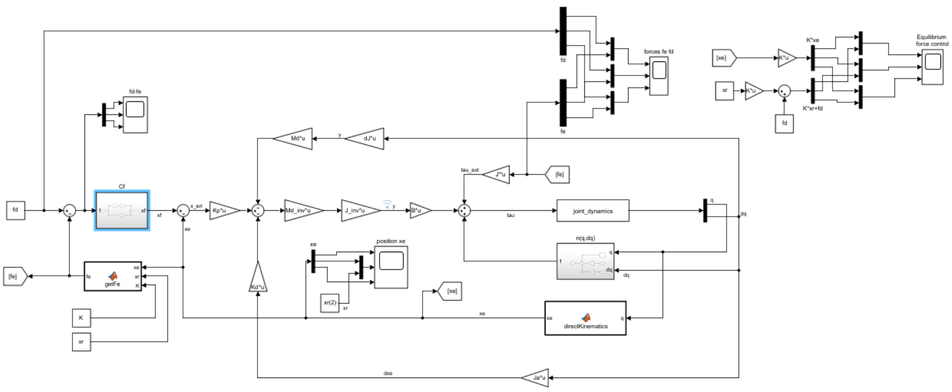


Figure 29: Implementation of the Force Control with inner position loop

The operational space E.E. starting position is $[0.2; 0.2; 0.2502]$. The following scopes represent the position of the end effector, and the interaction and desired force f_e and f_d. The environmental rest position was set to 0.3 along y. The environment was modelled ad an elastic system $f_e = K(x_e - x_r)$ only along the direction of contact of the E.E., for my robot it is y. The desired force was set to $f_d = 1$ along y.

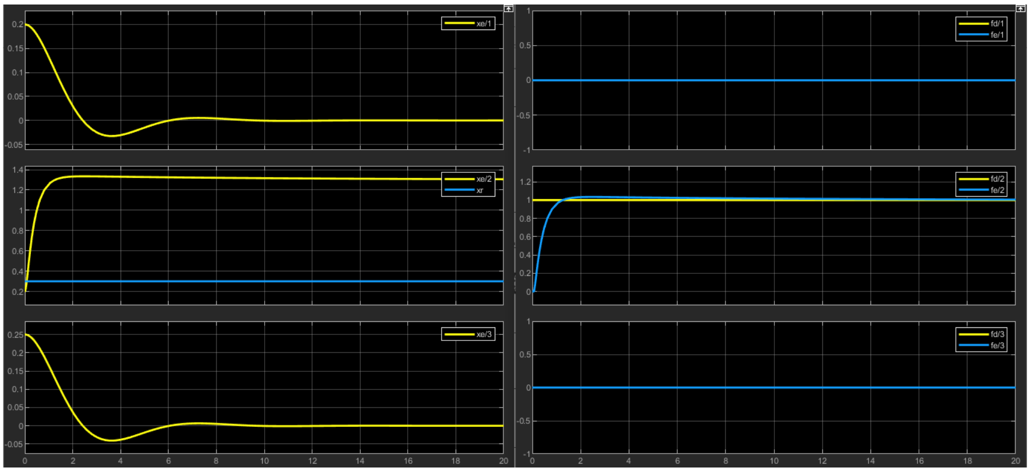


Figure 30: E.E. position, f_d and f_e comparison

Assuming a PI C_F, at steady state the error must be 0 and the following equation must be valid

$$Kx_e = Kx_r + f_d$$

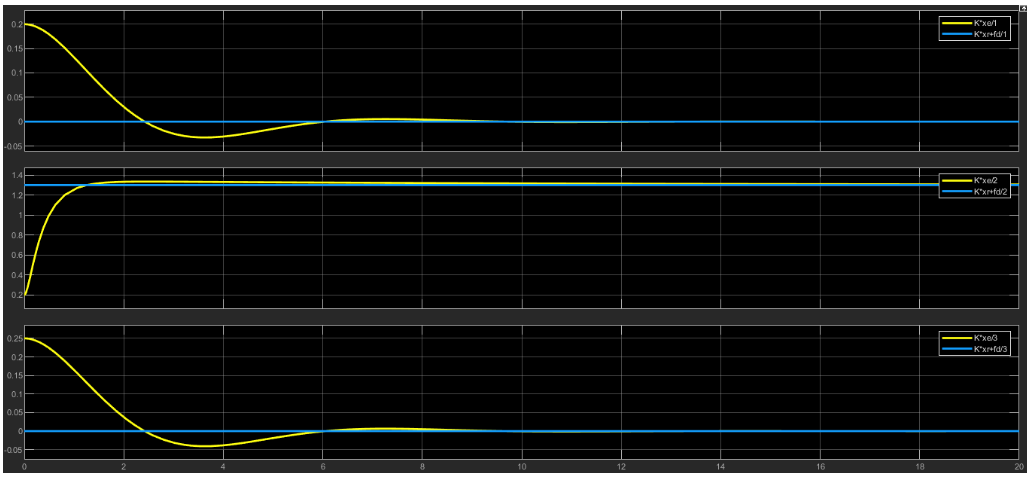


Figure 31: Plots which visualize the equivalence $Kx_e = Kx_r + f_d$ at steady state component by component

15 Homework 15

15.1 Parallel Force/Position Control

Description: Implement the parallel force/position control with the inner position loop

The control law is described by the following equations

$$\tau = B(q)y + C(q,\dot{q})\dot{q} + g(q) + J(q)^T h_e$$
$$y = J^{-1}(q)M_d^{-1}(-K_D\dot{x_e} + K_P(x_F - x_e + \boldsymbol{x_d}) - M_d\dot{J}(q,\dot{q})\dot{q})$$
$$x_F = C_F(f_d - f_e) \quad f_e = K(x_e - x_r) \quad C_F = K_F + \frac{K_I}{s}$$

The parallel Force/Position Control is a control scheme where both the reference force f_d and the desired reference position are provided.

- Along directions outside the $Image(K)$, i.e. unconstrained motion, x_d is reached by x_e
- Along directions belonging to $Image(K)$, i.e. constrained motion, x_d acts like an additional disturbance
- In our case we can impose a desired force f_d only along the y direction, and impose a desired position x_d along x and z

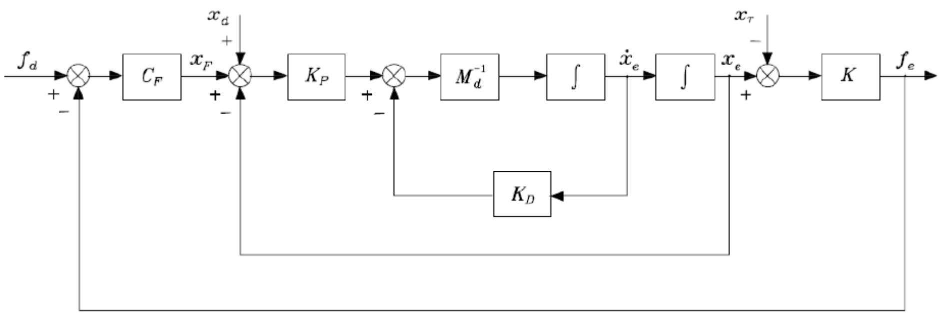


Figure 32: Scheme for the Parallel Force Position Control

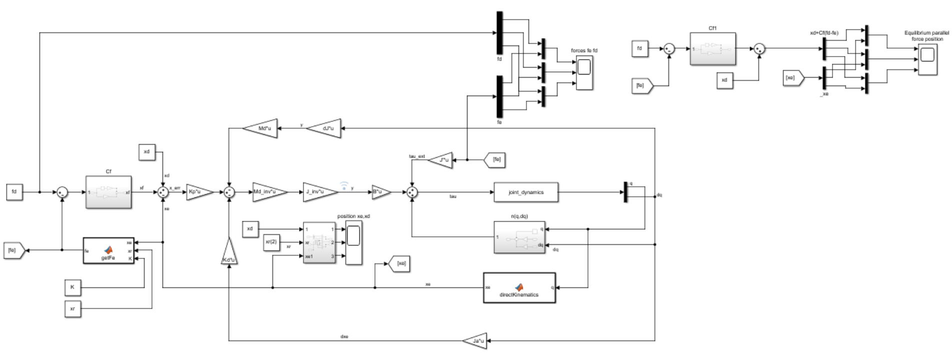


Figure 33: Implementation of the Parallel Force Position Control

The operational space E.E. starting position is $[0.2; 0.2; 0.2502]$. The following scopes represent the position of the end effector, and the interaction and desired force f_e and f_d. The environmental rest position was set to 0.3 along y. The environment was modelled ad an elastic system $f_e = K(x_e - x_r)$ only along the direction of contact of the E.E., for my robot it is y. The desired force was set to $f_d = 1$ along y. The desired position x_d was set to 0.3 along x and to 0.1 along z.

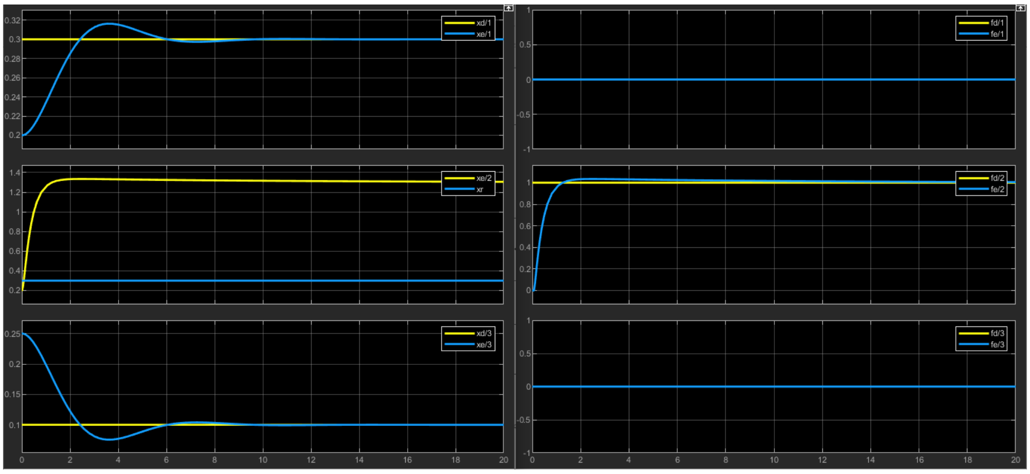


Figure 34: E.E. position, f_d and f_e comparison

Assuming a PI C_F, at steady state the error must be 0 and the following equation must be valid

$$x_e = x_d + C_F(K(x_r - x_e) + f_d)$$

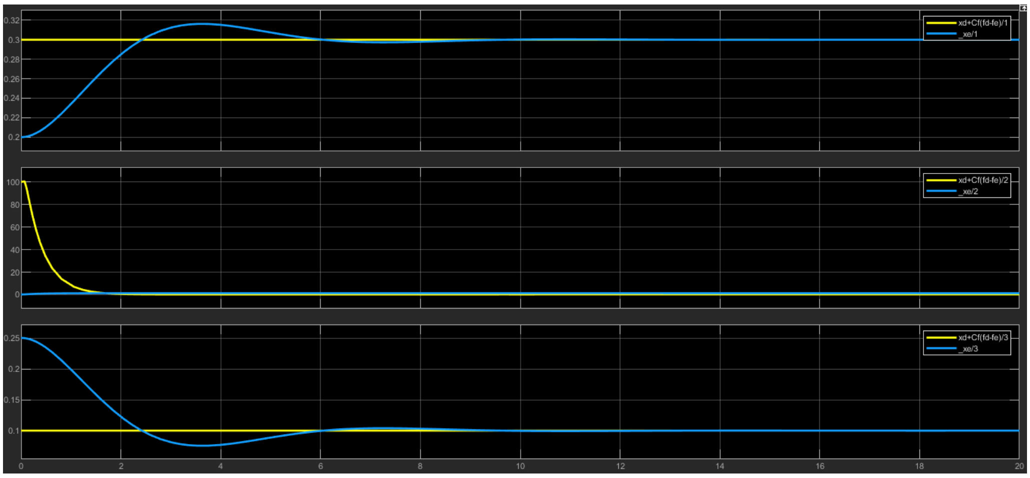


Figure 35: Plots which visualize the equivalence $x_e = x_d + C_F(K(x_r - x_e) + f_d)$ at steady state component by component