Advanced Control Systems Project

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

ACADEMIC YEAR: 2020-2021

Edoardo Fiorini

March 14, 2022

Summary

1 Assignment 1

In figure 1 is shown the robot which is a RRP(Rotative-Rotative-Prismatic) robot.

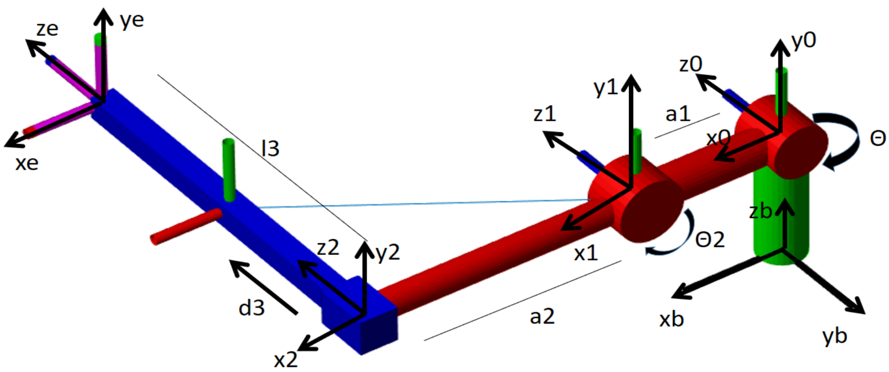

Figure 1: Robot visualization with Matlab toolbox

1.1 Denavit Hartenberg Table

The reference frame attached to each joint follows the Denavit Hartenberg convention, as it is shown by the table 1. The first line is only constant for the base reference frame.

	a$_i$	α_i	d$_i$	θ_i
b-0	0	$\pi/2$	d$_0$	0
0-1	a$_1$	0	0	θ_1^*
1-2	a$_2$	0	0	θ_2^*
2-e	0	0	l$_3 + d_3^*$	0

Table 1: Denavit Hartenberg Table of RRP robot

1.2 Direct Kinematics

In following the direct kinematics from the base frame to the end-effector. First of all we explicit all the single homogeneous transformation:

$$T_0^b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_1^0(\theta_1) = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1(\theta_2) = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_e^2(d_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_3 + d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then we can multiply in order to obtain the final transformation.

$$T_e^b = T_0^b T_1^0(\theta_1) T_2^1(\theta_2) T_e^2(d_3) = \begin{bmatrix} c_1c_2 - s_1s_2 & -c_1s_2 - c_2s_1 & 0 & a_1c_1 + a_2c_1c_2 - a_2s_1s_2 \\ 0 & 0 & -1 & -d_3 - l_3 \\ c_1s_2 + c_2s_1 & c_1c_2 - s_1s_2 & 0 & d_0 + a_1s_1 + a_2c_1s_2 + a_2c_2s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1.3 Inverse Kinematics

To solve the Inverse Kinematics we use the vector $[P_x P_y P_z]^\top$ from the homogeneous transformation T_4^0.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} a_1c_1 + a_2c_1c_2 - a_2s_1s_2 \\ -d_3 - l_3 \\ d_0 + a_1s_1 + a_2c_1s_2 + a_2c_2s_1 \end{bmatrix}$$

For the joint variable d_3 we can easily obtain:

$$d_3 = -P_y - l_3 \tag{1}$$

Instead for the joint variable θ_2 we apply square and summing, as it is shown in 2.

$$\begin{aligned} &P_x^2 + (P_z - d_0)^2 = (a_1c_1 + a_2c_{12})^2 + (a_1s_1 + a_2s_{12})^2 \\ &P_x^2 + (P_z - d_0)^2 = a_1^2 + a_2^2 + 2a_1a_2c_2 \\ &c_2 = \frac{P_x^2 + (P_z - a_0)^2 - a_1^2 - a_2^2}{2a_1a_2} \\ &s_2 = \pm\sqrt{1 - c_2^2} \\ &\theta_2 = Atan2(s_2, c_2) \end{aligned} \tag{2}$$

For the first joint variable θ_1 we apply square and summing again.

$$\begin{aligned} c_1 &= \frac{(a_1 + a_2c_2)P_x + a_2s_2(P_z - d_0)}{P_x^2 + (P_z - d_0)^2} \\ s_1 &= \frac{(a_1 + a_2c_2)(P_z - d_0) - a_2s_2P_x}{P_x^2 + (P_z - d_0)^2} \\ \theta_1 &= Atan2(s_1, c_1) \end{aligned} \tag{3}$$

1.4 Geometric Jacobian

The Geometric Jacobian is a blocks matrix 6 x n, where n is the degree of freedom in our case 3. It depends on $q = [\theta_1\theta_2d_3]^\top$. This is the Jacobian respect the base b of the robot, using the transformation T_{j0}^b from the Jacobian w.r.t frame 0.

$$J(q) = \begin{bmatrix} J_P(q) \\ J_O(q) \end{bmatrix} = T_{j1}^0 \begin{bmatrix} -a_1s_1 - a_2s_{12} & -a_2s_{12} & 0 \\ a_1c_1 + a_2c_{12} & a_2c_{12} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -a_1s_1 - a_2c_1s_2 - a_2c_2s_1 & -a_2c_1s_2 - a_2c_2s_1 & 0 \\ 0 & 0 & -1 \\ a_1c_1 + a_2c_1c_2 - a_2s_1s_2 & a_2c_1c_2 - a_2s_1s_2 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1.5 Analytical Jacobian

Instead of using the angular velocity ω_e, we want to use the rotational velocity $\dot{\phi}_e = \frac{\partial \phi_e}{\partial t}$. The relation between these two guys is:

$$\omega_e = T(\phi_e)\dot{\phi}_e \qquad \text{where} \qquad T(\phi_e) = \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\theta \\ 0 & c_\varphi & s_\varphi s_\theta \\ 1 & 0 & c_\theta \end{bmatrix}$$

So we have the following relation:

$$J(q) = T_A(\phi_e)J_A(q) \quad \text{where} \qquad T_A(\phi_e) = \begin{bmatrix} \mathbb{I}_3 & \varnothing_3 \\ \varnothing_3 & T(\phi_e) \end{bmatrix}$$

Inverting the $T_A(\phi_e)$ we can compute the analytical Jacobian, respect the frame 0 the result is:

$$J_A = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

2 Energy

In the second assignment we compute the energy of our robot: Kinetic and Potential.

2.1 Kinetic Energy

The kinetic energy has the following equation:

$$\begin{aligned}
&\mathcal{T}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^\top B(\boldsymbol{q})\dot{\boldsymbol{q}} \\
&\text{where} \quad B(\boldsymbol{q}) = \sum_{i=1}^{n} B_i(\boldsymbol{q}) = \sum_{i=1}^{n} m_{\ell_i}\big(J_P^{\ell_i\top} J_P^{\ell_i}\big) + \big(R_i^{0\top} J_O^{\ell_i}\big)^\top I_{\ell_i}^i \big(R_i^{0\top} J_O^{\ell_i}\big)
\end{aligned} \tag{4}$$

In order to obtain the matrix $B(\boldsymbol{q})$ first of all we have to compute the inertial tensor respect Σ_i. We have to apply the Steiner theorem to compute $I_{\ell_i}^i$ translated respect the centre of mass. For the cylinder: a is the major radius, b the minor one. Instead for the cube, b is the base and c the high. For the length we have used the same parameters of DH table.

$$\mathrm{I}_{\ell_1}^1 = I_{\ell_1}^{C_1} + link1_m\big(r^\top r\mathbb{I} - rr^\top\big) \quad \text{where} \quad r = p_{\ell_1}^1 = \begin{bmatrix} -\frac{a_1}{2} \\ 0 \\ 0 \end{bmatrix}$$

$$= \tfrac{1}{12}m_{\ell_1}\begin{bmatrix} link1_a^2 + link1_b^2 & & \\ & 3(link1_a^2 + link1_b^2) + 4a_1^2 & \\ & & 3(link1_a^2 + link1_b^2) + 4a_1^2 \end{bmatrix}$$

$$\mathrm{I}_{\ell_2}^2 = I_{\ell_2}^{C_2} + link2_m\big(r^\top r\mathbb{I} - rr^\top\big) \qquad \text{where} \qquad r = p_{\ell_2}^2 = \begin{bmatrix} -\frac{a_2}{2} \\ 0 \\ 0 \end{bmatrix}$$

$$= \tfrac{1}{12}m_{\ell_2}\begin{bmatrix} link2_a^2 + link2_b^2 & & \\ & 3(link2_a^2 + link2_b^2) + 4a_2^2 & \\ & & 3(link2_a^2 + link2_b^2) + 4a_2^2 \end{bmatrix}$$

$$\mathrm{I}_{\ell_3}^3 = I_{\ell_3}^{C_3} + link3_m\big(r^\top r\mathbb{I} - rr^\top\big) \qquad \text{where} \qquad r = p_{\ell_3}^3 = \begin{bmatrix} 0 \\ 0 \\ -\frac{\ell_3}{2} \end{bmatrix}$$

$$= \tfrac{1}{12}m_{\ell_3}\begin{bmatrix} link3_b^2 + 4\ell_3^2 & & \\ & link3_c^2 + 4\ell_3^2 & \\ & & link3_b^2 + link3_c^2 \end{bmatrix}$$

Now that we have all the inertial tensors, we compute the partial Jacobian. First of all, we have to compute the coordinate of the centre of mass $p_{\ell_i}^i$ respect the reference Σ_0, using the homogeneous matrix.

$$p_{\ell_1} = R_1^0 p_{\ell_1}^1 + P_1^0 = \begin{bmatrix} \frac{a_1}{2}c_1 \\ \frac{a_1}{2}s_1 \\ 0 \end{bmatrix}, \quad p_{\ell_2} = R_2^0 p_{\ell_2}^2 + P_2^0 = \begin{bmatrix} \frac{a_2}{2}c_{12} + a_1 c_1 \\ \frac{a_2}{2}s_{12} + a_1 s_1 \\ 0 \end{bmatrix}, \quad p_{\ell_3} = R_3^0 p_{\ell_3}^3 + P_3^0 = \begin{bmatrix} \frac{a_2}{2}c_{12} + a_1 c_1 \\ \frac{a_2}{2}s_{12} + a_1 s_1 \\ \frac{\ell_3}{2} + d_3 \end{bmatrix}$$

At this point, we are ready to compute the partial Jacobian for each link. We can consider the position $J_P^{\ell_i}$ and orientation $J_O^{\ell_i}$ separately.

$$J_{P_1}^{\ell_1} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_1} - P_0^0) = \begin{bmatrix} -\frac{a_1}{2} s_1 \\ \frac{a_1}{2} c_1 \\ 0 \end{bmatrix} \qquad J_{O_1}^{\ell_1} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_P^{\ell_1} = \begin{bmatrix} -\frac{a_1}{2} s_1 & 0 & 0 \\ \frac{a_1}{2} c_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad J_O^{\ell_1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$J_{P_1}^{\ell_2} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_2} - P_0^0) = \begin{bmatrix} -a_1 s_1 - \frac{a_2}{2} s_{12} \\ a_1 c_1 + \frac{a_2}{2} c_{12} \\ 0 \end{bmatrix} \qquad J_{O_1}^{\ell_2} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_{P_2}^{\ell_2} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_2} - P_1^0) = \begin{bmatrix} -\frac{a_2}{2} s_{12} \\ \frac{a_2}{2} c_{12} \\ 0 \end{bmatrix} \qquad J_{O_2}^{\ell_2} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_P^{\ell_2} = \begin{bmatrix} -a_1 s_1 - \frac{a_2}{2} s_{12} & -\frac{a_2}{2} s_{12} & 0 \\ a_1 c_1 + \frac{a_2}{2} c_{12} & \frac{a_2}{2} c_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad J_O^{\ell_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$J_{P_1}^{\ell_3} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_3} - P_0^0) = \begin{bmatrix} -a_2 s_{12} - a_1 s_1 \\ a_2 c_{12} + a_1 c_1 \\ 0 \end{bmatrix} \qquad J_{O_1}^{\ell_3} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_{P_2}^{\ell_3} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_3} - P_1^0) = \begin{bmatrix} -a_2 s_{12} \\ a_2 c_{12} \\ 0 \end{bmatrix} \qquad J_{O_2}^{\ell_3} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_{P_3}^{\ell_3} = R_2^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad J_{O_3}^{\ell_3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$J_P^{\ell_3} = \begin{bmatrix} -a_2 s_{12} - a_1 s_1 & -a_2 s_{12} & 0 \\ a_2 c_{12} + a_1 c_1 & a_2 c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad J_O^{\ell_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Having all the contribute in equation 4 we can compute $B(\boldsymbol{q})$ for each link. Summing all the components we end up with:

$$
\begin{aligned}
B(\boldsymbol{q}) &= B_1(\boldsymbol{q}) + B_2(\boldsymbol{q}) + B_3(\boldsymbol{q}) \\
&= \begin{bmatrix} E_1 + (link2_m + 2link3_m)a_1a_2c_2 & E_2 + (\frac{1}{2}link2_m + link3_m)a_1a_2c_2 & 0 \\ * & E_3 & 0 \\ * & * & link3_m \end{bmatrix} \\
E_1 &= link1_m(\tfrac{7}{12}a_1^2 + \tfrac{1}{4}link1_a^2 + link1_b^2) + link2_m(a_1^2 + \tfrac{7}{12}a_2^2 + \tfrac{1}{4}link2_a^2 + link2_b^2) + \\
&\quad + link3_m(a_1^2 + a_2^2 + \tfrac{1}{6}link3_b^2 + link3_c^2) \\
E_2 &= link2_m(\tfrac{7}{12}a_2^2 + \tfrac{1}{4}link2_a^2 + link2_b^2) + link3_m(a_2^2 + \tfrac{1}{6}link3_b^2 + link3_c^2) \\
E_3 &= link2_m(\tfrac{7}{12}a_2^2 + \tfrac{1}{4}link2_a^2 + link2_b^2) + link3_m(a_2^2 + \tfrac{1}{6}link3_b^2 + link3_c^2)
\end{aligned}
$$

2.2 Potential Energy

The Potential energy has the following equation:

$$
\mathcal{U}(\boldsymbol{q}) = \sum_{i=1}^{n} \mathcal{U}_i(\boldsymbol{q}) = -\sum_{i=1}^{n} m_{\ell_i} g_0^\top p_{\ell_i} \tag{5}
$$

We have already compute all the the coordinate of the centre of mass $p_{\ell_i}^i$ respect the reference Σ_0 and $g_0 = [0, -g, 0]^\top$. The result of the potential energy in our serial link manipulator is:

$$
\mathcal{U}(\boldsymbol{q}) = (\tfrac{1}{2}link1_m + link2_m + link3_m)ga_1s_1 + (\tfrac{1}{2}link2_m + link3_m)ga_2s_{12}
$$

3 Lagrangian Dynamic Model

In this assignment we compute the equations of motion(dynamic model). The equation is the following one:

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) = \boldsymbol{\tau} - F_v\dot{\boldsymbol{q}} - F_s\,\mathrm{sgn}(\dot{\boldsymbol{q}}) - J^\top(\boldsymbol{q})h_e \tag{6}$$

For the moment we discard the friction F_v e F_s and the external forces h_e. We already have the matrix $B(\boldsymbol{q})$ from the Kinetic energy computation. We have all the elements from the previous computation. For the $C(\boldsymbol{q},\dot{\boldsymbol{q}})$ the equation is the following one:

$$C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = -\sum_{j=1}^{n} c_{ij} \qquad \text{where} \qquad c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k$$

To compute c_{ij} we have to sum different partial derivative:

$$c_{ijk} = c_{ikj} = \tfrac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)$$

Each element of the previous equation is a partial derivative of the $B(\boldsymbol{q})$ matrix, where the indices on the numerator are the column and the row of the $B(\boldsymbol{q})$ that we derive respect the joint variable of the index of the denominator. The result is the following one:

$$C(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} -\frac{1}{2}(link2_m + 2link3_m)a_1a_2s_2\dot{\theta}_2 & -\frac{1}{2}(link2_m + 2link3_m)a_1a_2s_2(\dot{\theta}_1 + \dot{\theta}_2) & 0 \\ \frac{1}{2}(link2_m + 2link3_m)a_1a_2s_2\dot{\theta}_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Now we compute the g($\boldsymbol{q}$)$, as: g_i(\boldsymbol{q}) = -\sum_{j=1}^{n} m_{\ell_j} g_0^\top J_{P_i}^{\ell_j}$
After all the computation we end up with:

$$g(\boldsymbol{q}) = g\begin{bmatrix} (\frac{1}{2}link1_m + link2_m + link3_m)a_1c_1 + (\frac{1}{2}link2_m + link3_m)a_2c_{12} \\ (\frac{1}{2}link2_m + link3_m)a_2c_{12} \\ 0 \end{bmatrix}$$

4 Newton-Euler Dynamic Model

In this assignment we compute the dynamic model using the Newton-Euler formulation. The algorithm is based on two steps: the forward and the backward. In this section we report only the pseudo code because if we isolate the matrices $B(\boldsymbol{q})$, $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $g(\boldsymbol{q})$ the results are the same of the assignment 3.

The forward N-E takes as input joint positions, velocities accelerations $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ and some initial conditions, from which computes $\omega_i, \dot{\omega}_i, \ddot{p}_i, \ddot{p}_{ci}, \dot{\omega}_{mi}$ from the base link to the end-effector. The pseudo code is the following one:

/* Initial Conditions */
ω_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 $R_i^{i-1} = R_i^{i-1}(d_i)$
  $\omega_i^i = (R_i^{i-1})^T \omega_{i-1}^{i-1} + (R_i^{i-1})^T \dot{\vartheta}_i z_0$
  $\dot{\omega}_i^i = (R_i^{i-1})^T \dot{\omega}_{i-1}^{i-1} + (R_i^{i-1})^T (\ddot{\vartheta}_i z_0 + \dot{\vartheta}_i \omega_{i-1}^{i-1} \times z_0)$
  $\ddot{p}_i = (R_i^{i-1})^T \ddot{p}_{i-1}^{i-1} + \dot{\omega}_i^i \times r_{i-1,i}^i + \omega_i^i \times (\omega_i^i \times r_{i-1,i}^i) + (R_i^{i-1})^T \ddot{d}_i z_0 + 2\dot{d}_i \omega_i^i \times ((R_i^{i-1})^T z_0)$
  $\ddot{p}_{C_i}^i = \ddot{p}_i^i + \dot{\omega}_i^i \times r_{i,C_i}^i + \omega_i^i \times (\omega_i^i \times r_{i,C_i}^i)$
  $\dot{\omega}_{m_i}^{i-1} = \dot{\omega}_{i-1}^{i-1} + k_{ri} \ddot{q}_i z_{m_i}^{i-1} + k_{ri} \dot{q}_i \omega_{i-1}^{i-1} \times z_{m_i}^{i-1}$
end for

Figure 2: Forward Newton-Euler formulation

The backward N-E takes as input h_e and compute f_i μ_i from the end-effector to the base link. Then also the joint torques. The pseudo code is the following one:

/* 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 ω_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 $R_{i+1}^i = R_{i+1}^i(d_{i+1})$
  $f_i^i = R_{i+1}^i f_{i+1}^{i+1} + m_i \ddot{p}_{C_i}^i$
  $\mu_i^i = -f_i^i \times (r_{i-1,i}^i + r_{i,C_i}^i) + R_{i+1}^i \mu_{i+1}^{i+1} + R_{i+1}^i f_{i+1}^{i+1} \times r_{i,C_i}^i + \bar{I}_i^i \dot{\omega}_i^i + \omega_i^i \times (\bar{I}_i^i \omega_i^i) +$
   $+ k_{r,i+1} \ddot{q}_{i+1} I_{m_{i+1}} z_{m_{i+1}}^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \omega_i^i \times z_{m_{i+1}}^i$
  $\tau_i = \begin{cases} (f_i^i)^T (R_i^{i-1})^T z_0 + k_{ri} I_{m_i} (\dot{\omega}_{m_i}^{i-1})^T z_{m_i}^{i-1} + F_{vi} \dot{d}_i + F_{si} \text{sign}(\dot{d}_i) \\ (\mu_i^i)^T (R_i^{i-1})^T z_0 + k_{ri} I_{m_i} (\dot{\omega}_{m_i}^{i-1})^T z_{m_i}^{i-1} + F_{vi} \dot{\vartheta}_i + F_{si} \text{sign}(\dot{\vartheta}_i) \end{cases}$
end for

Figure 3: Backward Newton-Euler formulation

The equivalence with Lagrangian can be checked with the following relations:

$$g(q) = NE(q, 0, 0, g_0)$$

$$C(q, \dot{q})\dot{q} = NE(q, \dot{q}, 0, 0)$$

$$B(q) = \begin{bmatrix} B_1(q) & \dots & B_n(q) \end{bmatrix} \text{ with } B_i(q) = NE(q, 0, e_i, 0)$$

5 Lagrangian Dynamic Model in Operational Space

In this assignment we compute the dynamic model in the operational space. Exploiting some relationship from the Cartesian coordinate to the joint space we end up with:

$$B_A(x)\ddot{x} + C_A(x,\dot{x})\dot{x} + g_A(x) = u - u_e \tag{7}$$

where

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

It is important to underlined that in our case we have used the simplified equation for non-redundant manipulator in a non singular configuration.

In following we report the $\dot{J}_A$ respect frame 0, computed as derivative respect time:

$$\dot{J}_A = \begin{bmatrix} -a_2\dot{\theta}_1 c_1 c_2 - a_2\dot{\theta}_2 c_1 c_2 - a_1\dot{\theta}_1 c_1 & -a_2 c_1 c_2(\dot{\theta}_1 + \dot{\theta}_2) & 0 \\ -a_1\dot{\theta}_1 s_1 - a_2\dot{\theta}_1 s_{1s2} - a_2\dot{\theta}_2 s_1 s_2 & -a_2 s_1 s_2(\dot{\theta}_1 + \dot{\theta}_2) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

6 Motion Control

In this section we applied on the dynamic model of our robot some motion control scheme both in joint and operational space. Everything has been implemented in Matlab/Simulink, but here we report only the control scheme, reference configuration/trajectory, parameters of the controller and the results achieved.

6.1 Joint Space PD control with Gravity Compensation

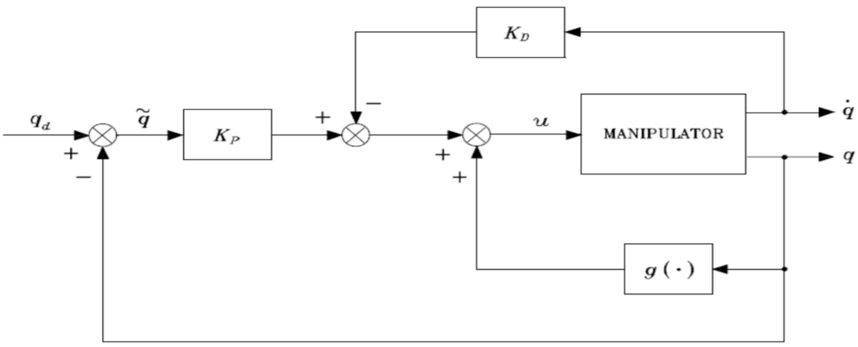

Figure 4: Joint Space PD Gravity Compensation control scheme

$$\boldsymbol{q}_d = \begin{bmatrix} \pi/4 \\ -\pi/6 \\ -0.3 \end{bmatrix} \qquad \begin{aligned} KP &= \mathrm{diag}(250, 240, 200) \\ KD &= \mathrm{diag}(25, 10, 21) \end{aligned}$$

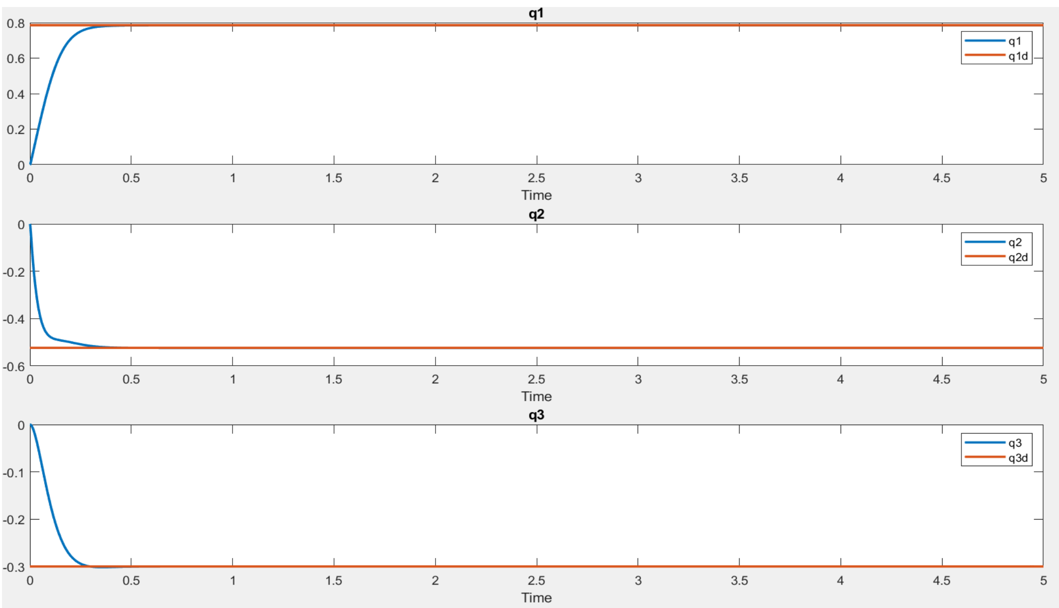

Figure 5: Joint Space PD Gravity Compensation control result

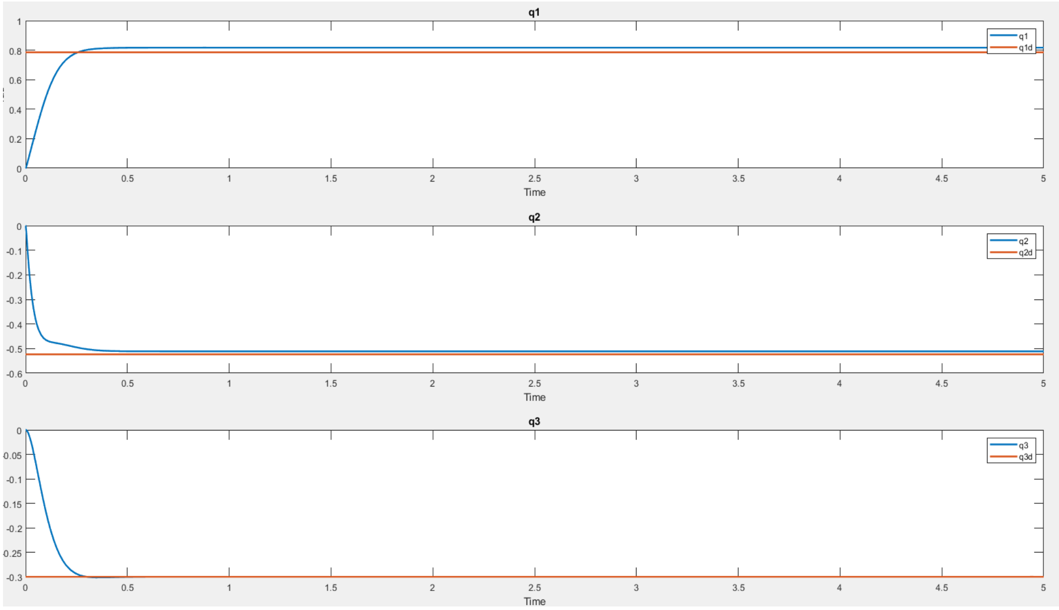

Figure 6: Joint Space PD Gravity Compensation control result when g(q) is not taken into account

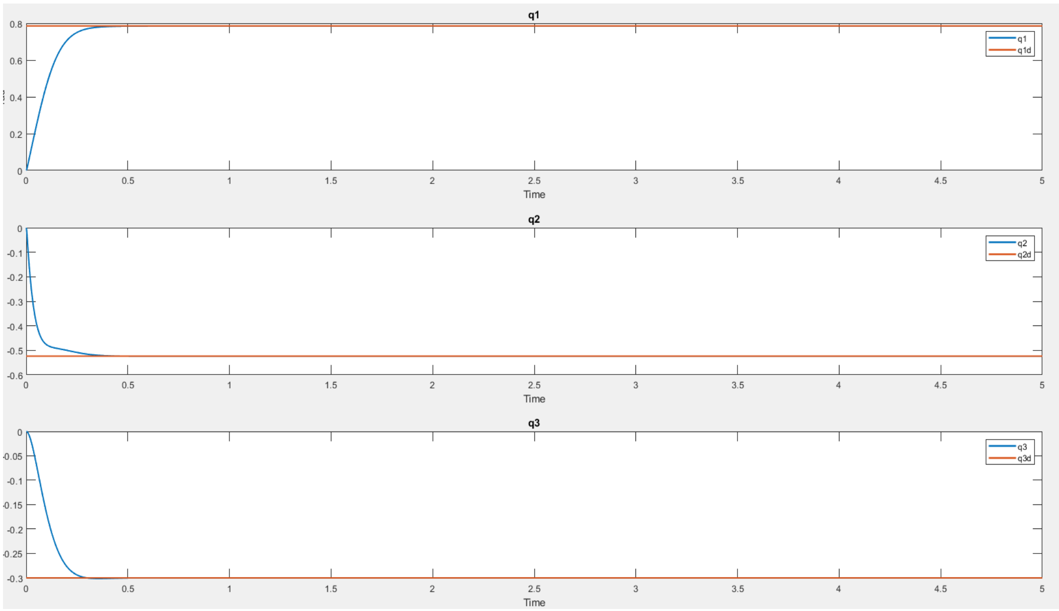

Figure 7: Joint Space PD Gravity Compensation control result when the gravity term is constant and equal to g(q_d)

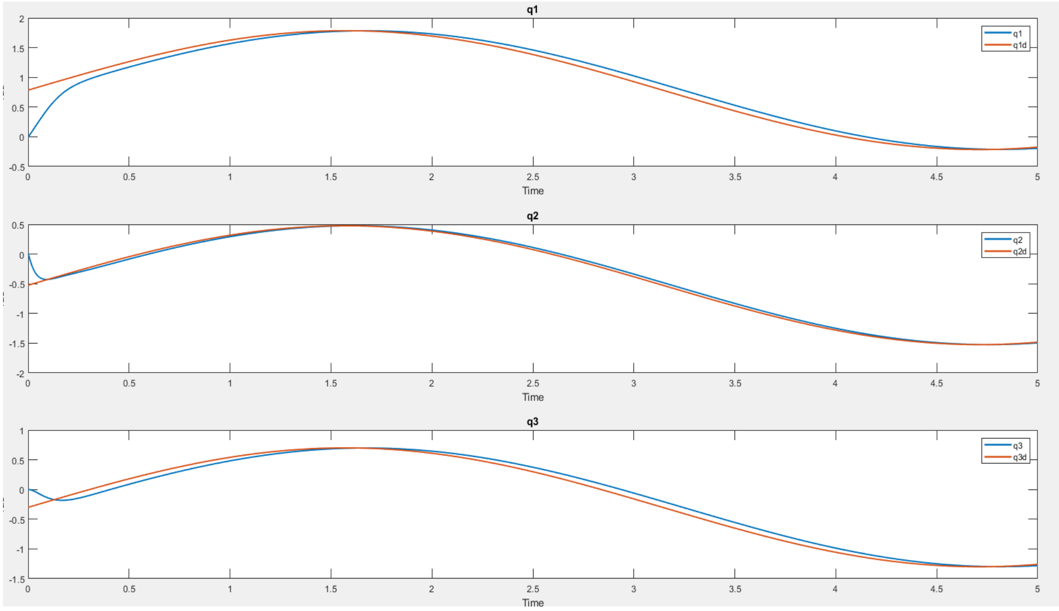

Figure 8: Joint Space PD Gravity Compensation control result when q_d is not constant

6.2 Joint Space Inverse Dynamic

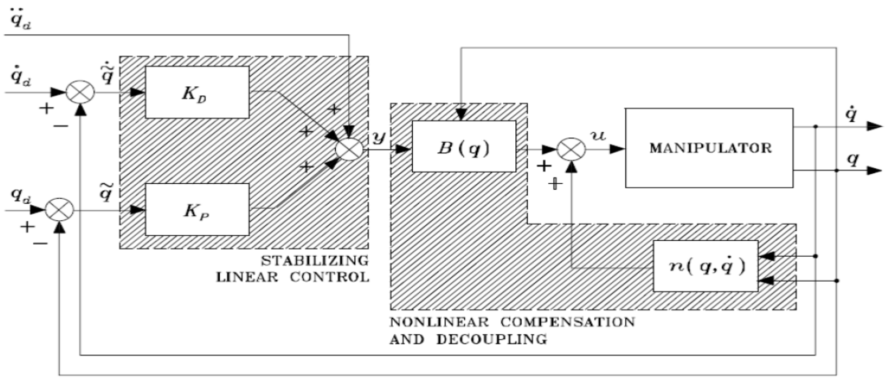

Figure 9: Joint Space Inverse Dynamic scheme

$$\frac{t_d}{\boldsymbol{q}_d} = \begin{array}{c|c|c|c} 0 & 1.5 & 3 & 5 \\ \hline \pi/4 & \pi/2 & \pi/6 & \pi/4 \\ -\pi/6 & \pi/4 & -\pi/3 & -\pi/6 \\ 0.1 & -0.3 & 0.1 & 0.2 \end{array} \qquad \begin{aligned} KP &= \mathrm{diag}(240, 240, 200) \\ KD &= \mathrm{diag}(20, 10, 21) \end{aligned}$$

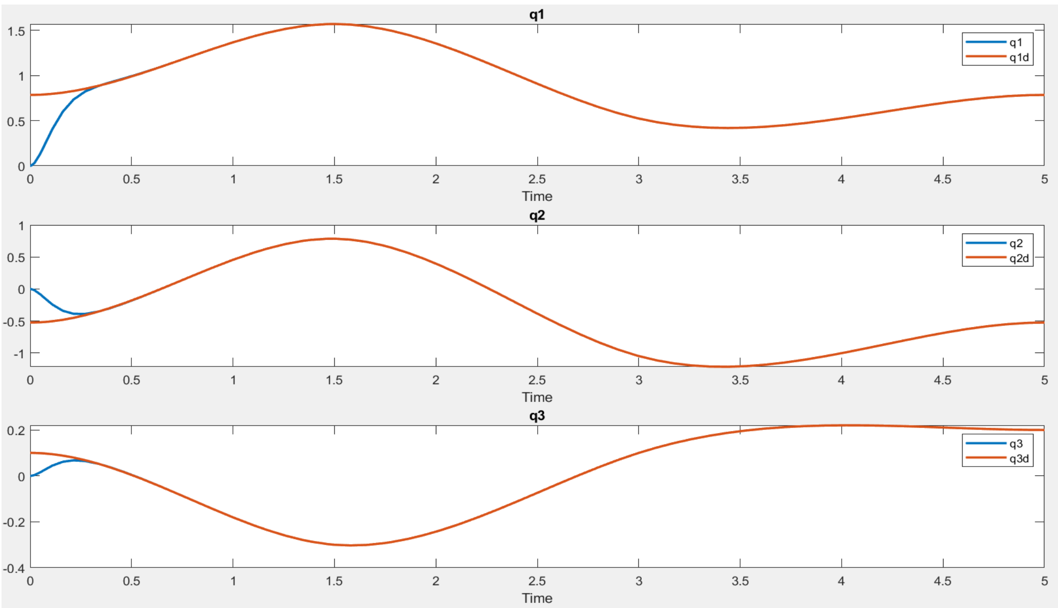

Figure 10: Joint Space Inverse Dynamic result

$$\boldsymbol{q}_d = \begin{bmatrix} \pi/4 \\ \pi/4 \\ \pi/4 \end{bmatrix} \qquad \begin{aligned} KP &= \mathrm{diag}(240, 240, 240) \\ KD &= \mathrm{diag}(24, 24, 24) \end{aligned}$$

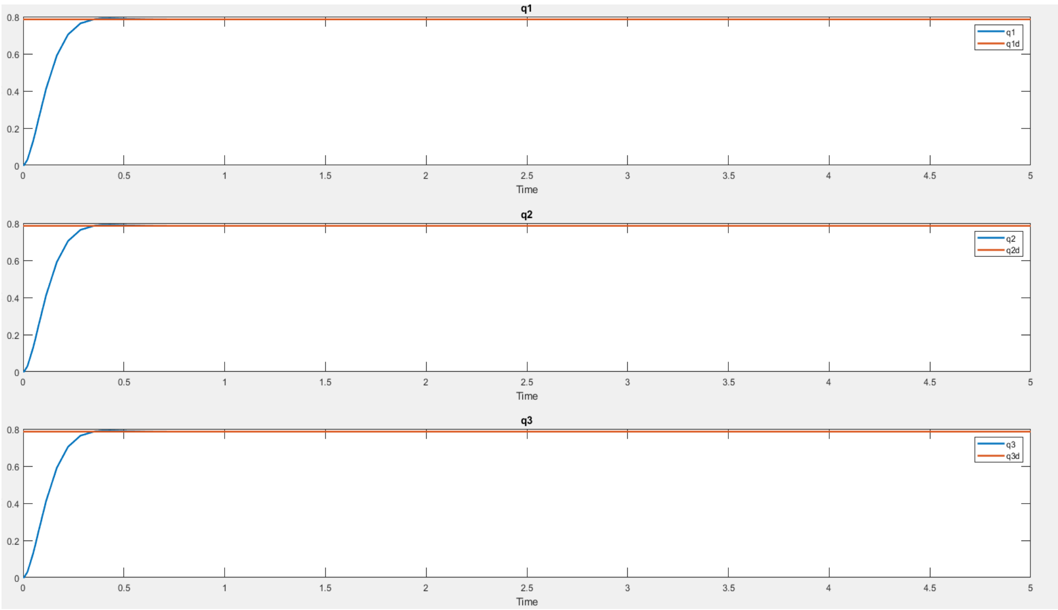

Figure 11: Joint Space Inverse Dynamic result equivalent to a set of stabilized double integrators

$$
\frac{t_d}{\boldsymbol{q}_d} = \begin{array}{c|c|c|c} 0 & 1.5 & 3 & 5 \\ \hline \pi/4 & \pi/2 & \pi/6 & \pi/4 \\ -\pi/6 & \pi/4 & -\pi/3 & -\pi/6 \\ 0.1 & -0.3 & 0.1 & 0.2 \end{array}
\qquad
\begin{aligned} KP &= \mathrm{diag}(240, 240, 200) \\ KD &= \mathrm{diag}(20, 10, 21) \\ \hat{a_1} &= 0.3 \\ \hat{l_3} &= 0.3 \end{aligned}
$$

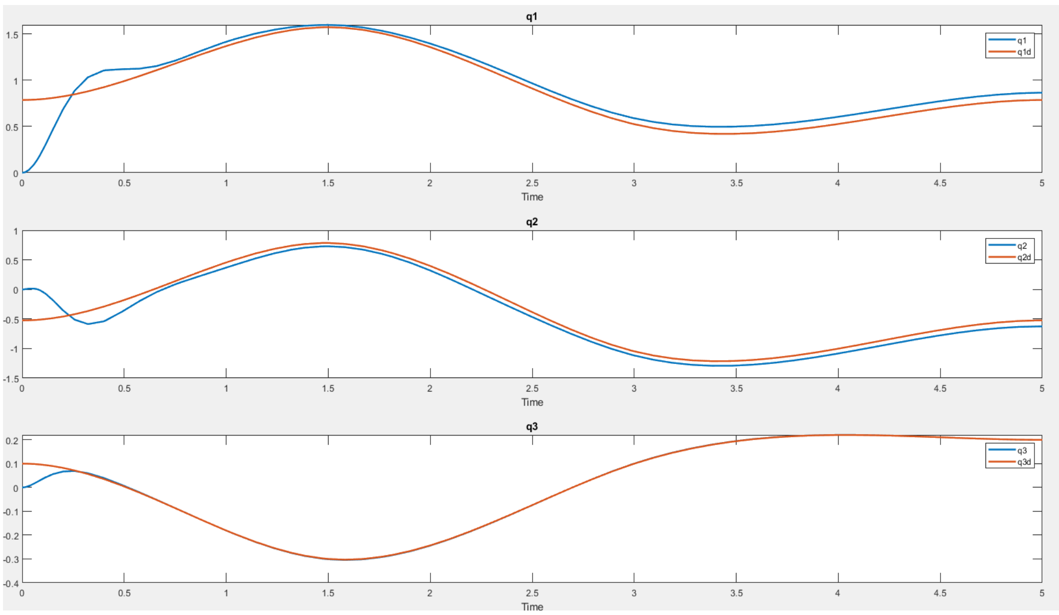

Figure 12: Joint Space Inverse Dynamic result with $\hat{B}$, $\hat{C}$, $\hat{g}$ within the controller

$$\boldsymbol{q}_d = \begin{bmatrix} \pi/4 \\ \pi/4 \\ \pi/4 \end{bmatrix} \qquad\qquad \begin{aligned} KP &= \text{diag}(500, 500, 500) \\ KD &= \text{diag}(40, 40, 40) \end{aligned}$$

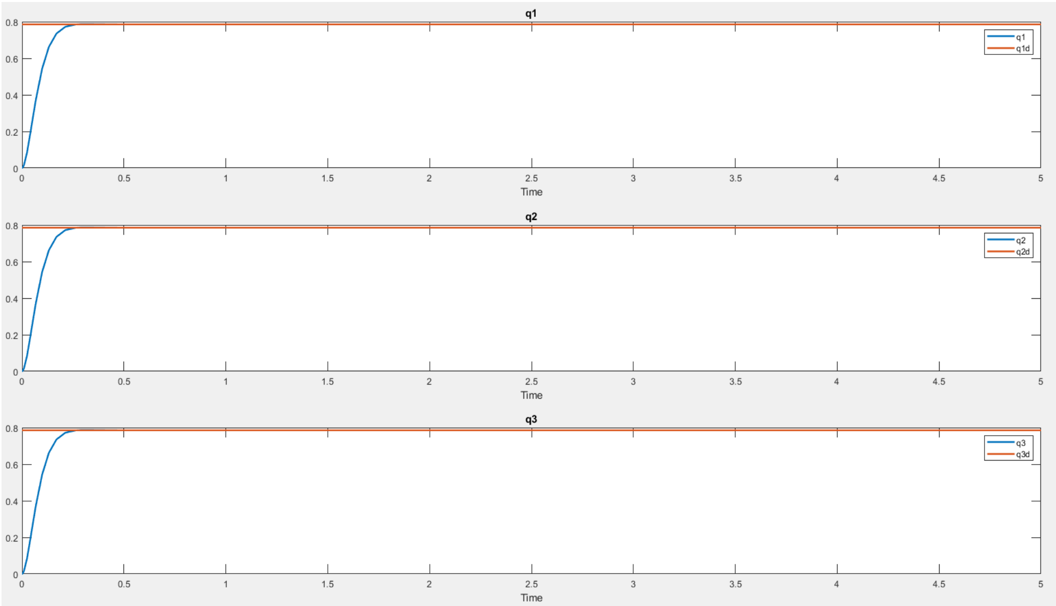

Figure 13: Joint Space Inverse Dynamic result with settling time very small

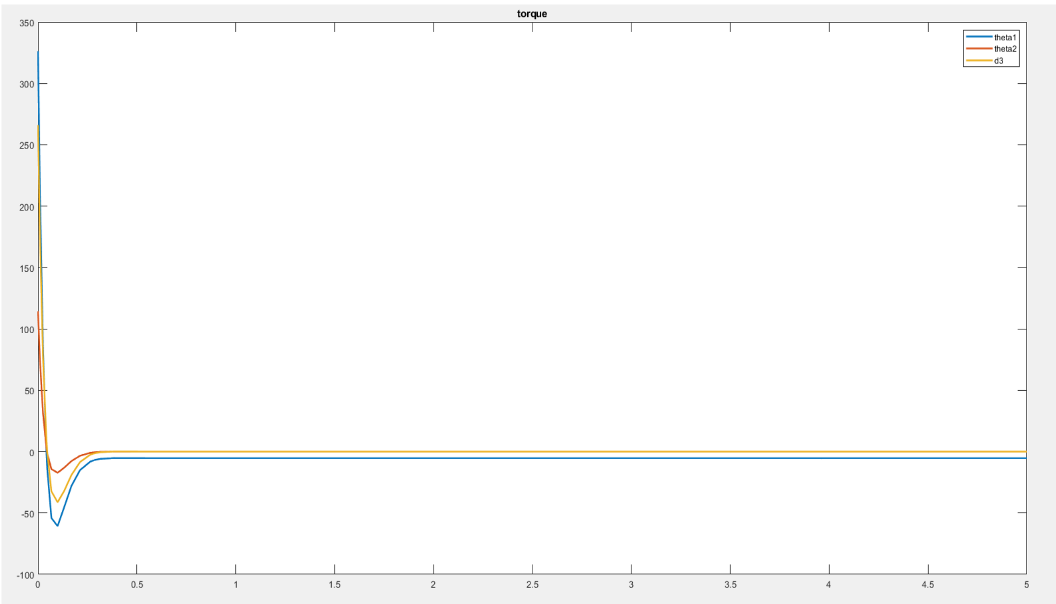

Figure 14: Joint Space Inverse Dynamic toruqe result when the settling time is very small

6.3 Adaptive Control

To simplify the computation of the matrix $Y(q, \dot{\boldsymbol{q}}, \dot{q}_r, \ddot{q}_r)$ we have implemented the Adaptive Control law for a 1-DoF link under gravity.

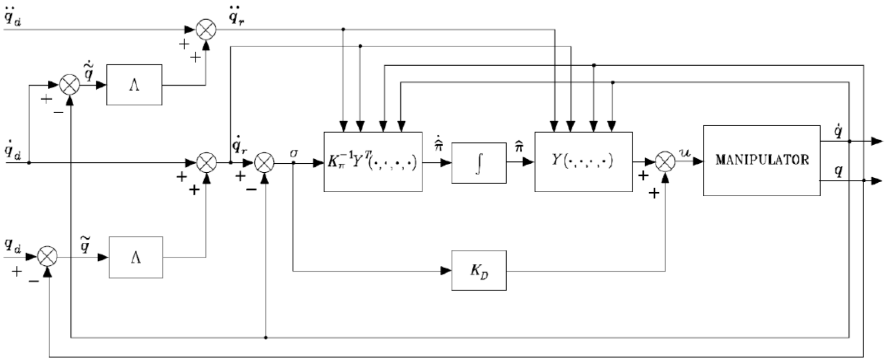

Figure 15: Adaptive Control scheme

$$
\begin{aligned}
\boldsymbol{q}_d &= A\sin(\omega t) & KD &= 100 \\
\ddot{\boldsymbol{q}}_d &= \textit{periodic square wave} \pm A & lambda &= 80 \\
A &= 1 & K_\theta &= \mathrm{diag}(1,1,1)
\end{aligned}
$$

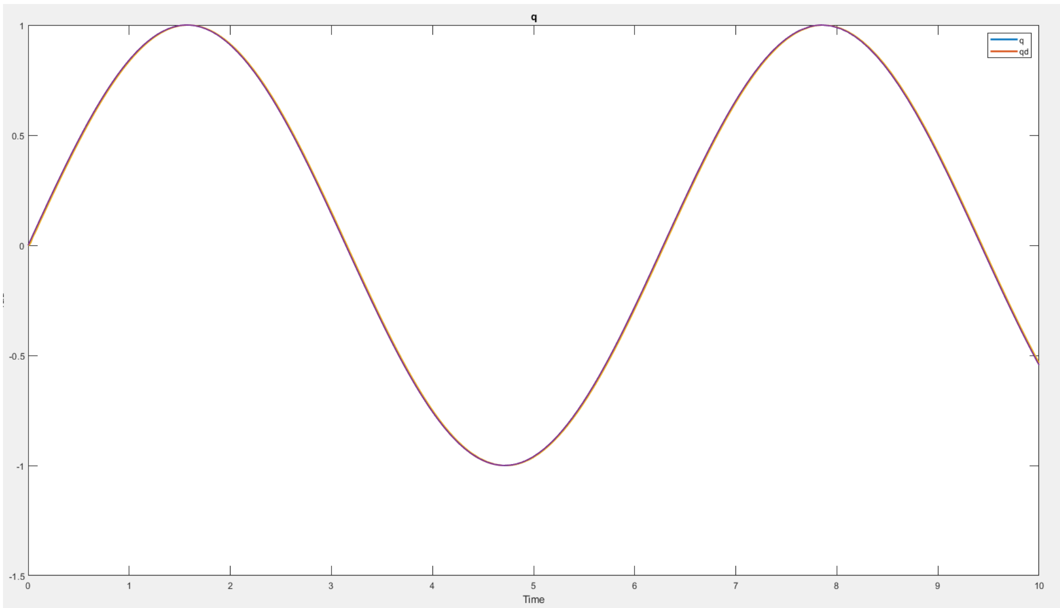

Figure 16: Adaptive Control result

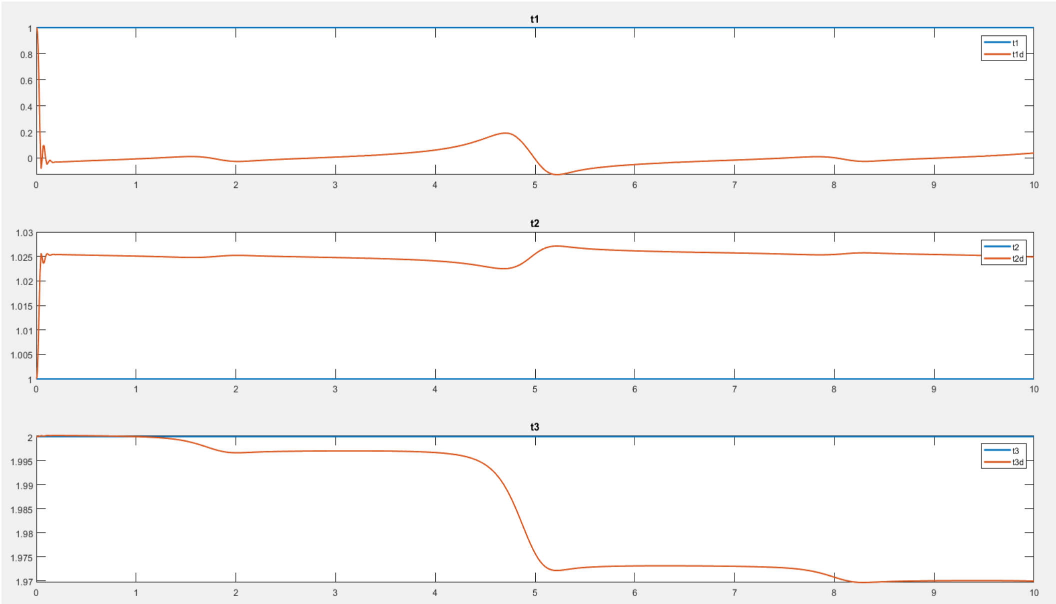

Figure 17: Adaptive Control $\hat{\theta}$ estimation with initial conditions

6.4 Operational Space PD Control with Gravity Compensation

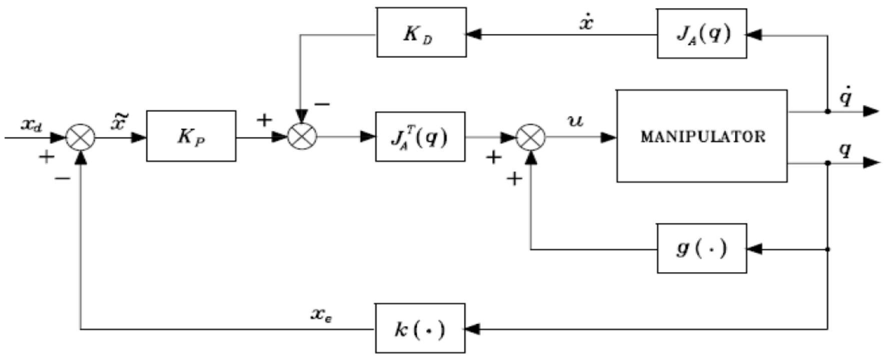

Figure 18: Operational Space PD Gravity Compensation scheme

$$\boldsymbol{q}_d = \begin{bmatrix} \pi/3 \\ -\pi/2 \\ -0.1 \end{bmatrix} \rightarrow \chi_d = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.3 \\ 0 \\ 0 \\ -0.6 \end{bmatrix} \qquad \begin{aligned} KP &= \text{diag}(1700, 3000, 1600, 100, 1, 4) \\ KD &= \text{diag}(150, 180, 160, 19, 0, 8) \end{aligned}$$

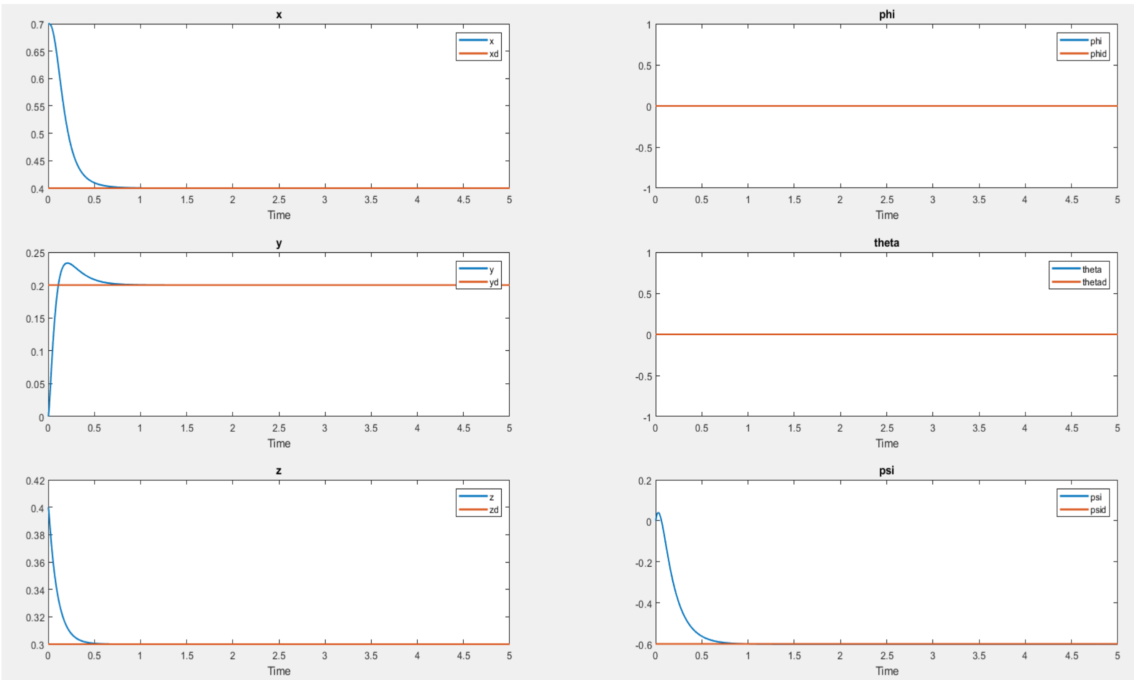

Figure 19: Operational Space PD Gravity Compensation result

6.5 Operational Space Inverse Dynamic

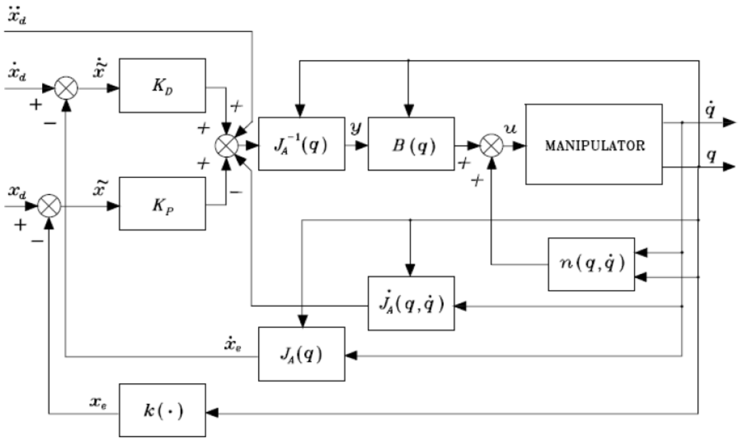

Figure 20: Operational Space Inverse Dynamic scheme

$$\frac{t_d}{\boldsymbol{q}_d} = \begin{array}{c|c|c|c} 0 & 1.5 & 3 & 5 \\ \hline \pi/4 & \pi/2 & \pi/6 & \pi/4 \\ -\pi/6 & \pi/4 & -\pi/3 & -\pi/6 \\ 0.1 & -0.3 & 0.1 & 0.2 \end{array} \qquad \begin{array}{l} KP = \text{diag}(900, 3000, 1600, 100, 1, 300) \\ KD = \text{diag}(49, 100, 160, 19, 0, 60) \end{array}$$

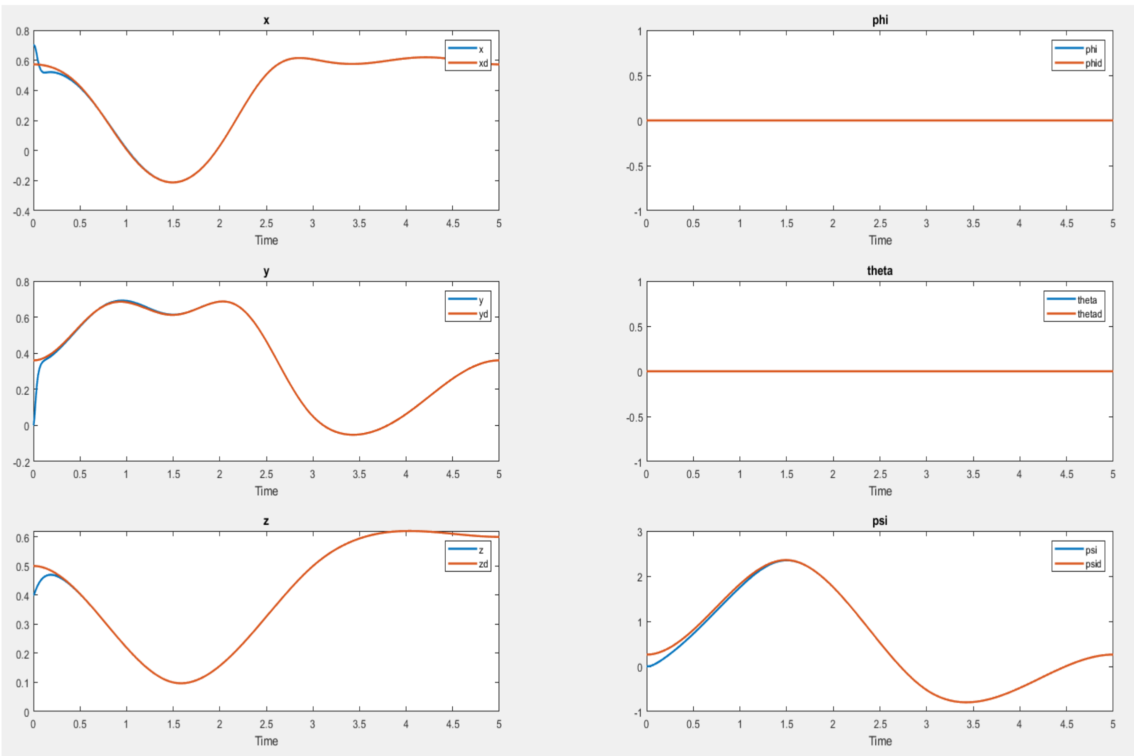

Figure 21: Operational Space Inverse Dynamic result

7 Force Control

In this section we applied on the dynamic model of our robot some force control scheme. Everything has been implemented in Matlab/Simulink, but here we report only the control scheme, reference trajectory(since the forces are generated on the end-effector, all trajectories are in Operational Space), force applied on the end-effector, parameters of the controller and the results achieved. We have simulated a plane at z = 0.5 respect to the frame 0 in order to have an interaction between the environment and the end-effector pose along the z-axis respect to the frame 0, so there is a generalized force h_e. To simplify the simulation the interaction is only along one axis and we assume that it is perfectly orthogonal(in order to have only the reaction force and avoid the moment). The environment is managed as a matlab-function and it has its own stiffness K.

7.1 Compliance Control

$$\chi_d = \begin{bmatrix} 0.4 & 0.2 & 0.7 & 0 & 0 & -0.6 \end{bmatrix}^\top$$

$$\begin{aligned} KP &= \text{diag}(1700, 3000, 1600, 100, 1, 4) \\ KD &= \text{diag}(150, 180, 160, 19, 0, 8) \\ K &= \text{diag}(10, 10, 10, 10, 10, 10) \\ plane &= 0.5 \end{aligned}$$

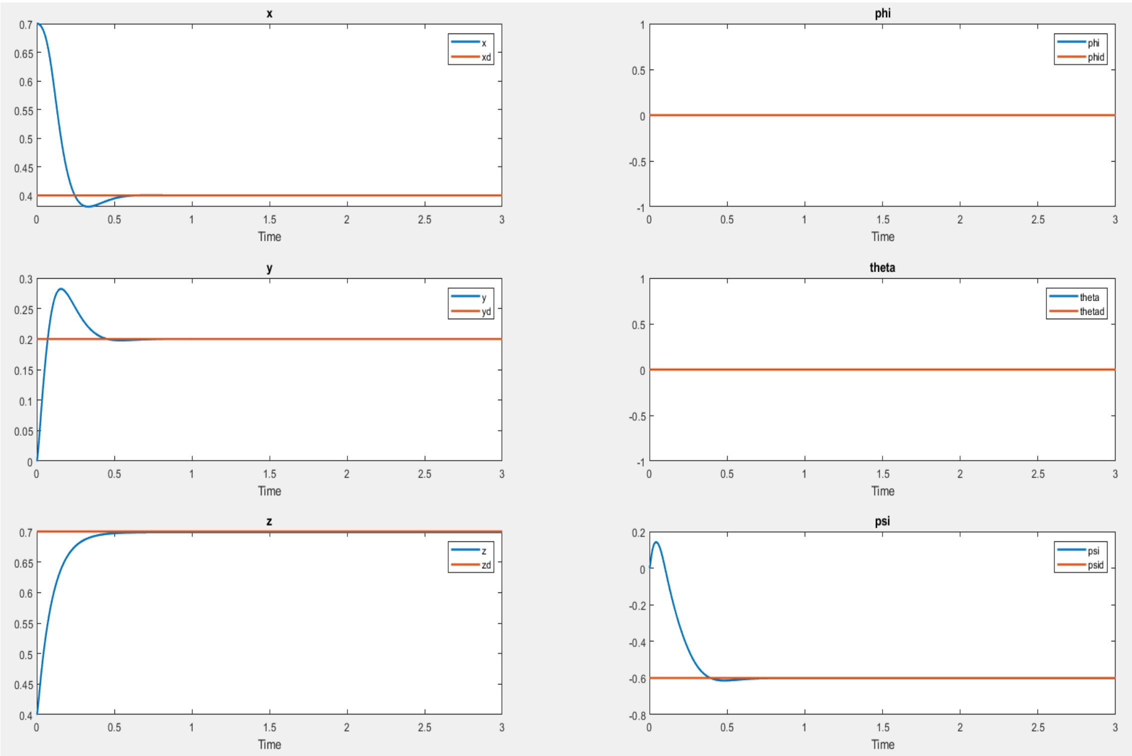

Figure 22: Compliance Control pose with KP >> K

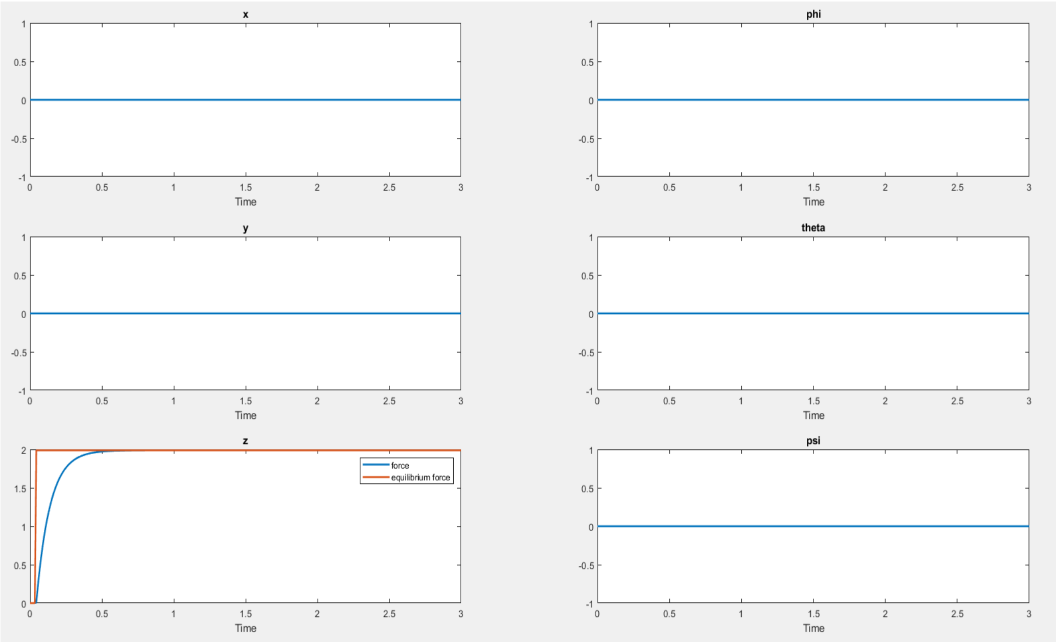

Figure 23: Compliance Control force

$$\chi_d = \begin{bmatrix} 0.4 & 0.2 & 0.7 & 0 & 0 & -0.6 \end{bmatrix}^\top$$

$$\begin{aligned} KP &= \text{diag}(1700, 3000, 1600, 100, 1, 4) \\ KD &= \text{diag}(150, 180, 160, 19, 0, 8) \\ K &= \text{diag}(10, 10, 100, 10, 10, 10) \\ plane &= 0.5 \end{aligned}$$

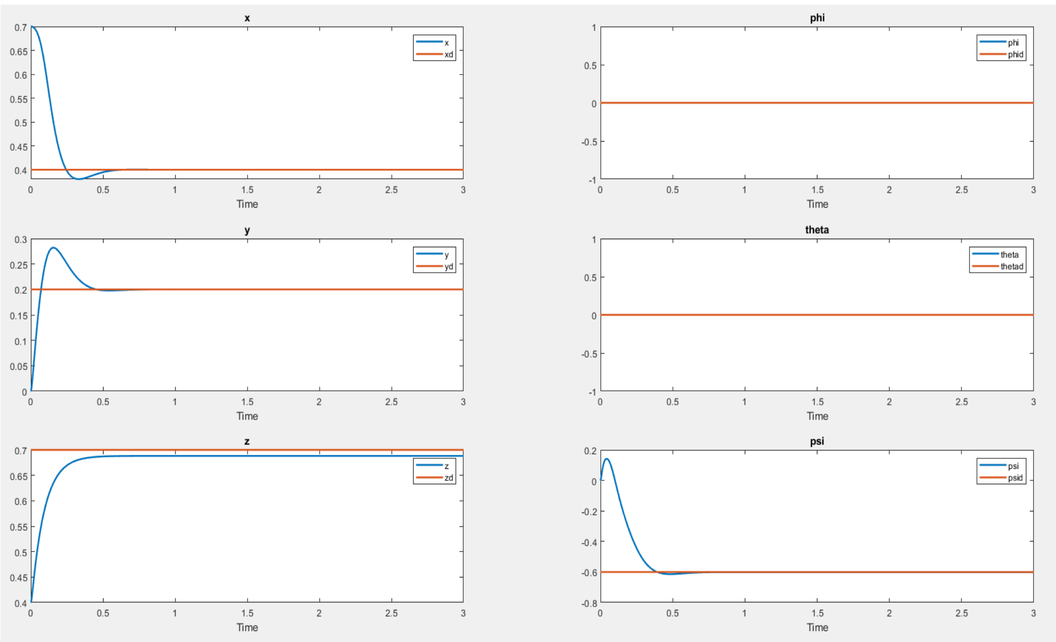

Figure 24: Compliance Control pose with with K << KP

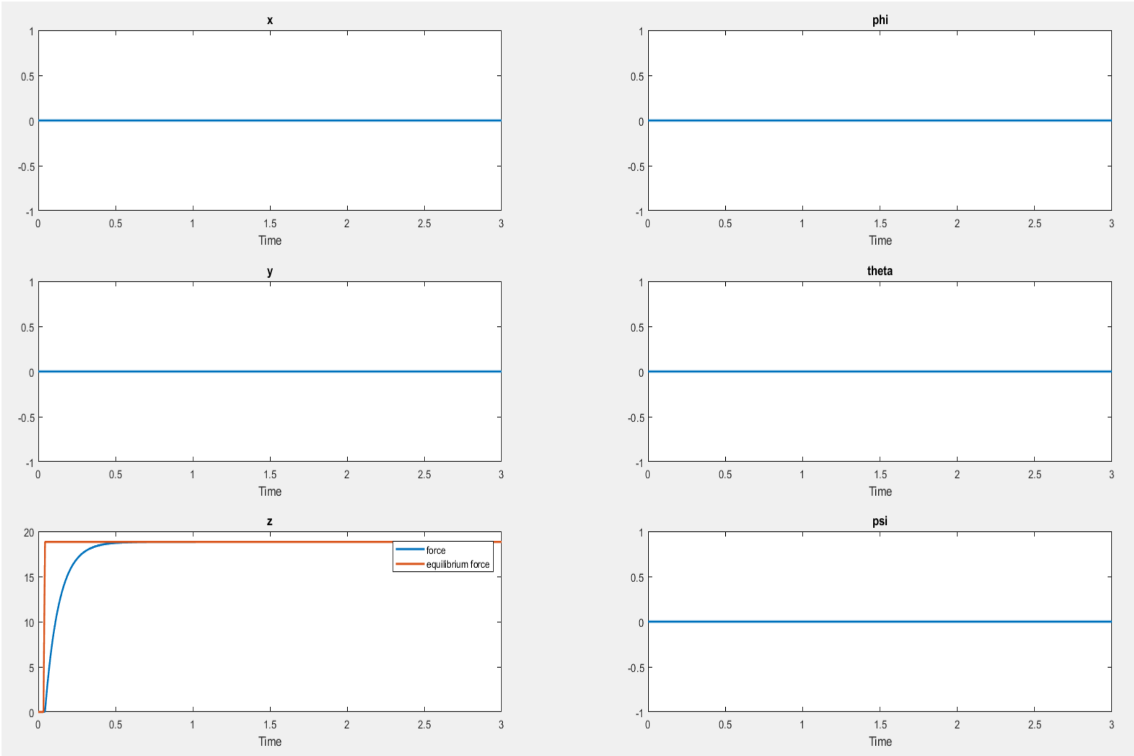

Figure 25: Compliance Control force with stiffer environment

7.2 Impedance Control

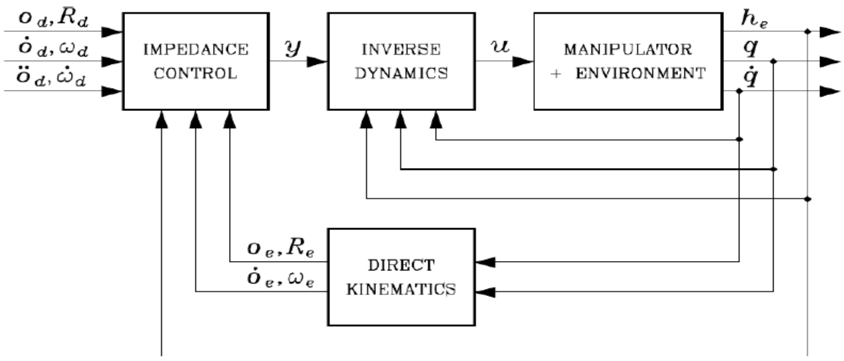

Figure 26: Impedance Control scheme

$$\frac{t_d}{\boldsymbol{q}_d} = \begin{array}{c|c|c} 0 & 1.5 & 5 \\ \hline \pi/4 & \pi/4 & \pi/4 \\ -\pi/6 & -\pi/6 & -\pi/6 \\ 0.1 & 0.2 & 0.1 \end{array}$$

$$\begin{aligned} KP &= \text{diag}(2500, 3000, 1600, 100, 1, 4) \\ KD &= \text{diag}(500, 180, 160, 19, 0, 8) \\ K &= \text{diag}(3, 1, 2, 2, 2, 2) \\ M &= \text{diag}(20, 20, 20, 20, 20, 20) \\ plane &= 0.5 \end{aligned}$$

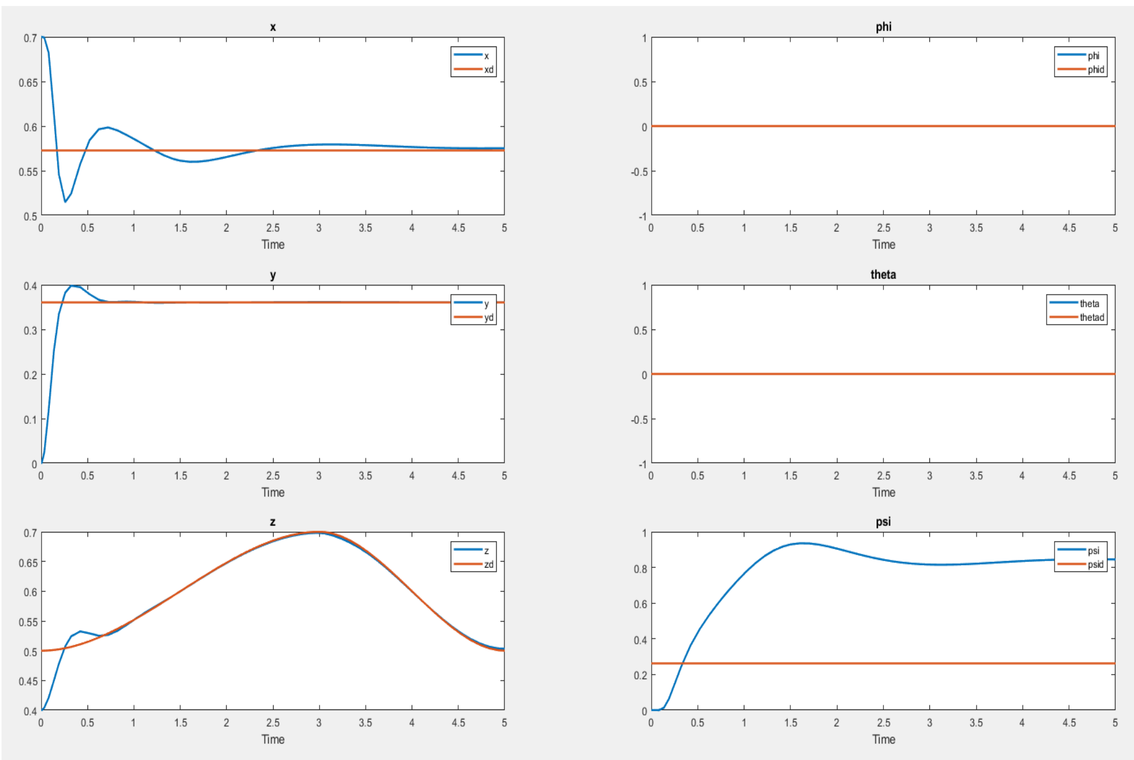

Figure 27: Impedance Control pose

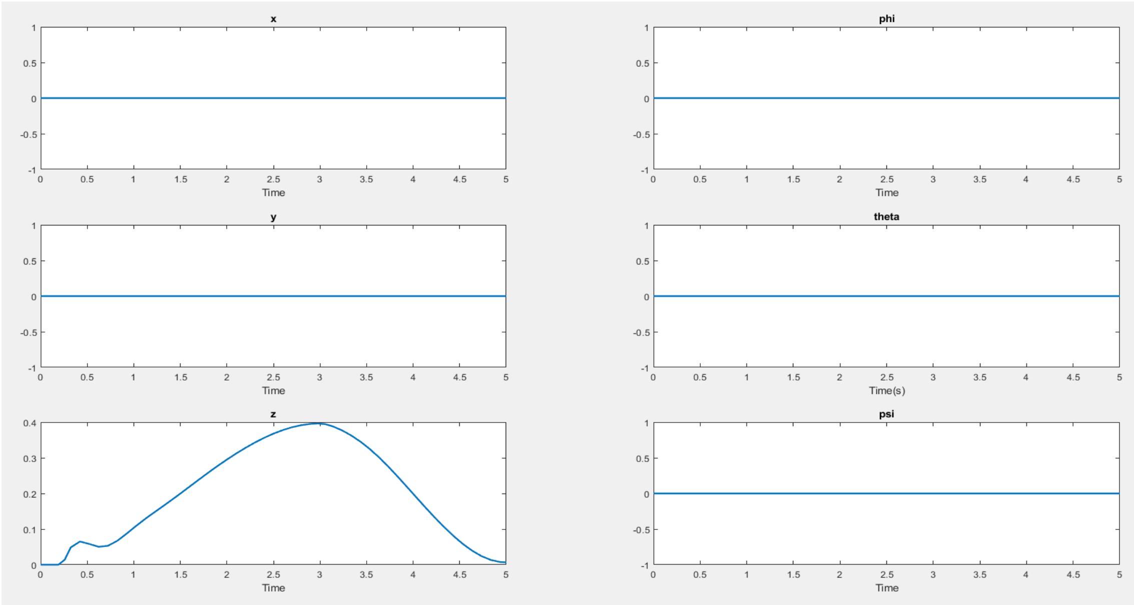

Figure 28: Impedance Control force

$$\frac{t_d}{\boldsymbol{q}_d} = \begin{array}{c|c|c} 0 & 1.5 & 5 \\ \hline \pi/4 & \pi/4 & \pi/4 \\ -\pi/6 & -\pi/6 & -\pi/6 \\ 0.1 & 0.2 & 0.1 \end{array}$$

$$\begin{aligned} KP &= \mathrm{diag}(2500, 3000, 1600, 100, 1, 4) \\ KD &= \mathrm{diag}(500, 180, 160, 19, 0, 8) \\ K &= \mathrm{diag}(3, 1, 200, 2, 2, 2) \\ M &= \mathrm{diag}(20, 20, 20, 20, 20, 20) \\ plane &= 0.5 \end{aligned}$$

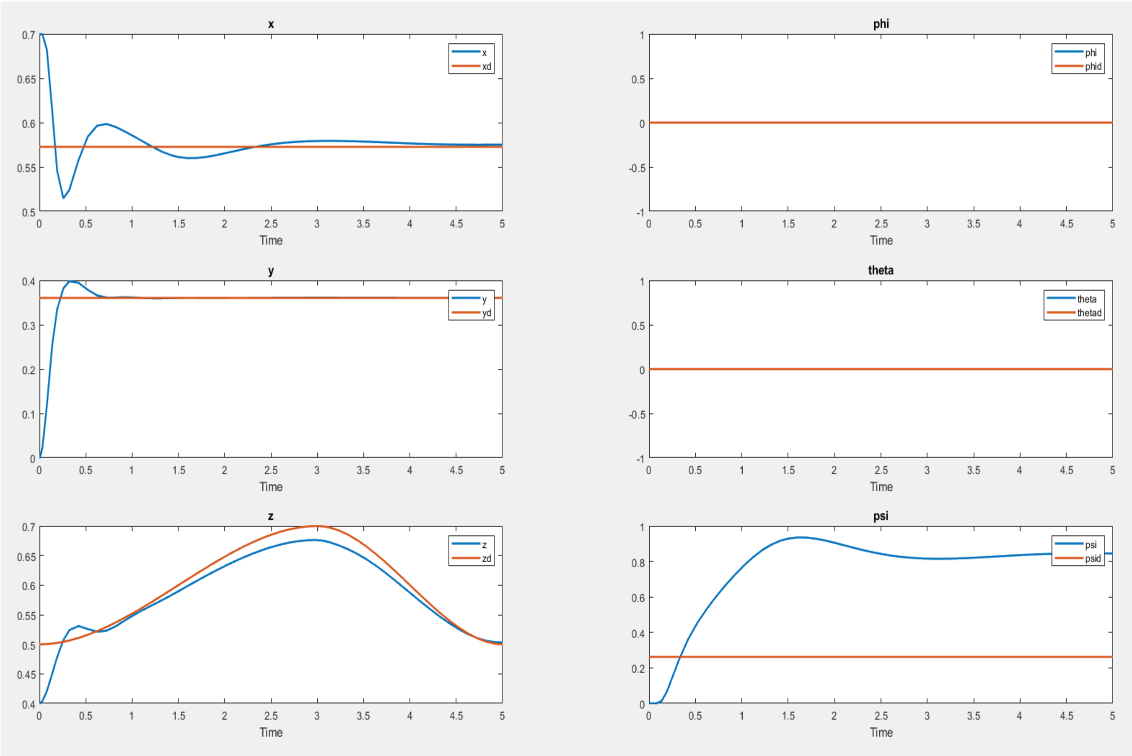

Figure 29: Impedance Control pose with stiffer environment

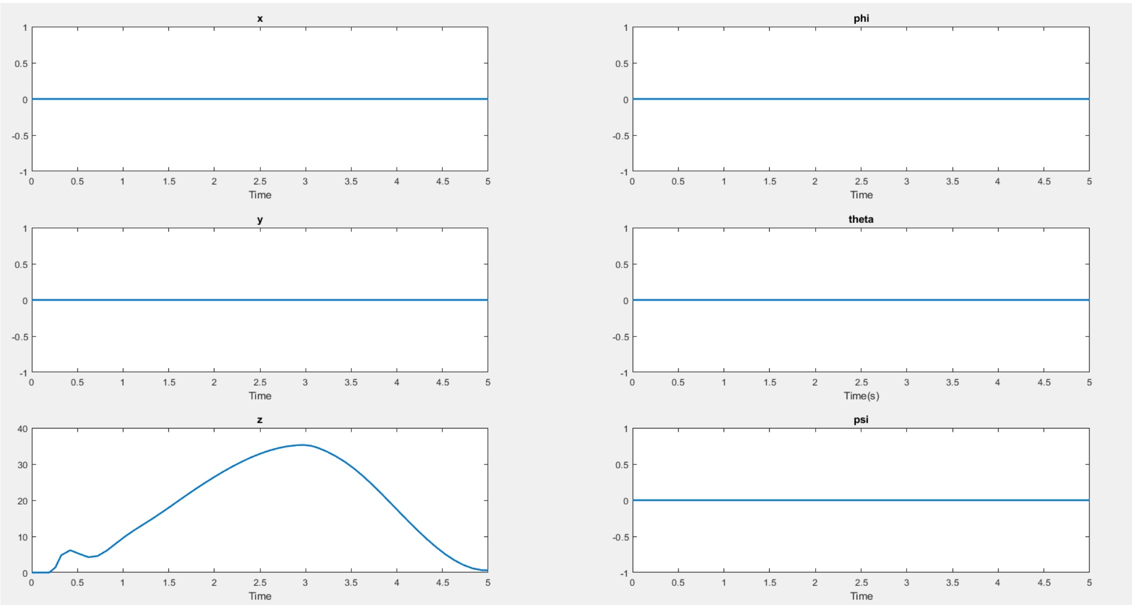

Figure 30: Impedance Control force with stiffer environment

7.3 Admittance Control

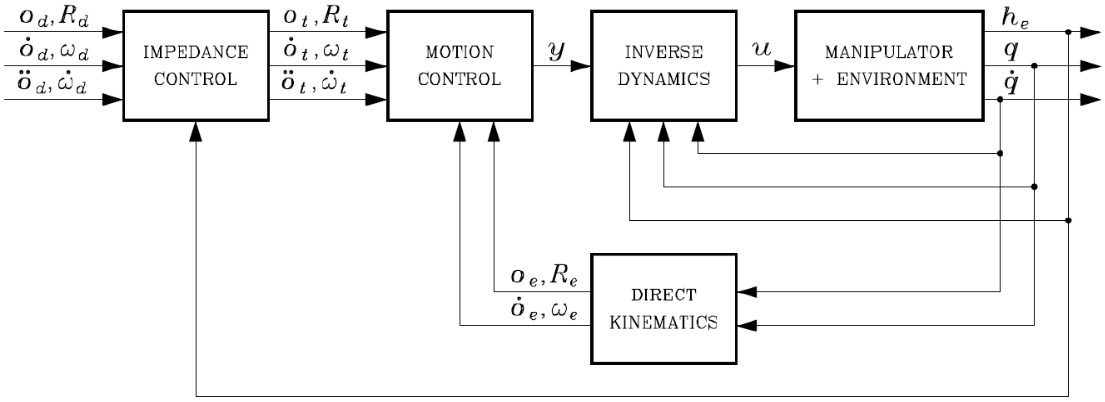

Figure 31: Admittance Control scheme

$$
\frac{t_d}{\boldsymbol{q}_d} = \begin{array}{c|c|c} 0 & 1.5 & 5 \\ \hline \pi/4 & \pi/4 & \pi/6 \\ -\pi/6 & -\pi/6 & -\pi/6 \\ 0.1 & 0.2 & 0.1 \end{array}
$$

$$
\begin{aligned}
plane &= 0.5 \\
K &= \mathrm{diag}(2, 2, 2, 0.1, 0.1, 0.1) \\
Md &= \mathrm{diag}(2, 2, 2, 2, 2, 2) \\
Mdt &= \mathrm{diag}(0.75, 0.75, 0.75, 0.75, 0.75, 0.75)
\end{aligned}
$$

$$
\begin{aligned}
KP &= \mathrm{diag}(1500, 1700, 2025, 1, 1, 100) \\
KD &= \mathrm{diag}(500, 500, 135, 0, 0, 400) \\
KPt &= \mathrm{diag}(100, 100, 100, 100, 100, 100) \\
KDt &= \mathrm{diag}(180, 180, 180, 180, 180, 180)
\end{aligned}
$$

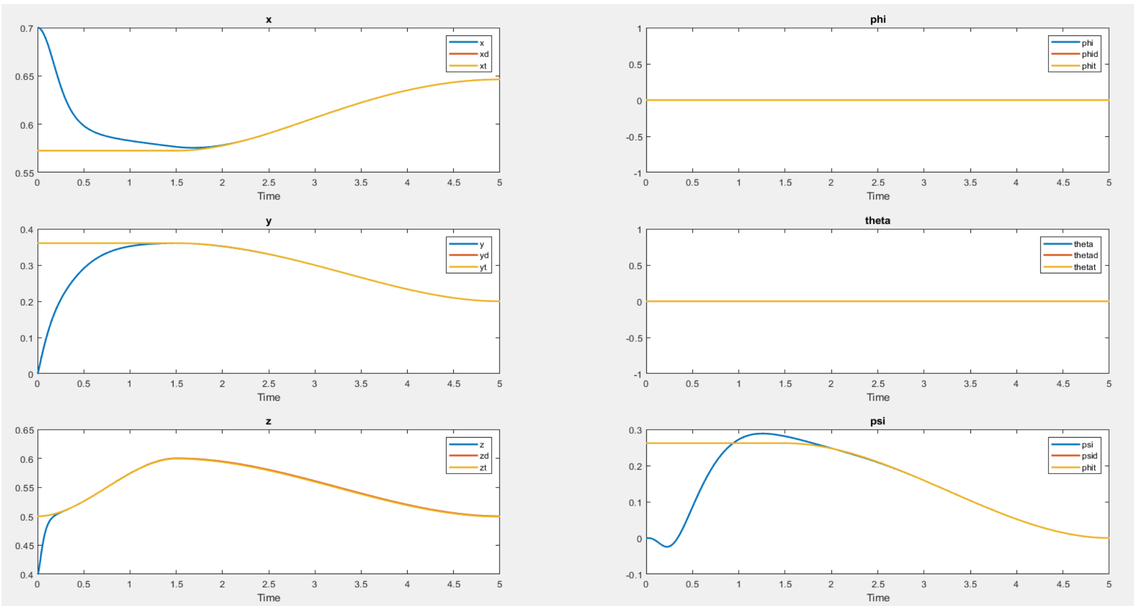

Figure 32: Admittance control position

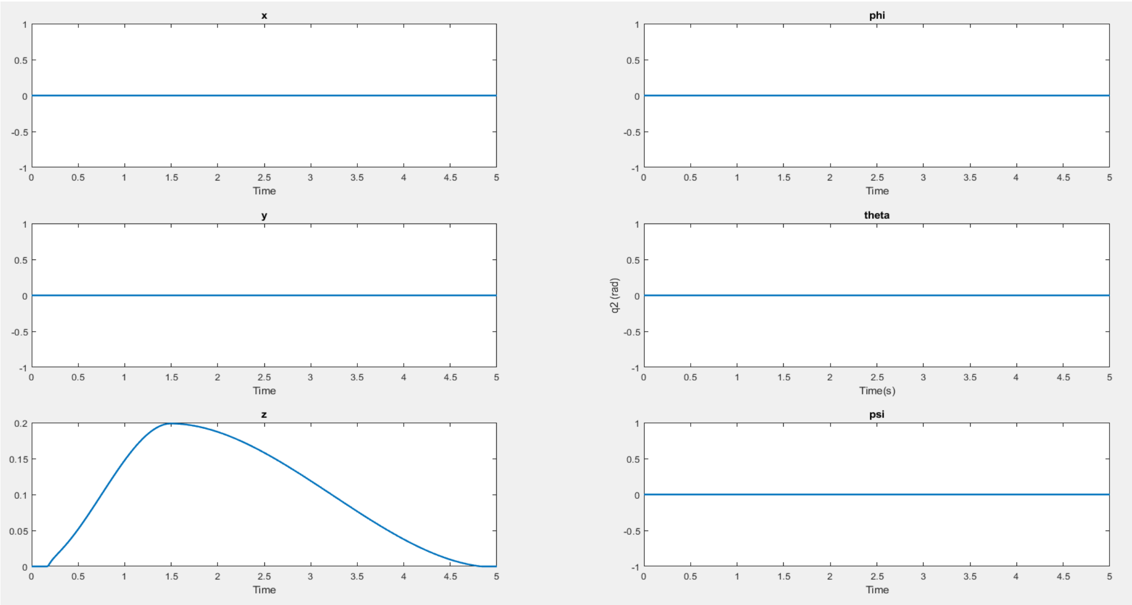

Figure 33: Admittance control force

$$\begin{array}{c|c|c|c} & 0 & 1.5 & 5 \\ \hline \frac{t_d}{\boldsymbol{q}_d} = & \pi/4 & \pi/4 & \pi/6 \\ & -\pi/6 & -\pi/6 & -\pi/6 \\ & 0.1 & 0.2 & 0.1 \end{array}$$

$$plane = 0.5$$

$$\begin{aligned} K &= \mathrm{diag}(2, 2, 200, 2, 2, 2) \\ Md &= \mathrm{diag}(2, 2, 2, 2, 2, 2) \\ Mdt &= \mathrm{diag}(0.75, 0.75, 0.75, 0.75, 0.75, 0.75) \end{aligned}$$

$$\begin{aligned} KP &= \mathrm{diag}(1500, 1700, 2025, 1, 1, 100) \\ KD &= \mathrm{diag}(500, 500, 135, 0, 0, 400) \\ KPt &= \mathrm{diag}(100, 100, 100, 100, 100, 100) \\ KDt &= \mathrm{diag}(180, 180, 180, 180, 180, 180) \end{aligned}$$

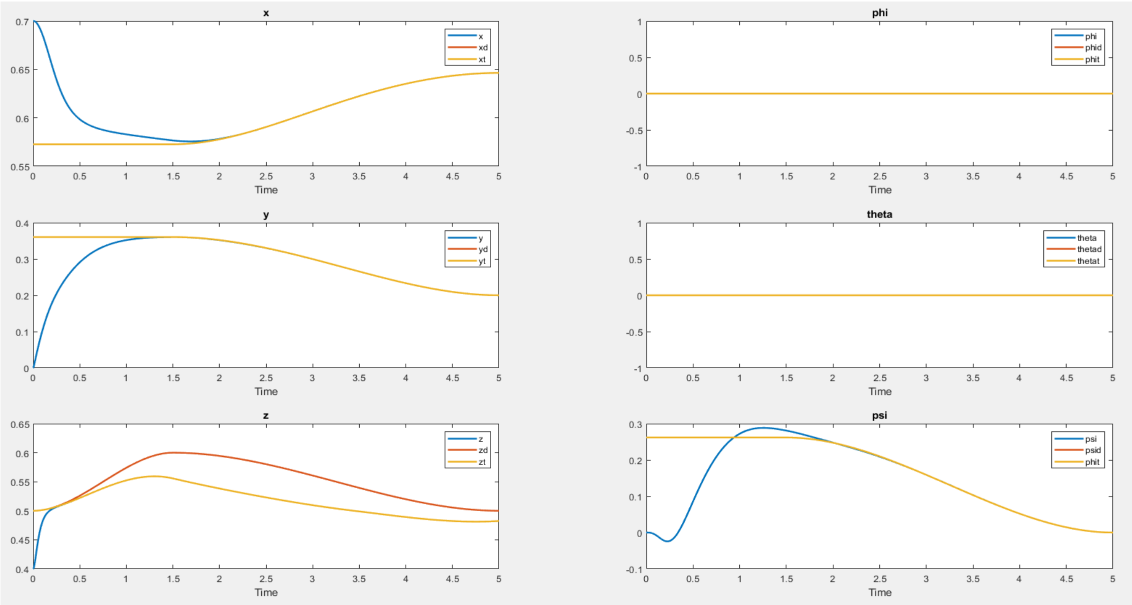

Figure 34: Admittance control position with stiffer environment

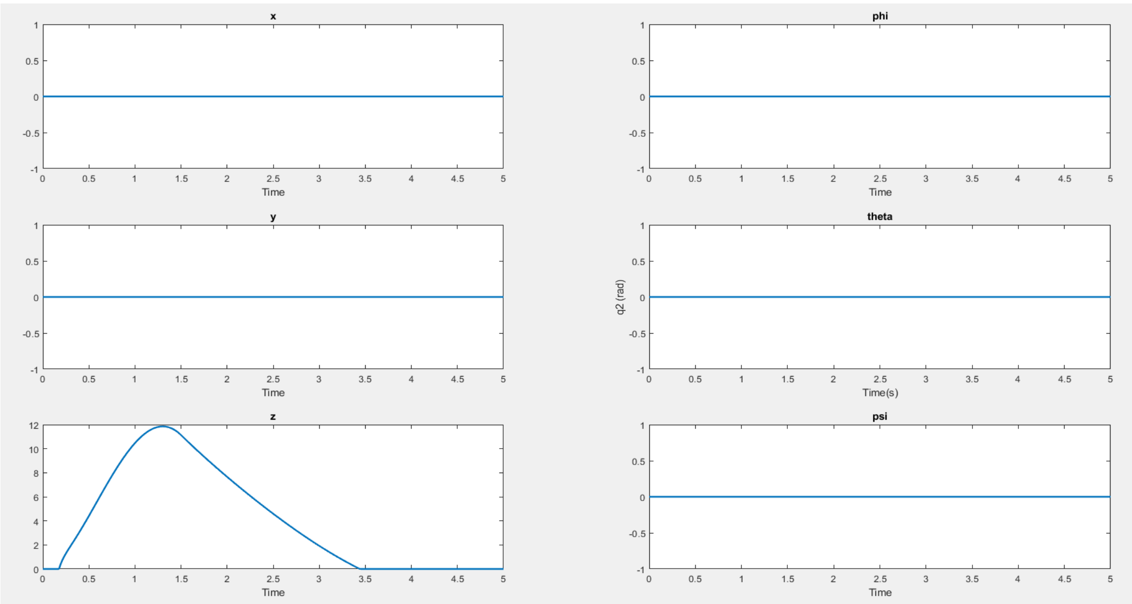

Figure 35: Admittance control force with stiffer environment

7.4 Force Control with Inner Position Loop

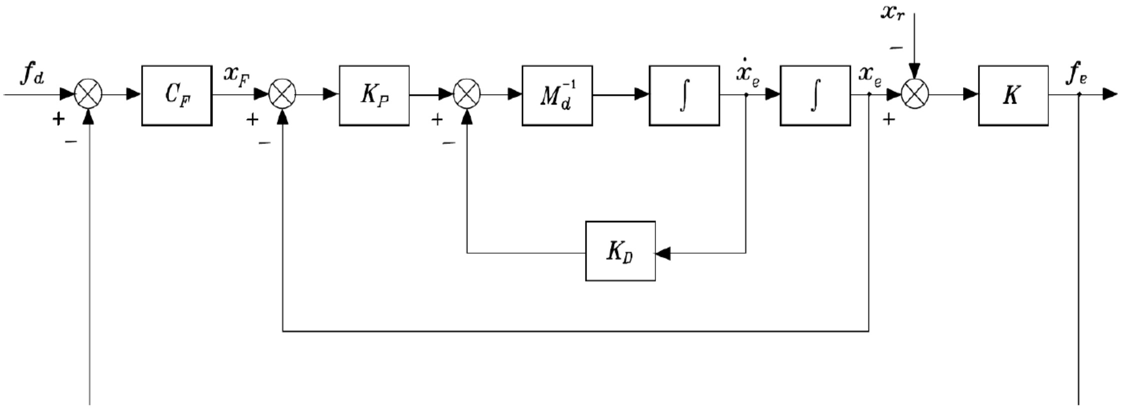

Figure 36: Force Control with Inner Position Loop scheme

$$f_d = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \quad \begin{aligned} KP &= \mathrm{diag}(20, 20, 20) \\ KD &= \mathrm{diag}(15, 15, 15) \\ M_d &= \mathrm{diag}(0.5, 0.5, 0.5) \end{aligned} \quad \begin{aligned} KFP &= \mathrm{diag}(0.5, 0.5, 0.5) \\ KFI &= \mathrm{diag}(1.5, 1.5, 1.5) \end{aligned} \quad \begin{aligned} K &= \mathrm{diag}(12, 10, 15, 0.1, 0.1, 0.1) \\ plane &= 0.5 \end{aligned}$$

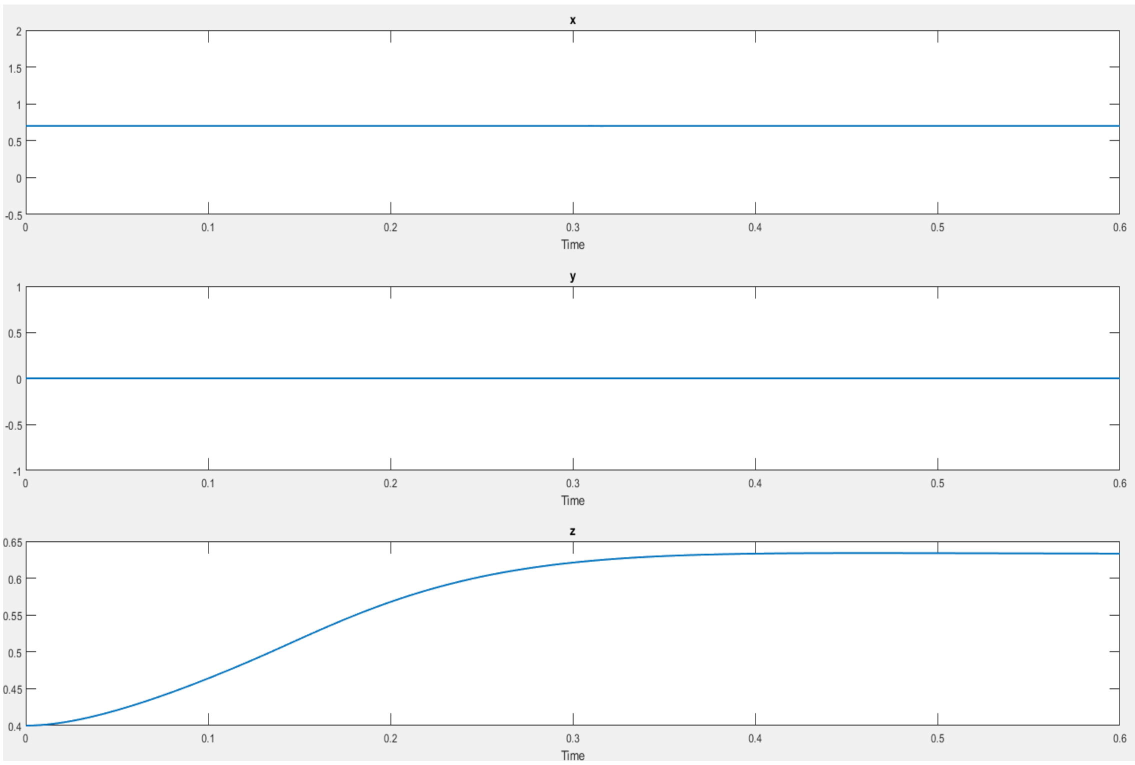

Figure 37: Force Control with inner position loop pose with integral action

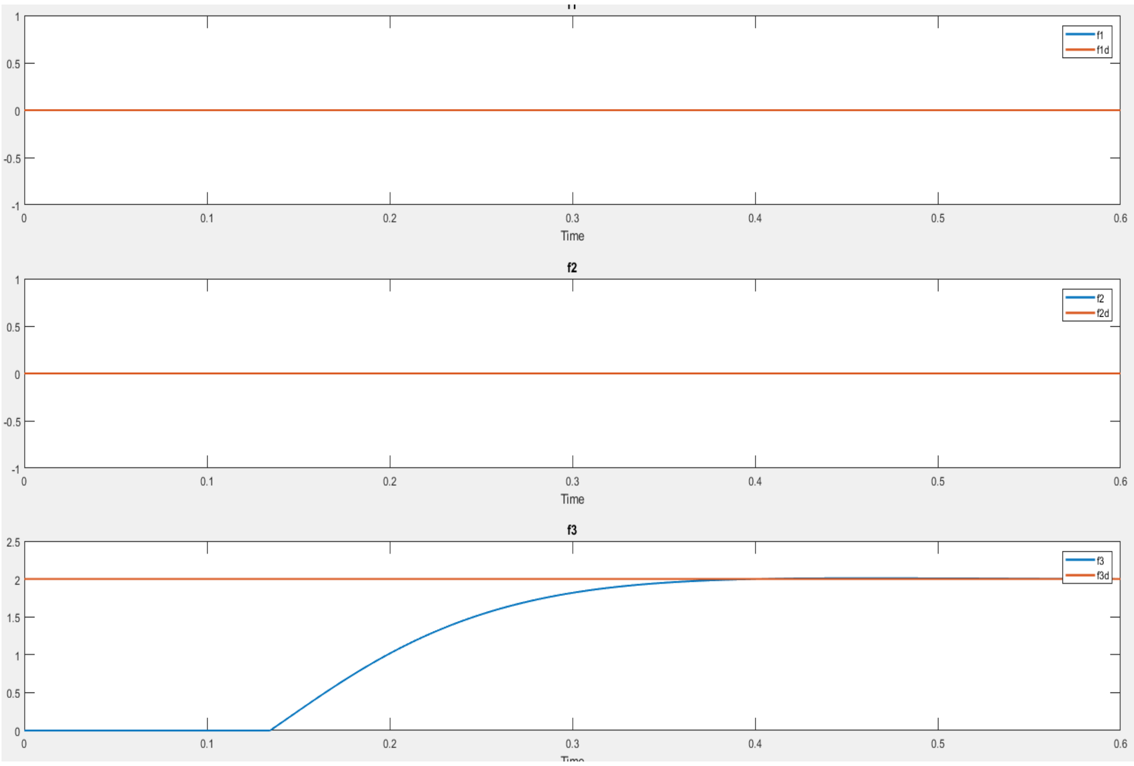

Figure 38: Force Control with inner position loop force with integral action

$$f_d = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \qquad \begin{array}{l} KP = \mathrm{diag}(20, 20, 20) \\ KD = \mathrm{diag}(15, 15, 15) \\ M_d = \mathrm{diag}(0.5, 0.5, 0.5) \end{array} \qquad \begin{array}{l} KFP = \mathrm{diag}(0.5, 0.5, 0.5) \\ KFI = \mathrm{diag}(0, 0, 0) \end{array} \qquad \begin{array}{l} K = \mathrm{diag}(12, 10, 15, 0.1, 0.1, 0.1) \\ plane = 0.5 \end{array}$$

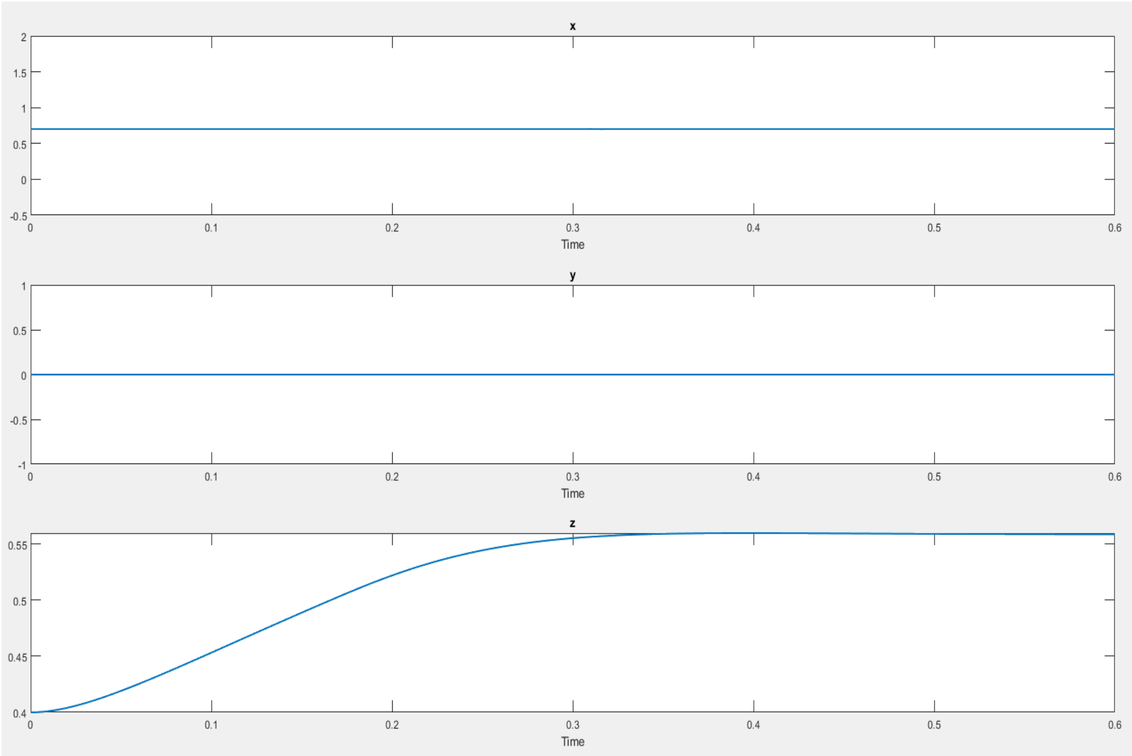

Figure 39: Force Control with inner position loop pose without integral action

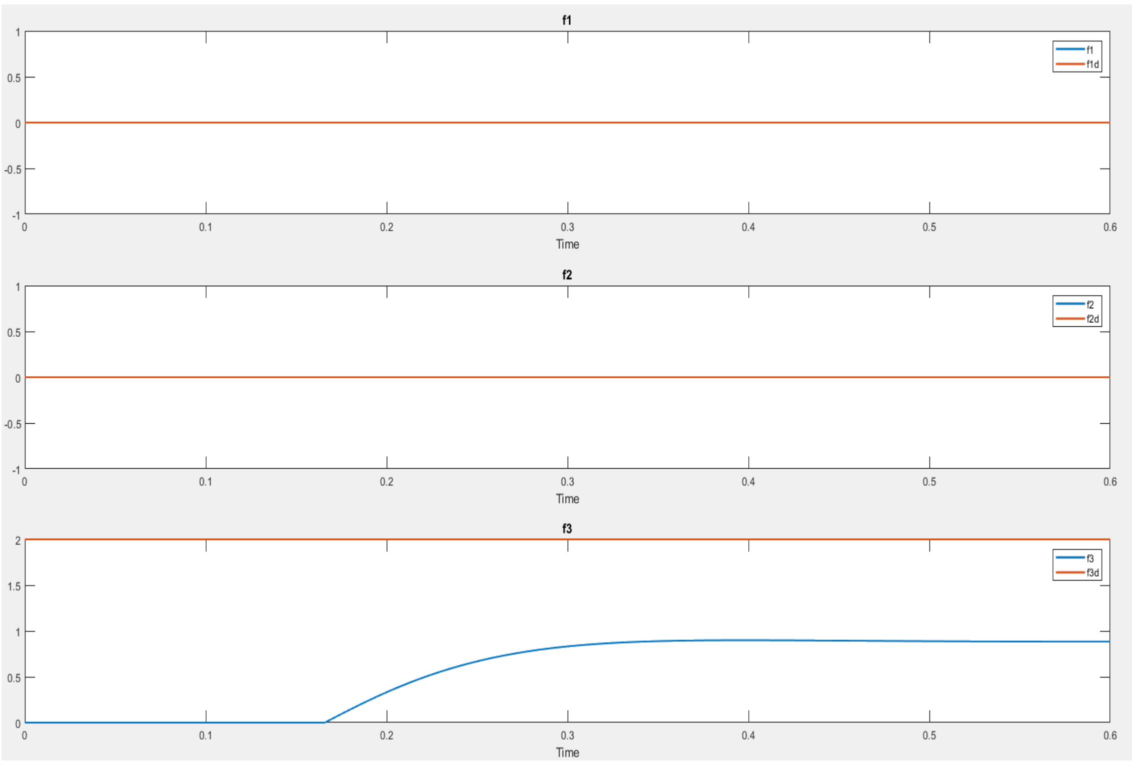

Figure 40: Force Control with inner position loop force without integral action

7.5 Parallel Force/Position Control

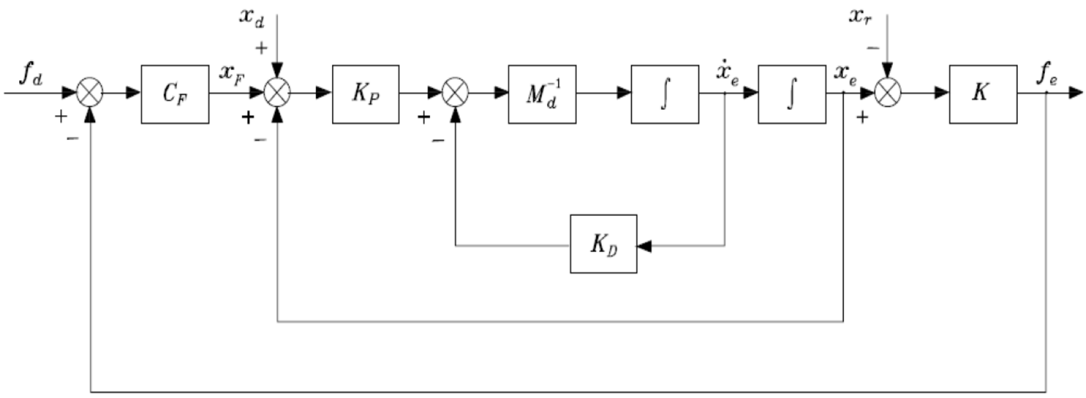

Figure 41: Parallel Force/Position Control scheme

$$f_d = \begin{bmatrix} 0 \\ 0 \\ 8 \end{bmatrix} \quad KP = \text{diag}(3700, 3700, 370) \quad KFP = \text{diag}(0.5, 0.5, 0.5) \quad K = \text{diag}(12, 10, 15, 0.1, 0.1, 0.1)$$

$$KD = \text{diag}(160, 180, 55) \quad KFI = \text{diag}(1.5, 1.5, 1.5) \quad plane = 0.5$$

$$x_d = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.9 \end{bmatrix} \quad M_d = \text{diag}(0.01, 0.01, 0.1)$$

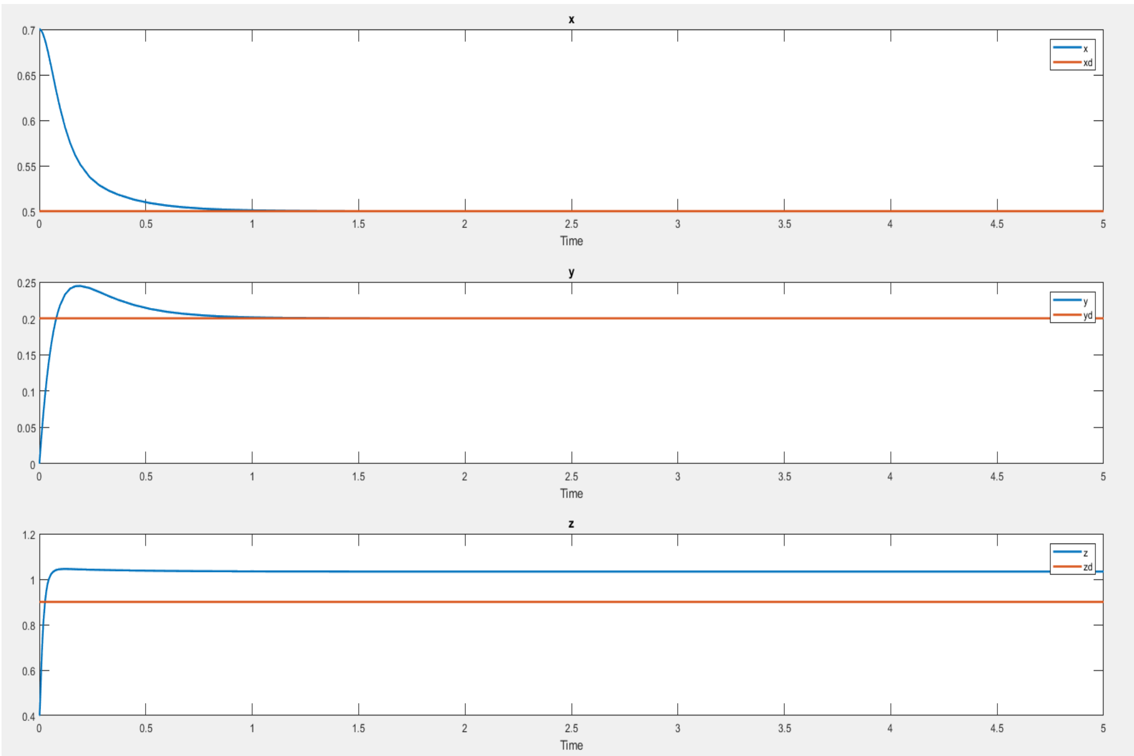

Figure 42: Parallel Force/Position Control pose with integral action

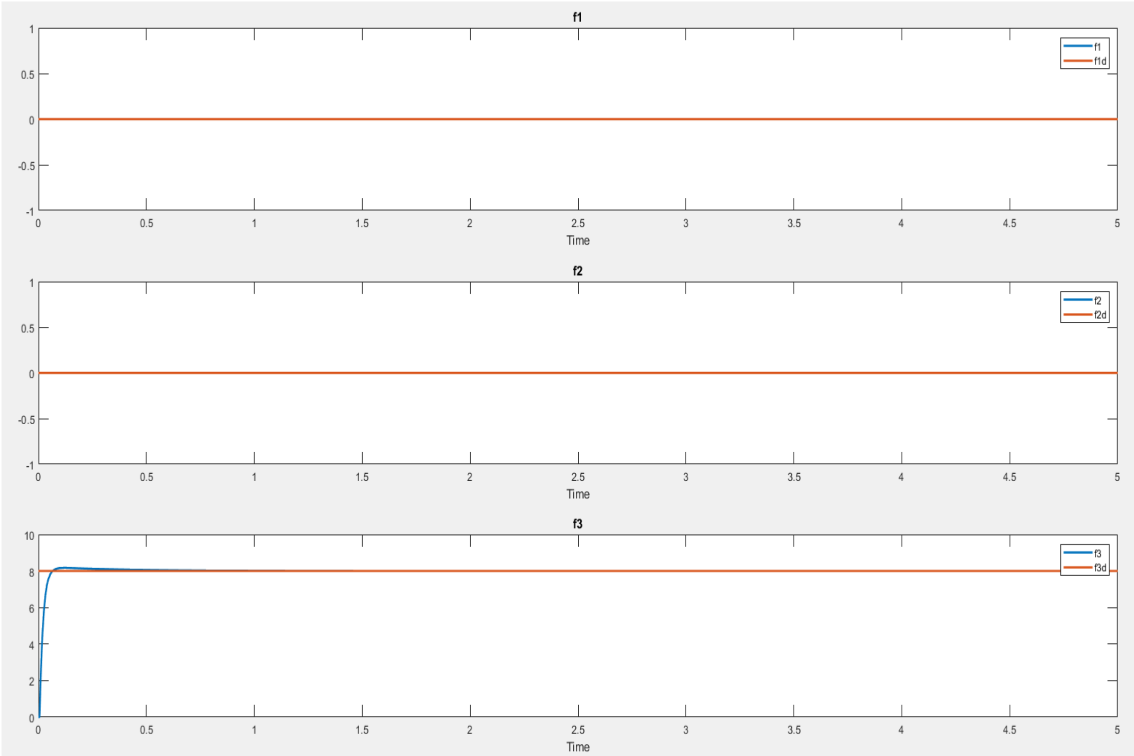

Figure 43: Parallel Force/Position Control force with integral action

$$f_d = \begin{bmatrix} 0 \\ 0 \\ 8 \end{bmatrix} \qquad x_d = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.9 \end{bmatrix}$$

$$KP = \text{diag}(3700, 3700, 370) \quad KD = \text{diag}(160, 180, 55) \quad M_d = \text{diag}(0.01, 0.01, 0.1)$$

$$KFP = \text{diag}(0.5, 0.5, 0.5) \quad KFI = \text{diag}(0, 0, 0)$$

$$K = \text{diag}(12, 10, 15, 0.1, 0.1, 0.1) \quad plane = 0.5$$

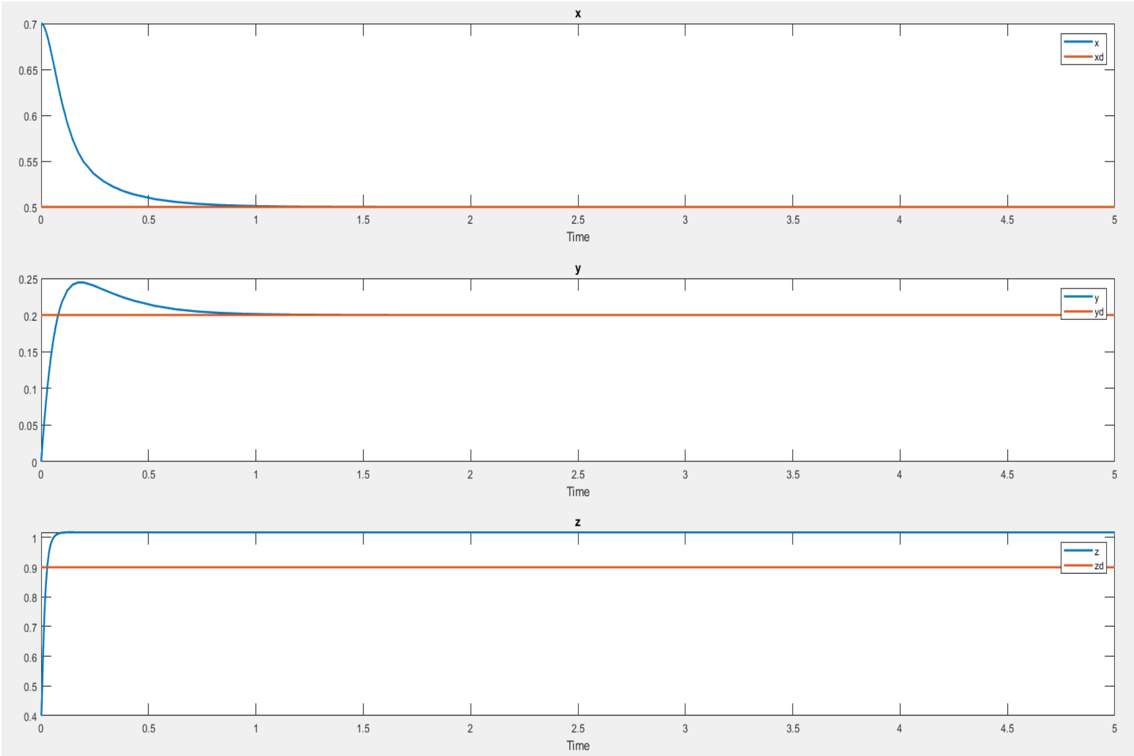

Figure 44: Parallel Force/Position Control pose without integral action

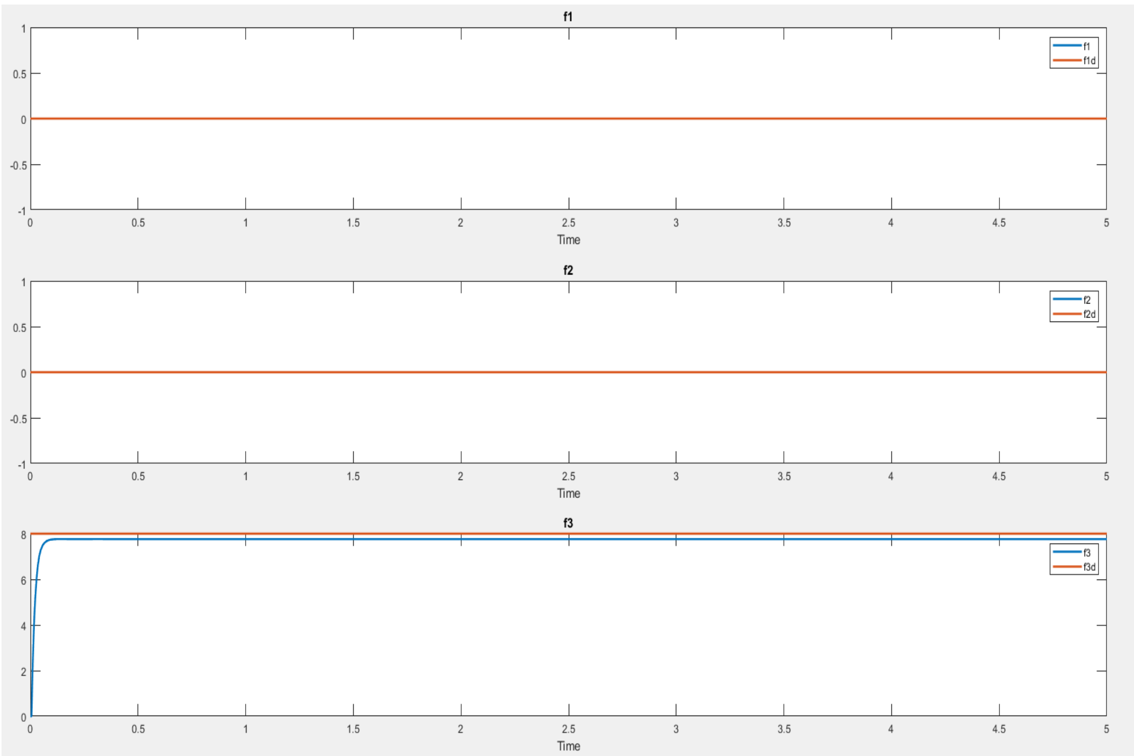

Figure 45: Parallel Force/Position Control force without integral action