Advanced Control Systems

Palladino Luigi

October 2020

Contents

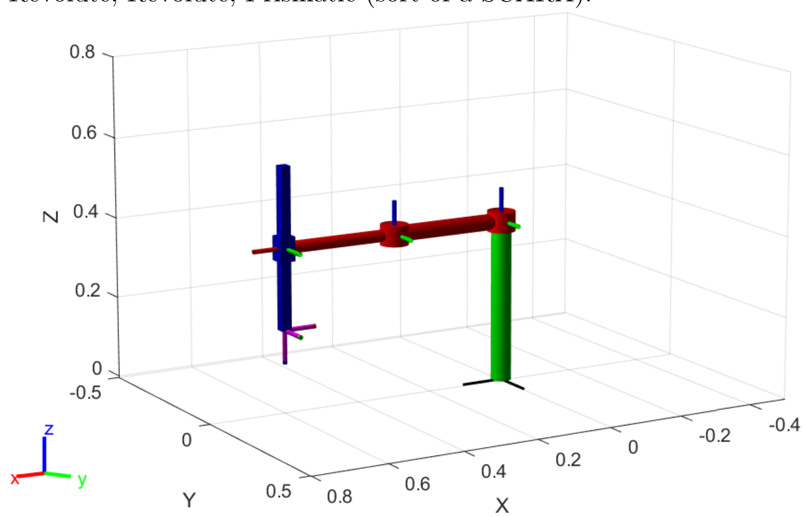
Chapter 1

Introduction

Robot with 3 degrees of freedom (3 DoF).

Three Joints in the sequence:

Revolute, Revolute, Prismatic (sort of a SCARA).



Chapter 2

First Assignment

2.1 DH table

Σ_i	a_i	α_i	d_i	θ_i
0	0	0	L_0	0
1	L_1	0	0	q_1
2	L_2	0	0	q_2
3	0	$-\pi$	$\frac{-L_3}{2} + q_3$	π
e	0	0	0	0

L_0, L_1, L_2, L_3 are constants. q_1, q_2, q_3 are joint variables.

2.2 Direct Kinematics

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} c_1 & -s_1 & 0 & L_1 c_1 \\ s_1 & c_1 & 0 & L_1 s_1 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} c_1 c_2 - s_1 s_2 & -c_1 s_2 - c_2 s_1 & 0 & L_1 c_1 + L_2 c_1 c_2 - L_2 s_1 s_2 \\ c_1 s_2 + c_2 s_1 & c_1 c_2 - s_1 s_2 & 0 & L_1 s_1 + L_2 c_1 s_2 + L_2 c_2 s_1 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} s_1 s_2 - c_1 c_2 & -c_1 s_2 - c_2 s_1 & 0 & L_1 c_1 + L_2 c_1 c_2 - L_2 s_1 s_2 \\ -c_1 s_2 - c_2 s_1 & c_1 c_2 - s_1 s_2 & 0 & L_1 s_1 + L_2 c_1 s_2 + L_2 c_2 s_1 \\ 0 & 0 & -1 & L_0 - \frac{L_3}{2} + q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T = T4 = T1*T2*T3

$$\text{Direct Kinematics} = T = \begin{bmatrix} -c_12 & -s_12 & 0 & L_2c_12 + L_1c_1 \\ -s_12 & c_12 & 0 & L_2s_12 + L_1s_1 \\ 0 & 0 & -1 & L_0 - \frac{L_3}{2} + q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3 Inverse Kinematics

To find the inverse kinematics for the given direct kinematics matrix, we need to solve for the joint angles (q1, q2, q3) in terms of the end-effector position and orientation.

Let's denote the given direct kinematics matrix as T, and break it down into translation and rotation components:

Translation:

$$L2cos(q1+q2) + L1cos(q1), L2sin(q1+q2) + L1sin(q1), L0 - L3/2 + q3$$

Rotation:

$$[-cos(q1+q2), -sin(q1+q2), 0]$$
$$[-sin(q1+q2), cos(q1+q2), 0]$$
$$[0, 0, -1]$$

To find the inverse kinematics, we need to solve the equations:

$$L2cos(q1+q2) + L1cos(q1) = x$$
$$L2sin(q1+q2) + L1sin(q1) = y$$
$$L0 - L3/2 + q3 = z$$

Solve for q3: From the third equation: $L0 - L3/2 + q3 = z$ Rearranging the terms: $q3 = z - L0 + L3/2$

where (x, y, z) are the desired end-effector coordinates.

Solving these equations will give us the values of q1, q2, and q3 that correspond to the desired end-effector position and orientation.

$$\begin{gathered} P_x = L_2\cos(q_1+q_2) + L_1\cos(q_1) \\ P_y = L_2\sin(q_1+q_2) + L_1\sin(q_1) \\ P_z = L_0 - \frac{L_3}{2} + q_3 \\ \downarrow \\ q_1 = \operatorname{atan2}(P_y, P_x) - \operatorname{atan2}(L_2\sin(q_2), L_1 + (L_2\cos(q_2)) \\ q_2 = -\cos^{-1}\left(\frac{P_x^2+P_y^2-L_1^2-L_2^2}{2L_1L_2}\right) \\ q_3 = P_z - L_0 + \frac{L_3}{2} \end{gathered}$$

2.4 Geometric Jacobian

$$J_g = \begin{bmatrix} -L_1 s_1 - L_2 c_1 s_2 - L_2 c_2 s_1 & -L_2 c_1 s_2 - L_2 c_2 s_1 & 0 \\ L_1 c_1 + L_2 c_1 c_2 - L_2 s_1 s_2 & L_2 c_1 c_2 - L_2 s_1 s_2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

2.5 Analytical Jacobian

Analitical Jacobian is obtained using ZYZ Euler angles as follows:

$J_a = T^{-1} \cdot J$

$T = \begin{bmatrix} I_3 & 0 \\ 0 & T_a \end{bmatrix}$

$T_a = \begin{bmatrix} 0 & -s\phi & c\phi s\theta \\ 0 & c\phi & s\phi s\theta \\ 1 & 0 & c\theta \end{bmatrix}$

$$J_A = \begin{bmatrix} -L_2 s_1 2 - L_1 s_1 & -L_2 s_{12} & 0 \\ L_2 c_1 2 + L_1 c_1 & L_2 c_{12} & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Chapter 3

Second Assignment

3.1 Kinetic Energy

To obtain the kinetic energy

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

We now need the inertia tensors with respect to Σ_i and the partial geometric Jacobians of each link. Let's start with the $I_{\ell_i}^i$ tensors: we have two solid cylinders (link 1 and link 2, with radii R_1 and R_2) and a solid parallelepiped (link 3, with a square base of side length L_3). However, all the frames Σ_i are translated by $p_{\ell_i}^i$ with respect to their respective centers of mass (by definition of $p_{\ell_i}^i$). Therefore, the inertia tensors are obtained using the Steiner's theorem.

$$
\begin{aligned}
I_{\ell_1}^1 &= I_{\ell_1}^{C_1} + m_{\ell_1}\left(r^\top r\mathbb{I} - rr^\top\right) \qquad \text{dove} \qquad r = p_{\ell_1}^1 \\
&= m_{\ell_1}\begin{bmatrix} \frac{1}{2}R_1^2 & & \\ & \frac{1}{4}R_1^2 + \frac{1}{12}\ell_1^2 & \\ & & \frac{1}{4}R_1^2 + \frac{1}{12}\ell_1^2 \end{bmatrix} + m_{\ell_1}\begin{bmatrix} 0 & & \\ & \frac{1}{4}\ell_1^2 & \\ & & \frac{1}{4}\ell_1^2 \end{bmatrix} \\
&= \frac{1}{12}m_{\ell_1}\begin{bmatrix} 6R_1^2 & & \\ & 3R_1^2 + 4\ell_1^2 & \\ & & 3R_1^2 + 4\ell_1^2 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
I_{\ell_2}^2 &= I_{\ell_2}^{C_2} + m_{\ell_2}\left(r^\top r\mathbb{I} - rr^\top\right) \qquad \text{dove} \qquad r = p_{\ell_2}^2 \\
&= m_{\ell_2}\begin{bmatrix} \frac{1}{2}R_2^2 & & \\ & \frac{1}{4}R_2^2 + \frac{1}{12}\ell_2^2 & \\ & & \frac{1}{4}R_2^2 + \frac{1}{12}\ell_2^2 \end{bmatrix} + m_{\ell_2}\begin{bmatrix} 0 & & \\ & \frac{1}{4}\ell_2^2 & \\ & & \frac{1}{4}\ell_2^2 \end{bmatrix} \\
&= \frac{1}{12}m_{\ell_2}\begin{bmatrix} 6R_2^2 & & \\ & 3R_2^2 + 4\ell_2^2 & \\ & & 3R_2^2 + 4\ell_2^2 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
I_{\ell_3}^3 &= I_{\ell_3}^{C_3} + m_{\ell_3}\left(r^\top r\mathbb{I} - rr^\top\right) \qquad \text{dove} \qquad r = p_{\ell_3}^3 \\
&= \frac{1}{12}m_{\ell_3}\begin{bmatrix} L_3^2 + \ell_3^2 & & \\ & L_3^2 + \ell_3^2 & \\ & & L_3^2 + L_3^2 \end{bmatrix} + m_{\ell_2}\begin{bmatrix} \frac{1}{4}\ell_3^2 & & \\ & \frac{1}{4}\ell_3^2 & \\ & & 0 \end{bmatrix} \\
&= \frac{1}{12}m_{\ell_3}\begin{bmatrix} L_3^2 + 4\ell_3^2 & & \\ & L_3^2 + 4\ell_3^2 & \\ & & 2L_3^2 \end{bmatrix}
\end{aligned}
$$

Now let's proceed with the partial geometric Jacobians J^{ℓ_i} of each link: we keep the upper part $J_P^{\ell_i}$ separate from the lower part $J_O^{\ell_i}$, which we recall are slightly different from the classical geometric Jacobian.

$$
\begin{aligned}
J_P^{\ell_i} &= \begin{bmatrix} J_{P_1}^{\ell_i} & \cdots & J_{P_j}^{\ell_i} & \cdots & J_{P_i}^{\ell_i} & \emptyset & \cdots & \emptyset \end{bmatrix} \\
J_O^{\ell_i} &= \begin{bmatrix} J_{O_1}^{\ell_i} & \cdots & J_{O_j}^{\ell_i} & \cdots & J_{O_i}^{\ell_i} & \emptyset & \cdots & \emptyset \end{bmatrix}
\end{aligned}
$$

and they are formed by blocks (which are also slightly different).

	Prismatic	*Rotational*
Linear	$R_{j-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$R_{j-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_i} - d_{j-1}^0)$
Angular	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$R_{j-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Let's perform all the calculations:

$$J_P^{\ell_1} = \begin{bmatrix} J_{P_1}^{\ell_1} & \emptyset & \emptyset \end{bmatrix} \qquad J_O^{\ell_1} = \begin{bmatrix} J_{O_1}^{\ell_1} & \emptyset & \emptyset \end{bmatrix}$$

$$J_{P_1}^{\ell_1} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_1} - d_0^0) = \begin{bmatrix} -\frac{\ell_1}{2} s_1 \\ \frac{\ell_1}{2} c_1 \\ 0 \end{bmatrix} \qquad J_{O_1}^{\ell_1} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_P^{\ell_2} = \begin{bmatrix} J_{P_1}^{\ell_2} & J_{P_2}^{\ell_2} & \emptyset \end{bmatrix} \qquad J_O^{\ell_2} = \begin{bmatrix} J_{O_1}^{\ell_2} & J_{O_2}^{\ell_2} & \emptyset \end{bmatrix}$$

$$J_{P_1}^{\ell_2} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_2} - d_0^0) = \begin{bmatrix} -\ell_1 s_1 - \frac{\ell_2}{2} s_{12} \\ \ell_1 c_1 + \frac{\ell_2}{2} c_{12} \\ 0 \end{bmatrix} \qquad J_{O_1}^{\ell_2} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_{P_2}^{\ell_2} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_2} - d_1^0) = \begin{bmatrix} -\frac{\ell_2}{2} s_{12} \\ \frac{\ell_2}{2} c_{12} \\ 0 \end{bmatrix} \qquad J_{O_2}^{\ell_2} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_P^{\ell_3} = \begin{bmatrix} J_{P_1}^{\ell_3} & J_{P_2}^{\ell_3} & J_{P_3}^{\ell_3} \end{bmatrix} \qquad J_O^{\ell_3} = \begin{bmatrix} J_{O_1}^{\ell_3} & J_{O_2}^{\ell_3} & J_{O_3}^{\ell_3} \end{bmatrix}$$

$$J_{P_1}^{\ell_3} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_3} - d_0^0) = \begin{bmatrix} -\ell_1 s_1 - \ell_2 s_{12} \\ \ell_1 c_1 + \ell_2 c_{12} \\ 0 \end{bmatrix} \qquad J_{O_1}^{\ell_3} = R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_{P_2}^{\ell_3} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (p_{\ell_3} - d_1^0) = \begin{bmatrix} -\ell_2 s_{12} \\ \ell_2 c_{12} \\ 0 \end{bmatrix} \qquad J_{O_2}^{\ell_3} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

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

In conclusion, then

$$J_P^{\ell_1} = \begin{bmatrix} -\frac{\ell_1}{2}s_1 & 0 & 0 \\ \frac{\ell_1}{2}c_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad J_P^{\ell_2} = \begin{bmatrix} -\ell_1 s_1 - \frac{\ell_2}{2}s_{12} & -\frac{\ell_2}{2}s_{12} & 0 \\ \ell_1 c_1 + \frac{\ell_2}{2}c_{12} & \frac{\ell_2}{2}c_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad J_P^{\ell_3} = \begin{bmatrix} -\ell_1 s_1 - \ell_2 s_{12} & -\ell_2 s_{12} & 0 \\ \ell_1 c_1 + \ell_2 c_{12} & \ell_2 c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_O^{\ell_1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad J_O^{\ell_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad J_O^{\ell_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Now we just need to compute the single inertia matrices $B_i(\boldsymbol{q})$

$$\begin{aligned} B_1(\boldsymbol{q}) &= m_{\ell_1}\big({J_P^{\ell_1}}^\top J_P^{\ell_1}\big) + \big({R_1^0}^\top J_O^{\ell_1}\big)^\top I_{\ell_1}^1 \big({R_1^0}^\top J_O^{\ell_1}\big) \\ &= \frac{1}{2}m_{\ell_1} \begin{bmatrix} 3\ell_1^2 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} + \frac{1}{2}m_{\ell_1} \begin{bmatrix} 4\ell_1^2 + 3R_1^2 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} \\ &= \frac{1}{2}m_{\ell_1} \begin{bmatrix} 7\ell_1^2 + 3R_1^2 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} B_2(\boldsymbol{q}) &= m_{\ell_2}\big({J_P^{\ell_2}}^\top J_P^{\ell_2}\big) + \big({R_2^0}^\top J_O^{\ell_2}\big)^\top I_{\ell_2}^2 \big({R_2^0}^\top J_O^{\ell_2}\big) \\ &= \frac{1}{2}m_{\ell_2} \begin{bmatrix} 12\ell_1^2 + 3\ell_2^2 + 12\ell_1\ell_2 c_2 & 3\ell_2^2 + 6\ell_1\ell_2 c_2 & 0 \\ * & 3\ell_2^2 & 0 \\ * & * & 0 \end{bmatrix} + \frac{1}{2}m_{\ell_2} \begin{bmatrix} 4\ell_2^2 + 3R_2^2 & 4\ell_2^2 + 3R_2^2 & 0 \\ * & 4\ell_2^2 + 3R_2^2 & 0 \\ * & * & 0 \end{bmatrix} \\ &= \frac{1}{2}m_{\ell_2} \begin{bmatrix} 12\ell_1^2 + 7\ell_2^2 + 12\ell_1\ell_2 c_2 + 3R_2^2 & 7\ell_2^2 + 6\ell_1\ell_2 c_2 + 3R_2^2 & 0 \\ * & 7\ell_2^2 + 3R_2^2 & 0 \\ * & * & 0 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} B_3(\boldsymbol{q}) &= m_{\ell_3}\big({J_P^{\ell_3}}^\top J_P^{\ell_3}\big) + \big({R_3^0}^\top J_O^{\ell_3}\big)^\top I_{\ell_3}^3 \big({R_3^0}^\top J_O^{\ell_3}\big) \\ &= \frac{1}{12}m_{\ell_3} \begin{bmatrix} 12\ell_1^2 + 12\ell_2^2 + 24\ell_1\ell_2 c_2 & 12\ell_2^2 + 12\ell_1\ell_2 c_2 & 0 \\ * & 12\ell_2^2 & 0 \\ * & * & 1 \end{bmatrix} + \frac{1}{12}m_{\ell_3} \begin{bmatrix} 2L_3^2 & 2L_3^2 & 0 \\ * & 2L_3^2 & 0 \\ * & * & 0 \end{bmatrix} \\ &= \frac{1}{12}m_{\ell_3} \begin{bmatrix} 12\ell_1^2 + 12\ell_2^2 + 24\ell_1\ell_2 c_2 + 2L_3^2 & 12\ell_2^2 + 12\ell_1\ell_2 c_2 + 2L_3^2 & 0 \\ * & 12\ell_2^2 + 2L_3^2 & 0 \\ * & * & 1 \end{bmatrix} \end{aligned}$$

to sum them up, to obtain $B(\boldsymbol{q})$

$$\begin{aligned}
B(\boldsymbol{q}) &= B_1(\boldsymbol{q}) + B_2(\boldsymbol{q}) + B_3(\boldsymbol{q}) \\
&= \begin{bmatrix} K_1 + (m_{\ell_2} + 2m_{\ell_3})\ell_1\ell_2 c_2 & K_2 + (\frac{1}{2}m_{\ell_2} + m_{\ell_3})\ell_1\ell_2 c_2 & 0 \\ * & K_3 & 0 \\ * & * & m_{\ell_3} \end{bmatrix} \\
K_1 &= m_{\ell_1}(\tfrac{7}{12}\ell_1^2 + \tfrac{1}{4}R_1^2) + m_{\ell_2}(\ell_1^2 + \tfrac{7}{12}\ell_2^2 + \tfrac{1}{4}R_2^2) + m_{\ell_3}(\ell_1^2 + \ell_2^2 + \tfrac{1}{6}L_3^2) \\
K_2 &= m_{\ell_2}(\tfrac{7}{12}\ell_2^2 + \tfrac{1}{4}R_2^2) + m_{\ell_3}(\ell_2^2 + \tfrac{1}{6}L_3^2) \\
K_3 &= m_{\ell_2}(\tfrac{7}{12}\ell_2^2 + \tfrac{1}{4}R_2^2) + m_{\ell_3}(\ell_2^2 + \tfrac{1}{6}L_3^2)
\end{aligned}$$

In the end the kinetic energy is given by:

$$\begin{aligned}
\mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \frac{1}{2}\dot{\boldsymbol{q}}^\top B(\boldsymbol{q})\dot{\boldsymbol{q}} \\
&= \tfrac{1}{2}\big(K_1 + (m_{\ell_2} + 2m_{\ell_3})\ell_1\ell_2 c_2\big)\dot{\theta}_1^2 + \big(K_2 + (\tfrac{1}{2}m_{\ell_2} + m_{\ell_3})\ell_1\ell_2 c_2\big)\dot{\theta}_1\dot{\theta}_2 + \tfrac{1}{2}\big(K_3\big)\dot{\theta}_2^2 + \tfrac{1}{2}\big(m_{\ell_3}\big)\dot{d}_3^2
\end{aligned}$$

3.2 Potential Energy

To obtain

$$\mathcal{U}(\mathbf{q}) = \sum_{i=1}^{n} \mathcal{U}_i(\mathbf{q}) = -\sum_{i=1}^{n} m_{\ell_i}\mathbf{g}_0^\intercal \mathbf{p}_{\ell_i}$$

where $\mathbf{g}_0 = [0, 0, -g]^\intercal$ is the gravity vector with respect to Σ_b, we simply sum the individual contributions $\mathcal{U}_i(\mathbf{q})$ as follows:

$$\begin{aligned}
\mathcal{U}_1(\mathbf{q}) &= m_{\ell_1}\mathbf{g}_0^\intercal \mathbf{p}_{\ell_1} = -m_{\ell_1} g \frac{\ell_1}{2}\sin(\theta_1) \\
\mathcal{U}_2(\mathbf{q}) &= m_{\ell_2}\mathbf{g}_0^\intercal \mathbf{p}_{\ell_2} = -m_{\ell_2} g(\ell_1 \sin(\theta_1) + \frac{\ell_2}{2}\sin(\theta_{12})) \\
\mathcal{U}_3(\mathbf{q}) &= m_{\ell_3}\mathbf{g}_0^\intercal \mathbf{p}_{\ell_3} = -m_{\ell_3} g(\ell_1 \sin(\theta_1) + \ell_2 \sin(\theta_{12}))
\end{aligned}$$

Therefore, the potential energy is given by

$$\mathcal{U}(\boldsymbol{q}) = -gm_3(L0 + q3) - L0gm1 - L0gm_2$$

Chapter 4

Third Assignment

4.1 Equation of motion

The equation of motion of the system is given in the form:

$$B(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) = \boldsymbol{\zeta} \quad \text{with} \quad \boldsymbol{\zeta} = \boldsymbol{\tau} - F_v\dot{\mathbf{q}} - F_s\text{sgn}(\dot{\mathbf{q}}) - J(\mathbf{q})^\intercal \mathbf{h}_e$$

However, for now, we will ignore the friction terms F_v and F_s as well as the external forces $\mathbf{h}_e$ on the end effector. Therefore, we want to find:

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

where $\boldsymbol{\tau}$ is the input and $\mathbf{q}$ is the output of the system. Since $B(\mathbf{q})$ has already been determined, we can easily compute $g(\mathbf{q})$ using the components $g_i(\mathbf{q})$:

$$g_i(\mathbf{q}) = -\sum_{j=1}^{n} m_{\ell_j} \mathbf{g}_0^\intercal J_{P_i}^{\ell_j}$$

Hence,

$$\begin{aligned} g_1(\mathbf{q}) &= 0 \\ g_2(\mathbf{q}) &= 0 \\ g_3(\mathbf{q}) &= -gm_3 \end{aligned}$$

Finally, we have:

$$g(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \\ -gm_3 \end{bmatrix} g$$

For the matrix $C(\mathbf{q}, \dot{\mathbf{q}})$, we need to do some additional work. The elements c_{ij} are defined as:

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

For convenience, let's first differentiate the entire matrix $B(\mathbf{q})$ with respect to each parameter q_i:

$$\frac{\partial B(\mathbf{q})}{\partial q_1} = \begin{bmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix}$$

$$\frac{\partial B(\mathbf{q})}{\partial q_2} = \begin{bmatrix} -(m_{\ell_2} + 2m_{\ell_3})\ell_1\ell_2 s_2 & -(\frac{1}{2}m_{\ell_2} + m_{\ell_3})\ell_1\ell_2 s_2 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix}$$

$$\frac{\partial B(\mathbf{q})}{\partial q_3} = \begin{bmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix}$$

Using these elements, we can calculate all the c_{ij}. Finally, we group these c_{ij} elements into the matrix:

$$C = \begin{bmatrix} -\frac{L1 \cdot L2 \cdot \dot{q}_2 \cdot \sin(q_2) \cdot (m_2 + 2 \cdot m_3)}{2} & -\frac{L1 \cdot L2 \cdot \sin(q_2) \cdot (\dot{q}_1 + \dot{q}_2) \cdot (m_2 + 2 \cdot m_3)}{2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Chapter 5

Assignment 4

5.1 RNE formulation

The Newton-Euler algorithm is a recursive function in the form

$$\tau = NE(q, \dot{q}, \ddot{q}, g_0)$$

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

$$g(q) = NE(q, 0, 0, g_0)$$
$$C(q, \dot{q})\dot{q} = NE(q, \dot{q}, 0, 0)$$
$$B(q) = \begin{bmatrix} B_1(q) & ... & B_n(q) \end{bmatrix} \text{ with } B_i(q) = NE(q, 0, e_i, 0)$$

To observe the differences, we can compare the values of the matrices under the given configuration:

$$q = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ -0.05 \end{bmatrix}, \ \dot{q} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

The matrices B, C and G are:

$$B_{Lag} = \begin{bmatrix} 0.1095 & 0.05534 & 0 \\ 0.05534 & 0.05534 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} B_{RNE} = \begin{bmatrix} 0.1428 & 0.07966 & 0 \\ 0.06166 & 0.06166 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

$$C_{Lag} * dq = \begin{bmatrix} -0.00054 \\ 0 \\ 0 \end{bmatrix} C_{RNE} * dq = \begin{bmatrix} -0.00072 \\ -0.00018 \\ 0 \end{bmatrix}$$

$$G_{Lag} = \begin{bmatrix} 0 \\ 0 \\ 0.981 \end{bmatrix} G_{RNE} = \begin{bmatrix} 0 \\ 0 \\ 0.981 \end{bmatrix}$$

In the end we compare the TAU:

$$TAU_{Lag} = \begin{bmatrix} 0.1095 \cdot ddq1 + 0.05534 \cdot ddq2 - 0.00054 \\ 0.05534 \cdot ddq1 + 0.05534 \cdot ddq2 \\ 0.1 \cdot ddq3 + 0.981 \end{bmatrix}$$

$$TAU_{RNE} = \begin{bmatrix} 0.1428 \cdot ddq1 + 0.07966 \cdot ddq2 - 0.00072 \\ 0.06166 \cdot ddq1 + 0.06166 \cdot ddq2 - 0.00018 \\ 0.1 \cdot ddq3 + 0.981 \end{bmatrix}$$

Chapter 6

Assignment 5

6.1 Dynamic model in operational space

To compute the dynamic model in operational space we use the following relationship for non-redundant manipulator:

$$B_A(x) = J_A^{-T} B J_A^{-1}$$
$$C_A \dot{x} = J_A^{-T} C \dot{q} - B_A \dot{J}_A \dot{q}$$
$$g_A(x) = J_A^{-T} g$$

where

$$\dot{J}_A = \begin{bmatrix} -L_2 s_1 2 - L_1 s_1 & -L_2 s_{12} & 0 \\ L_2 c_1 2 + L_1 c_1 & L_2 c_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The matrices B_A, C_A, g_A are too big to be displayed here, they can be visualized in 'assignmen5.m' file in the 'assignments' folder of the project.

Chapter 7

Assignment 6

The following simulations are using as parameters: $K_D = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}$ $K_P = \begin{bmatrix} 50 \\ 50 \\ 50 \end{bmatrix}$ with a desired position $q_d = \begin{bmatrix} \pi \\ \frac{\pi}{2} \\ -0.1 \end{bmatrix}$

7.1 Design the Joint Space PD control law with gravity compensation

The schema can be seen in Figure 7.1. For this case we use the computed gravity component for compensation $g(q) = \begin{bmatrix} 0 \\ 0 \\ -gm_3 \end{bmatrix}$ which is actually independent from configuration. In Figure 7.2 we can see how the joints evolve to desired configuration.

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

As we can see in Figure 7.3 the joint 3 which is the only affected by gravity overshoot the desired position.

7.1.2 What happens if the gravity term is set constant and equal to $g(q_d)$ within the control law?

As in this case the gravity term is already constant, a test has been made using a wrong value for gravity $g(q_d) = \begin{bmatrix} 0 \\ 0 \\ -gm_3 \cdot 1.5 \end{bmatrix}$

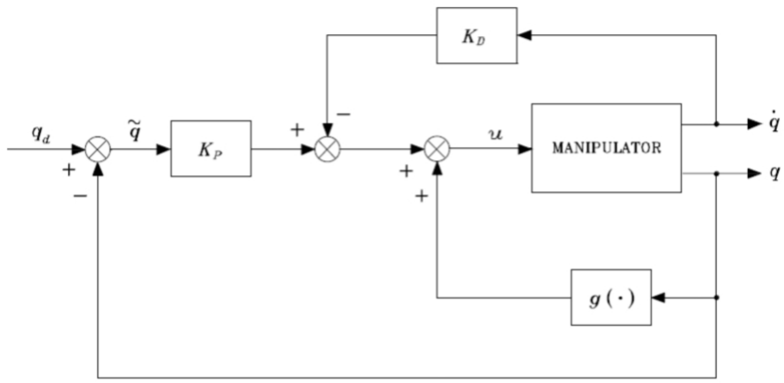


Figure 7.1: Joint space PD with gravity compensation

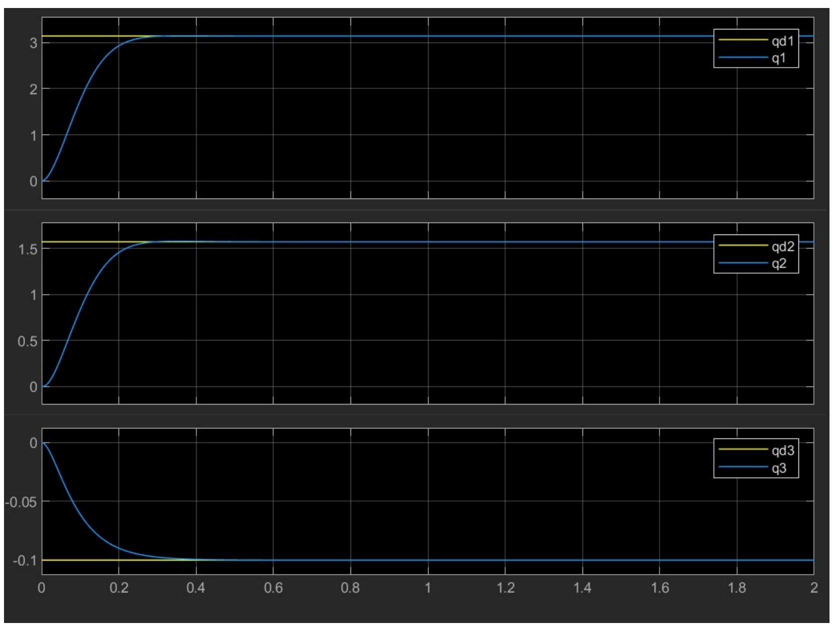


Figure 7.2: Joint space PD: Position values from simulation with gravity compensation

We can see the effect on Figure 7.4 where the control architecture is overcompensating as expected (the steady state position is a bit above the desired one).

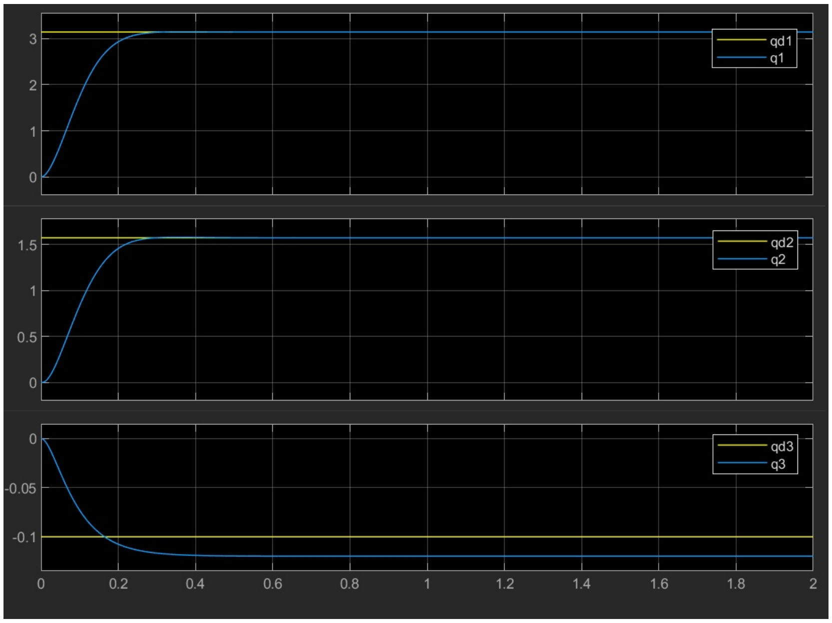


Figure 7.3: Joint space PD: Position values from simulation without gravity compensation

7.1.3 What happens if q_d is not constant (e.g. $q_d(t) = \overline{q}_d + \Delta sin(\omega t)$)?

The sine wave has an amplitude of 0.1 with a frequency of 5 rad/s and the result can be seen in Figure 7.5 where we can notice that the controller is not able to follow the trajectory perfectly.

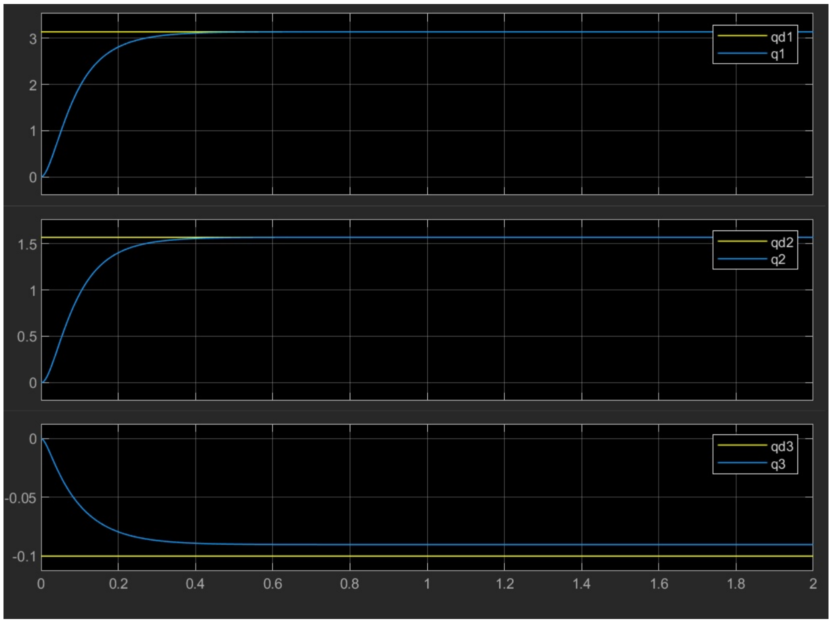


Figure 7.4: Joint space PD: Position values from simulation with wrong gravity compensation

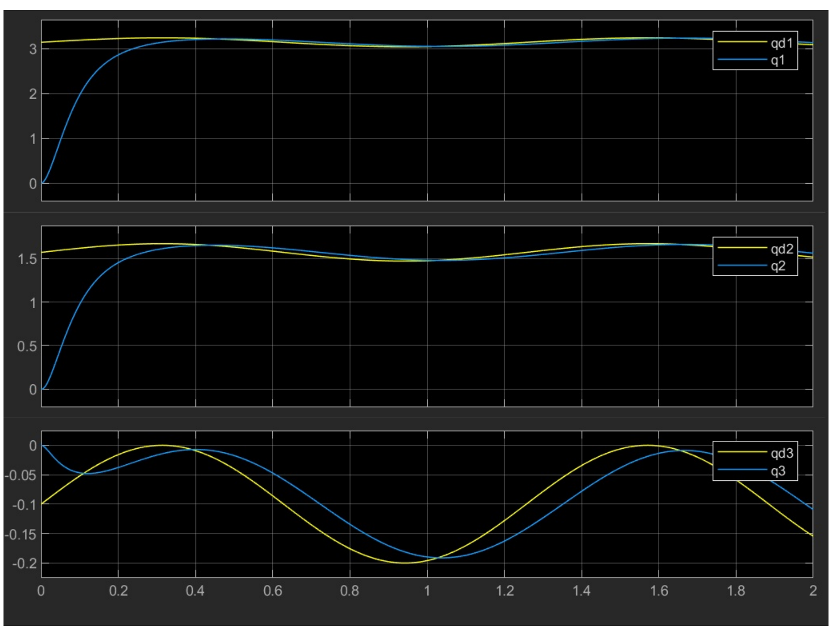


Figure 7.5: Joint space PD: Position values from simulation with gravity compensation and non constant input

Chapter 8

Assignment 7

8.1 Design the Joint Space Inverse Dynamics Control law

The schema can be seen in Figure 8.1. The trajectory is given by a "cubicpolytraj" function by matlab. Wich generates a third-order polynomial that achieves a given set of input waypoints with corresponding time points:

$$q_1 = \begin{bmatrix} \pi \\ \pi \\ \pi \end{bmatrix},\ q_2 = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix},\ q_3 = \begin{bmatrix} -0.2 \\ -0.2 \\ -0.2 \end{bmatrix}.$$

and the control parameters are $K_D = \begin{bmatrix} 20 \\ 100 \\ 20 \end{bmatrix}$, $K_P = \begin{bmatrix} 80 \\ 1000 \\ 100 \end{bmatrix}$. With those the tracking is very good as we can see in Figure 8.1.

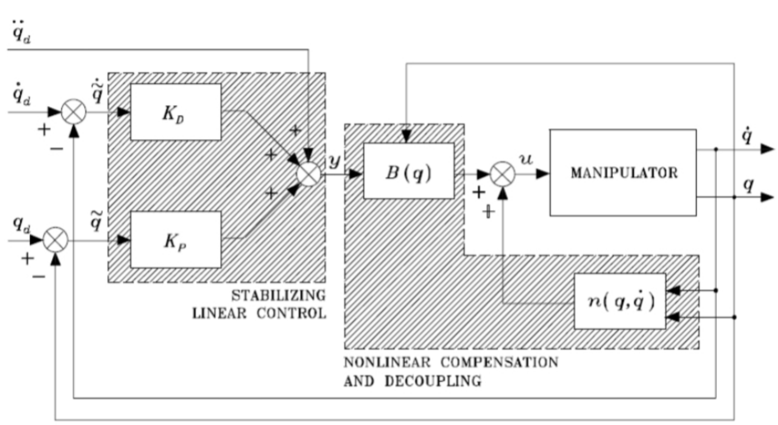


Figure 8.1: Joint space Inverse Dynamics Control law

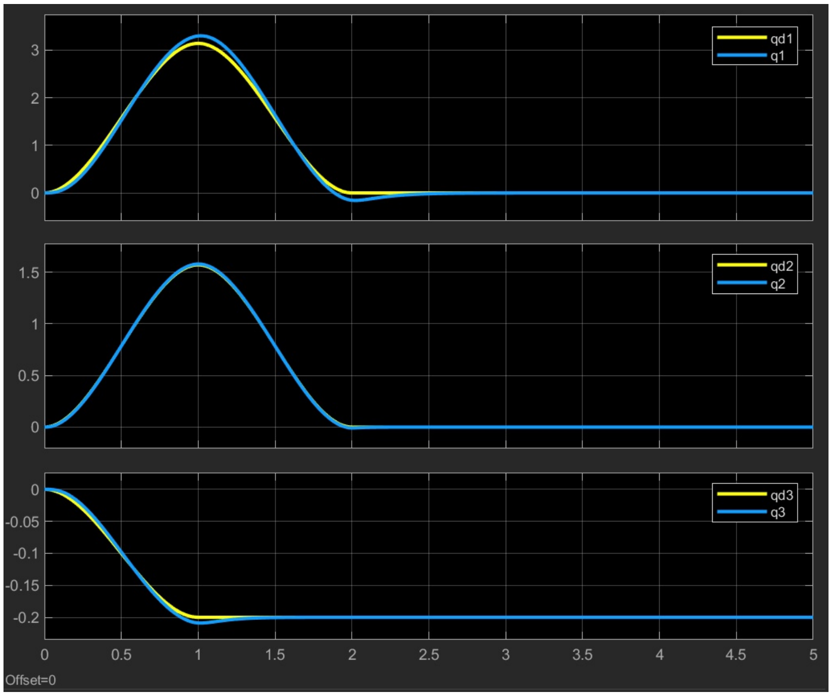


Figure 8.2: Joint space Inverse Dynamics: Position values from simulation

8.1.1 Check that in the nominal case the dynamic behaviour is equivalent to the one of a set of stabilized double integrators

For this scenario the following values have been used: $K_D = \begin{bmatrix} 100 \\ 20 \\ 100 \end{bmatrix}$, $K_P = \begin{bmatrix} 80 \\ 1000 \\ 100 \end{bmatrix}$ and the result is shown in Figure 8.2.

8.1.2 Check the behavior of the control law when the $\hat{B}, \hat{C}, \hat{g}$ used within the controller are different than the "true ones" B;C; g (e.g. slightly modify the masses, the frictions, ...)

The joint with worst result is the one affected by gravity as we can see in Figure 8.3

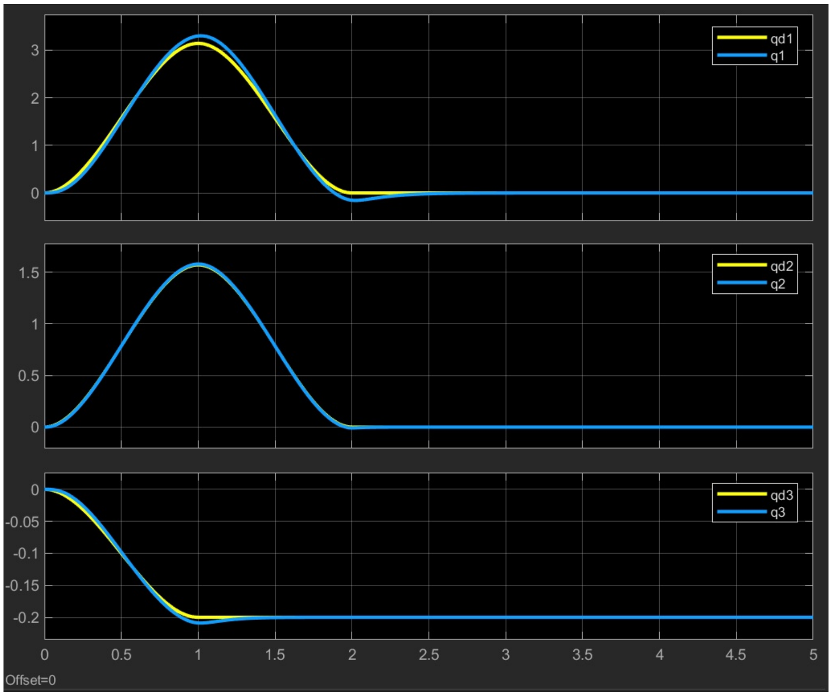


Figure 8.3: Joint space Inverse Dynamics: Position values from simulation with wrong model (different B,C,g)

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

For having a small settling time high gains are needed, which will result in peaks in torques as we can see in picture Figure 8.4.

The gains used are: $K_D = \begin{bmatrix} 50 \\ 40 \\ 40 \end{bmatrix}$, $K_P = \begin{bmatrix} 1500 \\ 1000 \\ 1000 \end{bmatrix}$

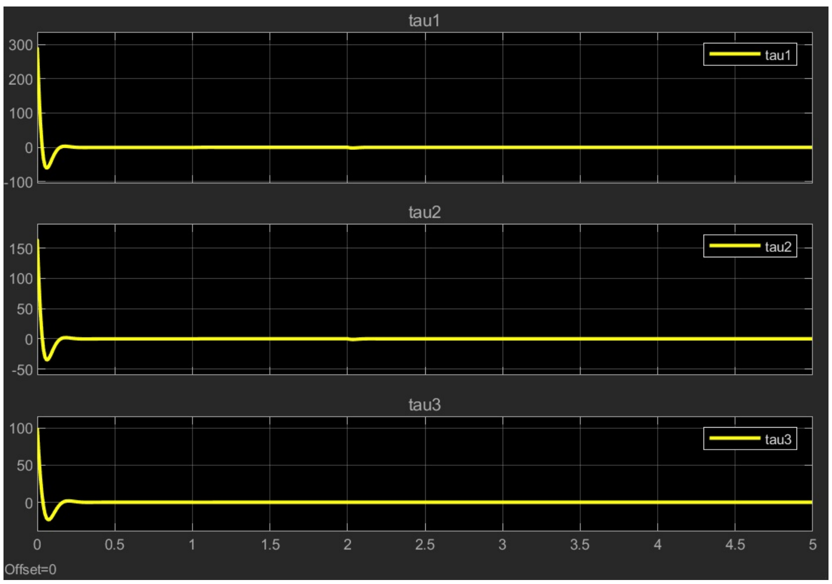


Figure 8.4: Joint space Inverse Dynamics: Torques with high gains

Chapter 9

Assignment 8

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

The schema can be seen in Figure 9.1.

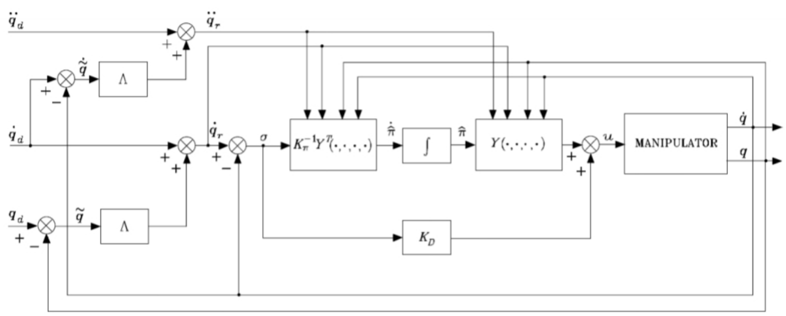


Figure 9.1: Adaptive Control law for the a 1-DoF link under gravity

Dynamic parameters: $I = 0.3, F = 0.1, G = 2.943$;
Initial estimate: $\hat{I} = 0.29, \hat{F} = 0.09, \hat{G} = 0.2933$;
Trajectory: $q_d(t) = Asin(\omega t), \ddot{q}_d(t) = square\_wave(\pm A)$
Control parameters: $K_D = 40, \lambda = 300, K_\theta^- 1 = \begin{bmatrix} 10000 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$
Linear parameterization: $\tau = Y(q, \dot{q}_r, \ddot{q}_r)\theta$
with $\tau = Y(q, \dot{q}_r, \ddot{q}_r) = \begin{bmatrix} \ddot{q}_r & \dot{q}_r & sin(q) \end{bmatrix}$ and $\theta = \begin{bmatrix} I \\ F \\ G \end{bmatrix}$

The tracking of the trajectory is well performing as we can see in Figure 9.2, and by feeding the integrator with the initial estimations as described before we can see in Figure 9.3 that the values oscillates back and forth around the true value. Moreover the high value in K_θ keeps the value closer, otherwise the amplitude of the oscillation would be much higher.

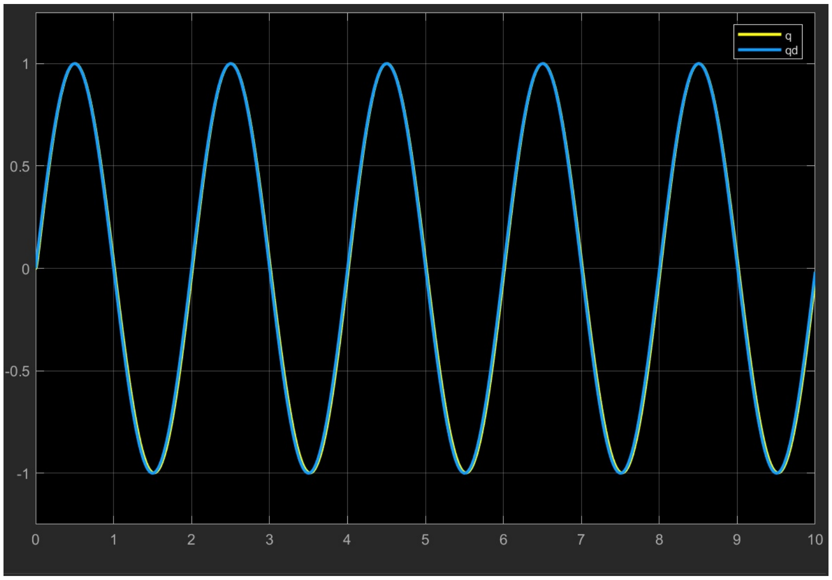


Figure 9.2: Adaptive control: position

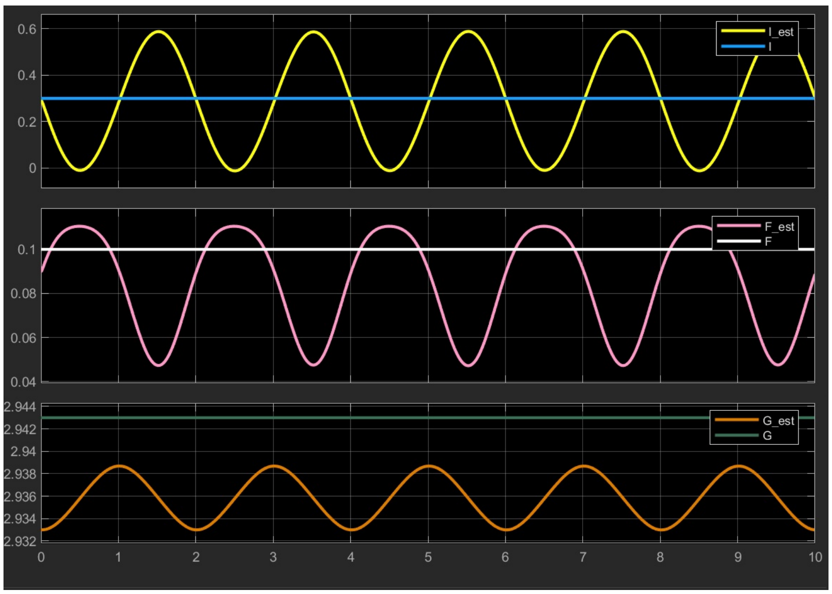


Figure 9.3: Adaptive control: dynamic parameters evolution

Chapter 10

Assignment 9

10.1 Design the Operational Space PD control law with gravity compensation

The schema can be seen in Figure 10.1. This time the gains refers to x,y,z directions not on joints and are defined as: $K_D = \begin{bmatrix} 80 \\ 80 \\ 80 \end{bmatrix}$, $K_P = \begin{bmatrix} 1000 \\ 1000 \\ 1000 \end{bmatrix}$

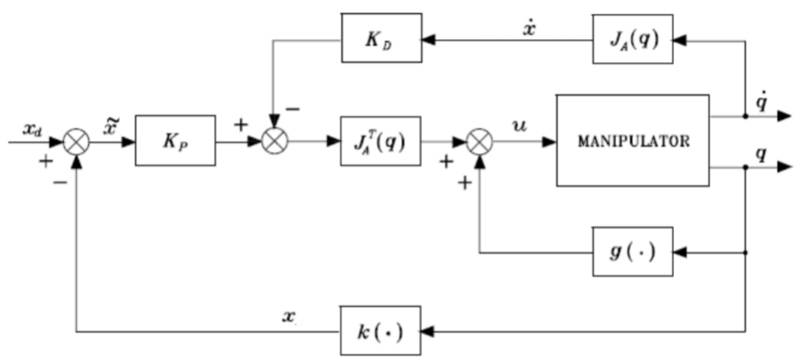


Figure 10.1: Operational space PD with gravity compensation

The gravity compensation case is reported in Figure 10.2. The other cases resembles the joint space cases: without gravity compensation (or with a wrong g model) the z direction has a steady state error and with a sin wave added to input the tracking is not perfect Figure 10.3.

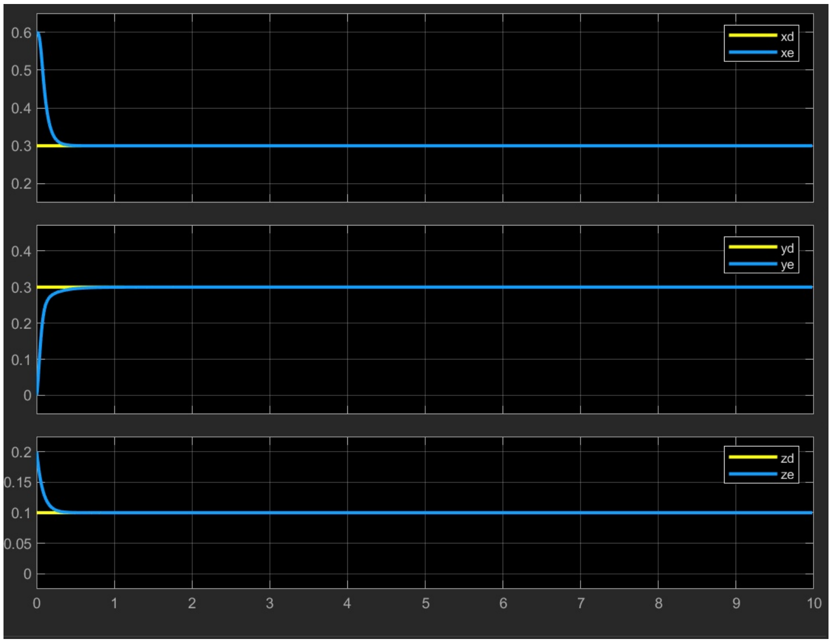


Figure 10.2: Operational space PD: positions

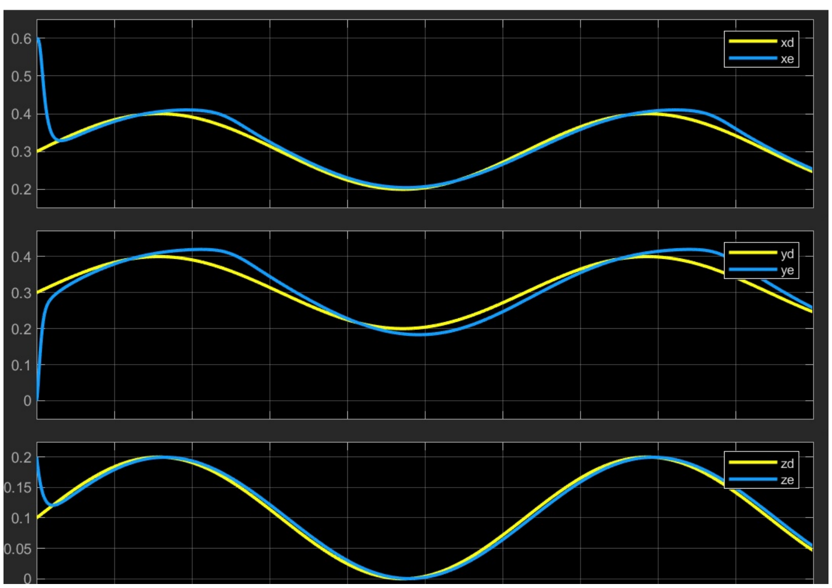


Figure 10.3: Operational space PD: positions

Chapter 11

Assignment 10

11.0.1 Design the Operational Space Inverse Dynamics Control law

The schema can be seen in Figure 11.1. The trajectory is shown in Figure 11.2. The gains used are: $K_D = \begin{bmatrix} 250 \\ 150 \\ 250 \end{bmatrix}$, $K_P = \begin{bmatrix} 1000 \\ 500 \\ 500 \end{bmatrix}$ It's worth noticing that as the gains are referring to operational space, the joint torques are still low as we can see in Figure 11.3.

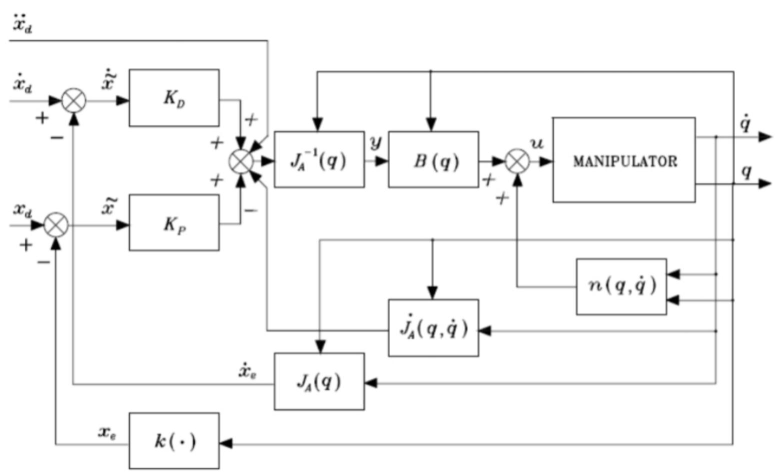


Figure 11.1: Operational Space Inverse Dynamics Control law

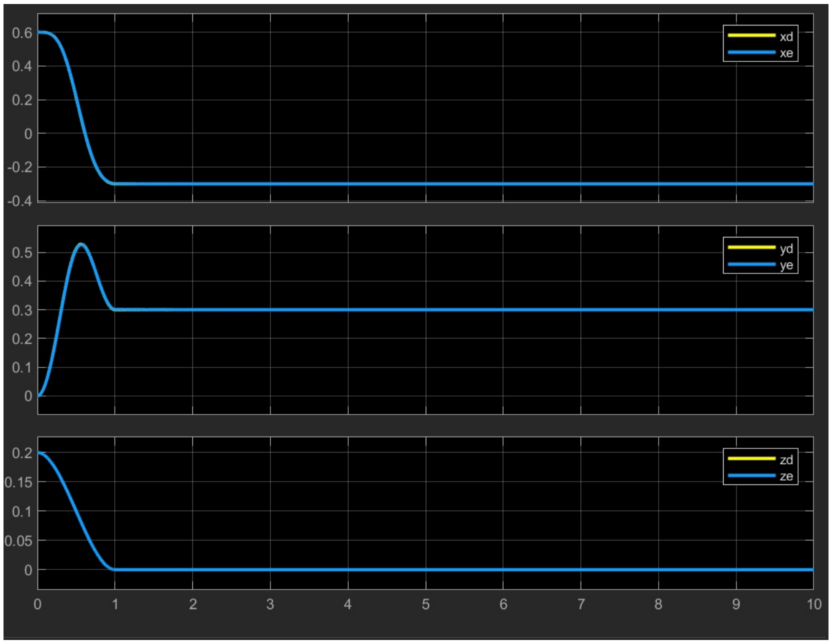


Figure 11.2: Operational Space Inverse Dynamics: positions

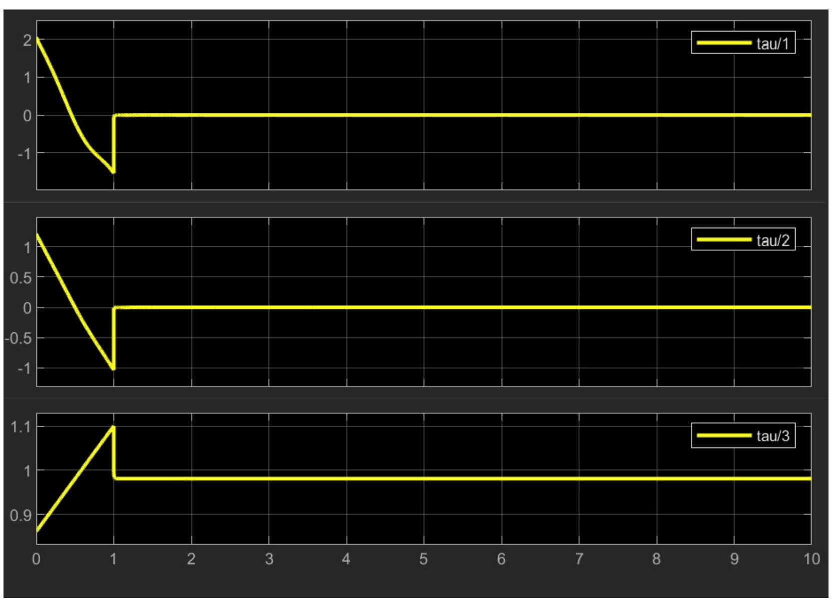


Figure 11.3: Operational Space Inverse Dynamics: velocities

Chapter 12

Assignment 11

12.0.1 Study the compliance control

The environment is simulated as a plane parallel to xy at $z = 0.25$. The robot end effector start at $z = 0.4$ and its goal is to get to $z = 0.2$, so it must go down and collide with the environment. The gain for the z direction is $K_P = 100$.

In the following sections the different scenarios are presented for different values of stiffness K of the environment.

12.0.2 Compliance Control: $K << K_P$

In this case we have $K = 1$ and as we can see in Figure 12.1 the desired position is almost reached.

12.0.3 Compliance Control: $K >> K_P$

In this case we have $K = 10000$ and as we can see in Figure 12.2 the end effector stops when get in contact with the environment and cannot move forward.

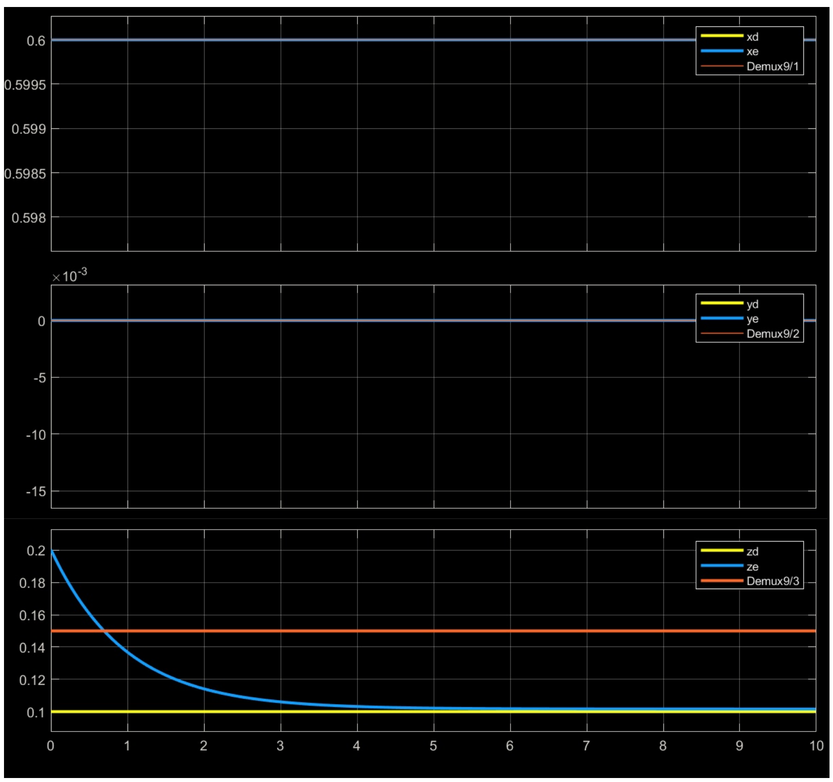


Figure 12.1: Compliance Control: $K << K_P$

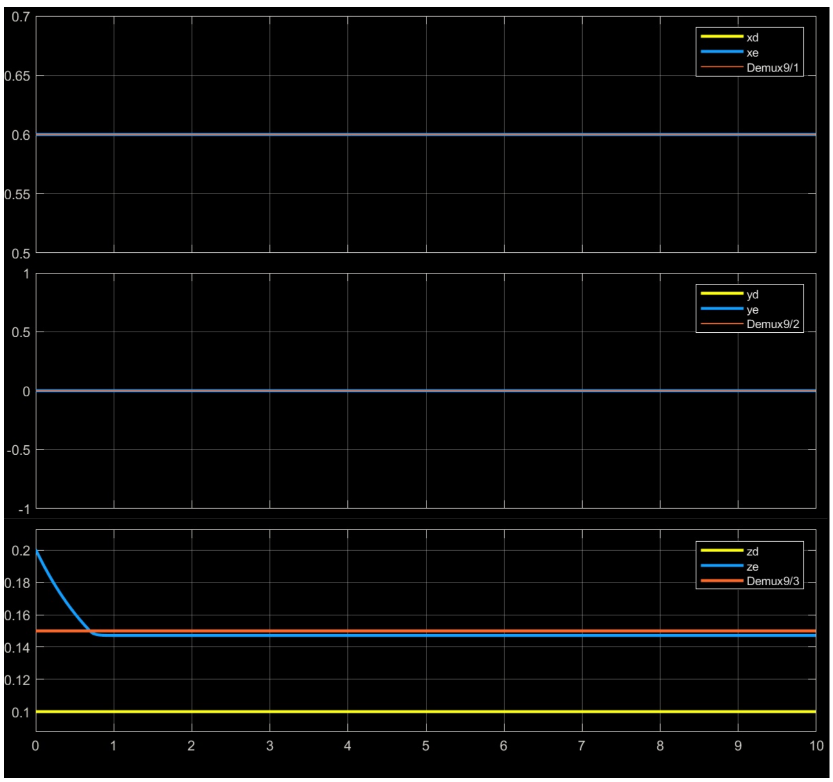


Figure 12.2: Compliance Control: $K >> K_P$

12.0.4 Compliance Control: $K == K_P$

In this case we have $K = 100$ and as we can see in Figure 12.3 the end effector penetrates the environment while getting some resistance and it stops half way through the desired position.

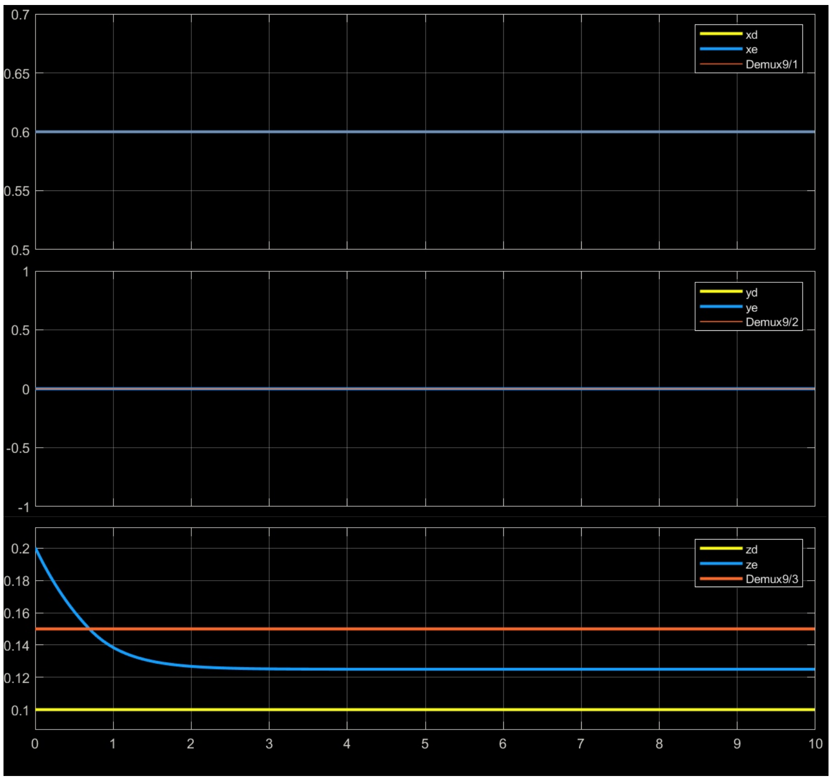


Figure 12.3: Compliance Control: $K == K_P$

Chapter 13

Assignment 12

13.0.1 Implement the impedance control in the operational space

The schema can be seen in Figure 13.1.

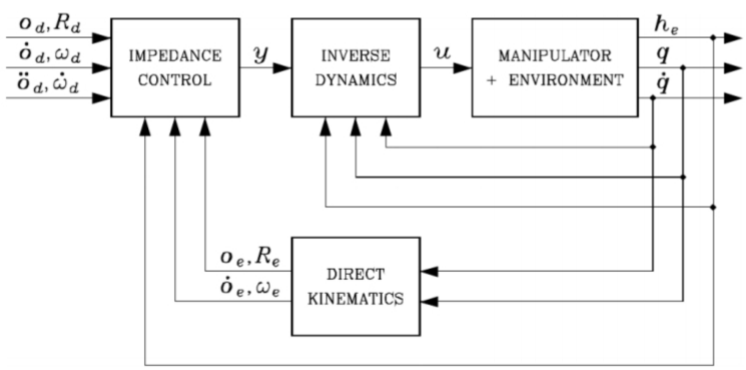


Figure 13.1: Impedance Control

The environment is described as before but this time on the z direction we have $K_P = 50$ and $K = 10$. Moreover now we have a double S trajectory with the same final value as before and we can see in Figure 13.2 that the presence of the stiffness of the environment prevents the robot to reach the target.

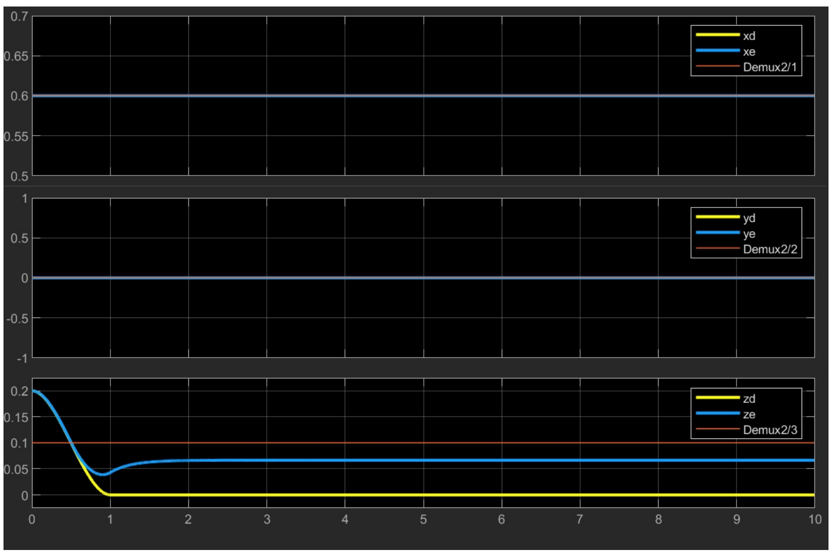


Figure 13.2: Impedance Control: Trajectory

Chapter 14

Assignment 13

14.0.1 Implement the admittance control in the operational space

The schema can be seen in Figure 14.1.

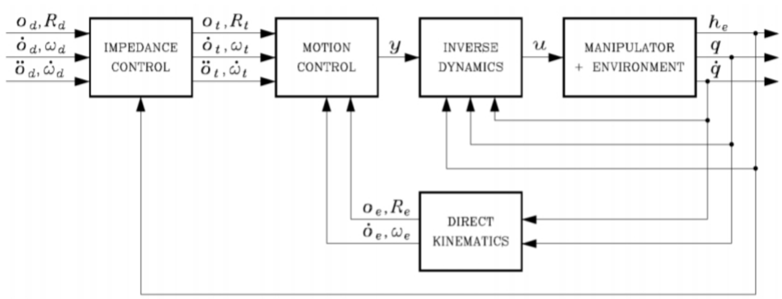


Figure 14.1: Admittance Control

We have same situation as with the impedance about the position of the environment and the trajectory used. In the admittance model we have an admittance block that generates a new reference trajectory x_t based on the desired trajectory and the contact force with the environment. This new trajectory is then fed to a motion control. We can see in Figure 14.2 that again after the contact the trajectory in z direction is following x_t and no longer x_d. Also here the ratio between K_P and K is 50 to 10.

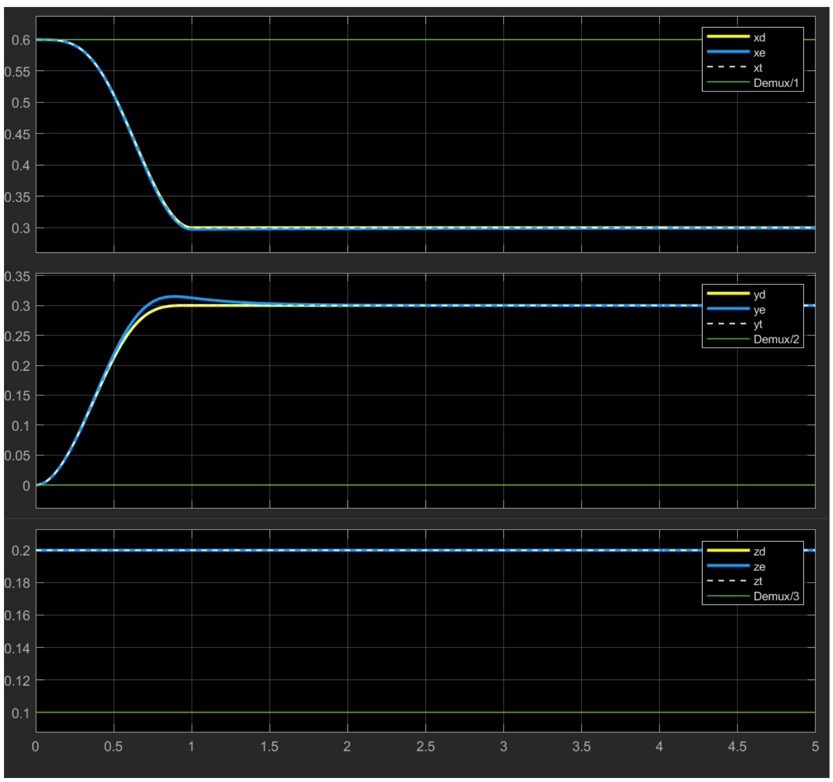


Figure 14.2: Admittance Control: Trajectory

Chapter 15

Assignment 14

15.0.1 Implement the Force Control with Inner Position Loop

The simplified version of the schema can be seen in Figure 15.1.

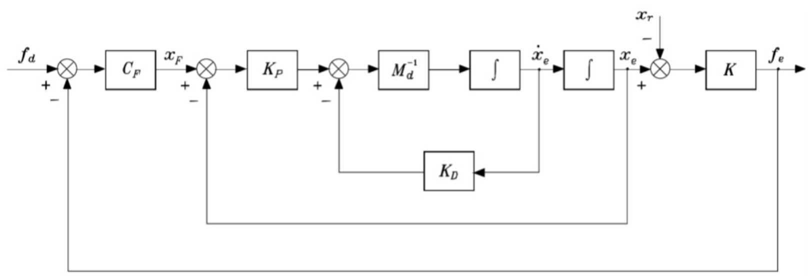


Figure 15.1: Force control

The situation of environment is the same as described before. As the controller is only in force we will show only the z direction.

Using only a proportional force controller we have a steady state error as we can see in Figure 15.2. The gain value is $K_F = 5$.

To overcome this gap a PI controller is used and we reach zero steady error as we can see in Figure 15.3. The proportional gain is still the same while the integral is $K_I = 4$.

The steady state position is reached at $Kx_e = Kx_r + f_d$ and it can be seen in Figure 15.4.

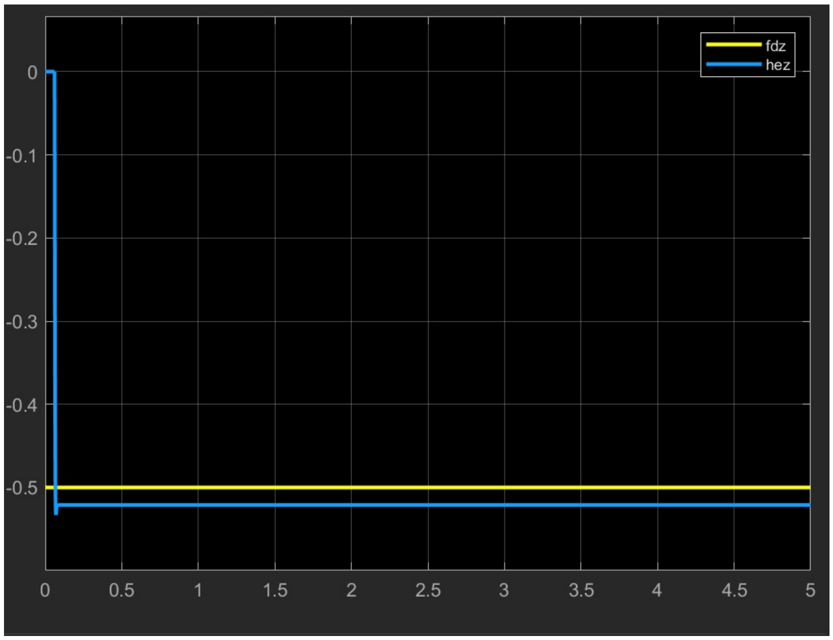


Figure 15.2: Force Control: $C_F = K_F$

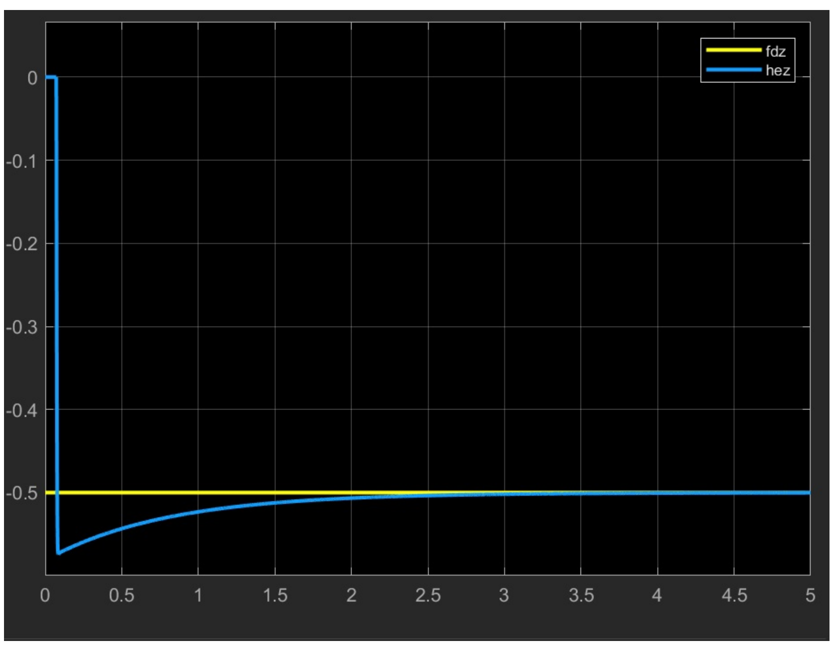


Figure 15.3: Force Control: $C_F = K_F + \frac{K_I}{s}$

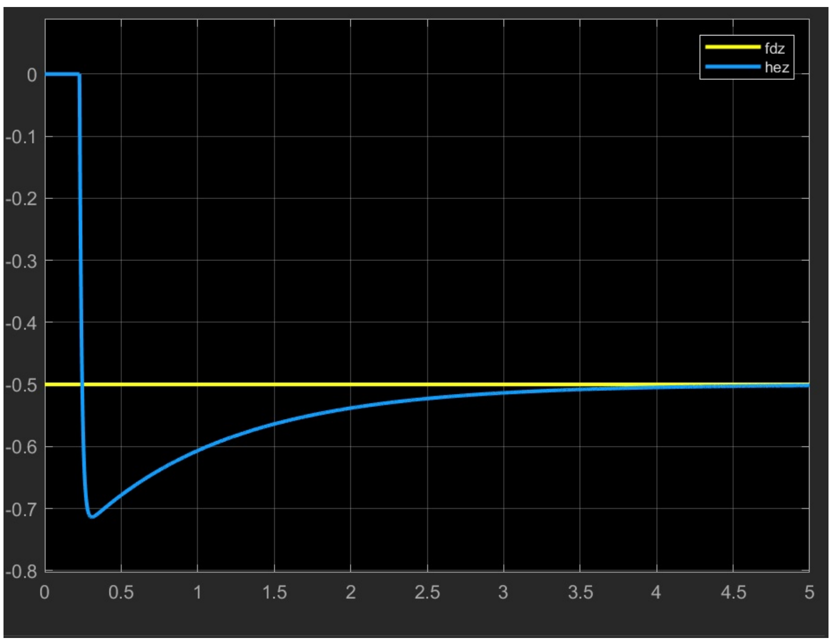


Figure 15.4: Force Control: equilibrium position

Chapter 16

Assignment 15

16.0.1 Implement the Parallel Force/Position Control

The simplified version of the schema can be seen in Figure 16.1.

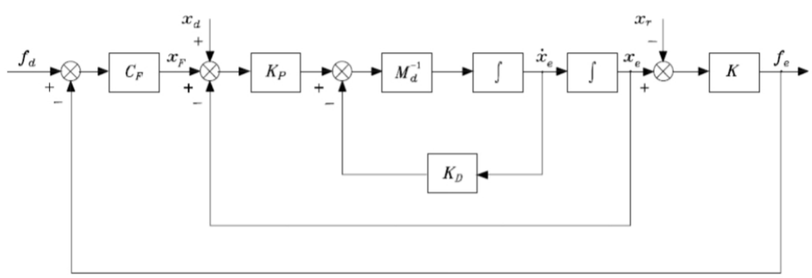


Figure 16.1: Parallel Force/Position Control

In this scenario, the objective remains unchanged: controlling the robot's force in the z direction while maintaining positional control in the x and y directions.

The observation of Figure 16.2 illustrates the desired outcome, wherein the robot successfully achieves the desired positions for x and y. However, due to the rigidity of the environment, the robot struggles to attain the desired position in the z direction.

As for pure force control we can also see that the force is reaching the desired force target in z direction (Figure 16.3) using a PI controller while it reaches a steady state error using only a P controller (Figure 16.3).

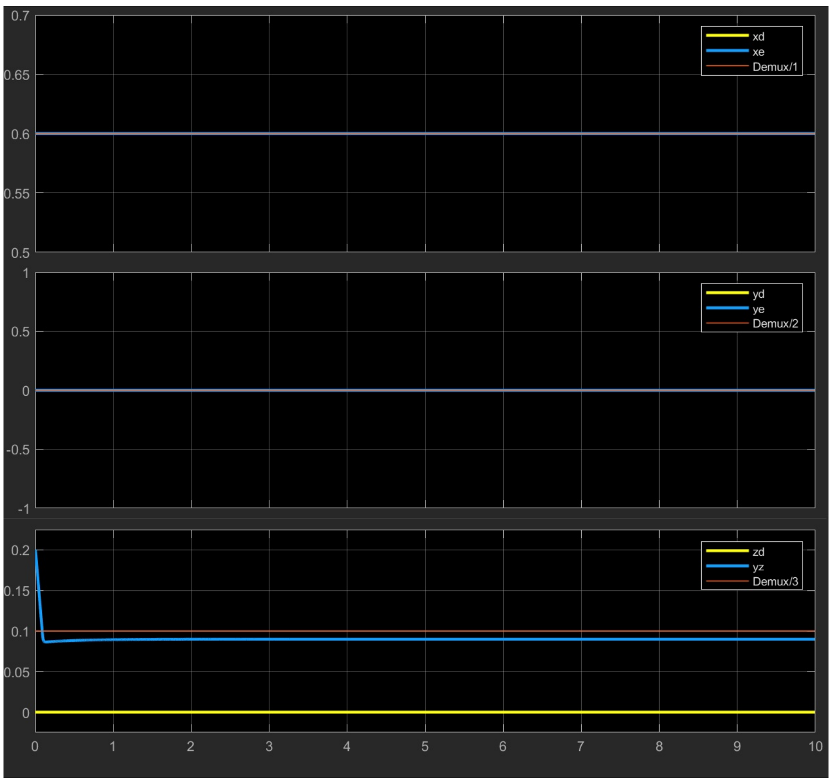


Figure 16.2: Parallel Force/Position Control: positions

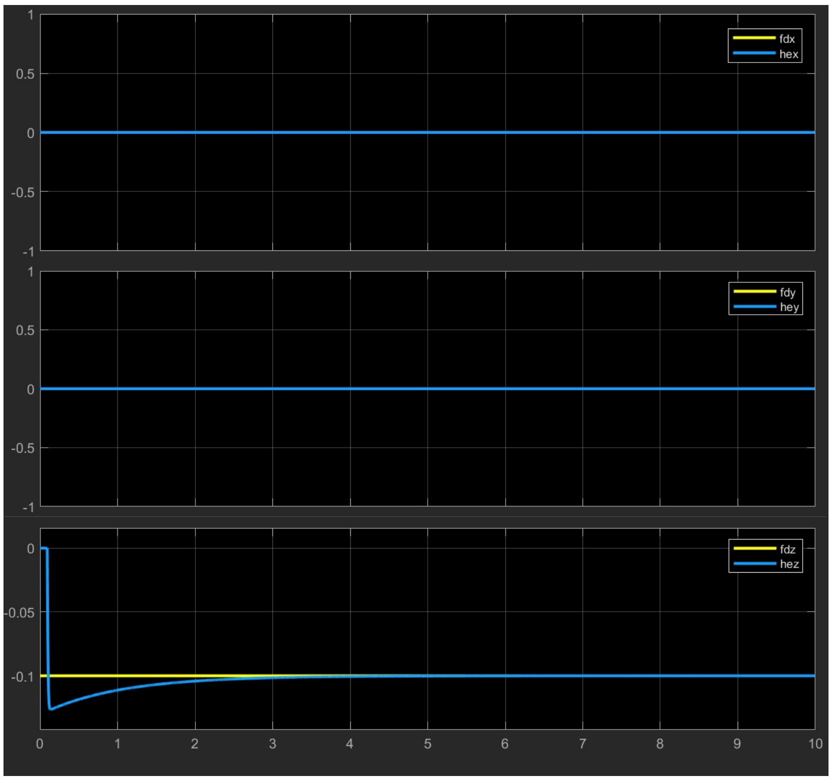


Figure 16.3: Parallel Force/Position Control: forces with PI controller

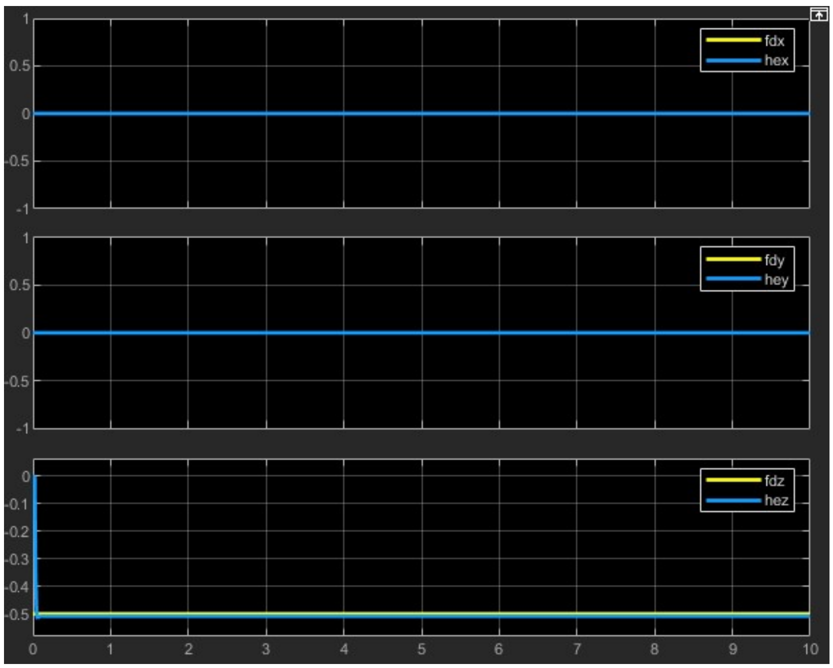


Figure 16.4: Parallel Force/Position Control: forces with P controller