Long Assignment 1

Sayan Basu Roy (sayan@iiitd.ac.in)

November 5, 2020

Question 1.

Consider the SCARA robot illustrated in Figure 1. The generalized coordinates of the system are denoted as $q = [q_1\ q_2\ q_3\ q_4]^T$. The final structure of the above SCARA robot dynamics will be as follows

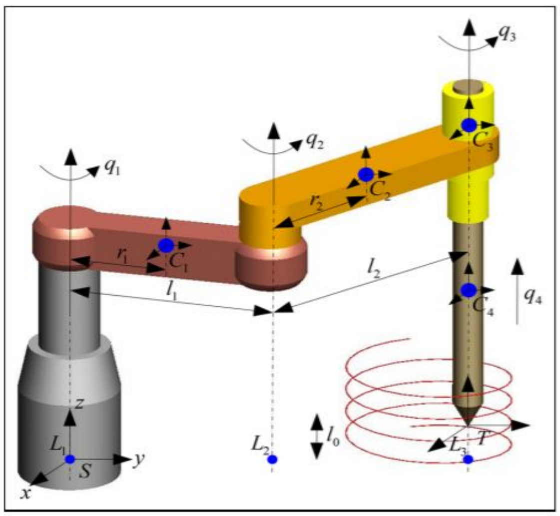

Figure 1: SCARA robot in its reference configuration.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q,\dot{q}) = \tau \tag{1}$$

Where the inertia matrix $M(q) \in \mathbb{R}^{4\times4}$ for the SCARA robot is

$$M(q) = \begin{bmatrix} \alpha+\beta+2\gamma cosq_2 & \beta+\gamma cosq_2 & \delta & 0 \\ \beta+\gamma cosq_2 & \beta & \delta & 0 \\ \delta & \delta & \delta & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix}$$

where

$\alpha = I_{z1} + r_1^2 m_1 + l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4$
$\beta = I_{z1} + I_{z2} + I_{z3} + I_{z4} + l_2^2 m_3 + l_2^2 m_4 + m_2 r_2^2$
$\gamma = l_1 l_2 m_3 + l_1 l_2 m_4 + l_1 m_2 r_2$
$\delta = I_{z3} + I_{z4}$

The Coriolis matrix $C(q,\dot{q}) \in \mathbb{R}^{4\times4}$ for the SCARA robot is

$$C(q,\dot{q}) = \begin{bmatrix} -\gamma sinq_2\dot{q}_2 & -\gamma sinq_2(\dot{q}_1+\dot{q}_2) & 0 & 0 \\ -\gamma sinq_2\dot{q}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The gravitational forces $G(q,\dot{q}) \in \mathbb{R}^{4\times 1}$ on the SCARA robot is expressed as

$$G(q,\dot{q}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_4 g \end{bmatrix}$$

where $\tau \in \mathbb{R}^{4\times 1}$ is the joint torque vector.

Consider the following the proportional derivative (PD) control input

$$\tau(t) = -k_1 e(t) - k_2 \dot{e}(t) \tag{2}$$

where $e(t) \triangleq q(t) - q_d(t)$ is tracking error, $q_d(t)$ is the desired trajectory, and $k_1 > 0$ and $k_2 > 0$ are controller gains. In many real world applications, instead of continuous time trajectories you will be provided with the way-points and their respective time stamps and you need to design a smooth trajectory that passes through all the waypoints complying with the time constraints. In the "Waypoints.xlsx", the desired position of the robot (end-effector) is given in the form of such waypoints. These waypoints are chosen by uniformly sampling a spiral trajectory as shown in Figure 1. You have to generate a smooth trajectory [1] $q_d(t) = [q_{1d}(t), q_{2d}(t), q_{3d}(t), q_{4d}(t)]^T$ from the given end effector position waypoints. Assume that $q_{3d}(t) = 0$, $\forall t \geq 0$ since it is independent of end-effector's position. (Hint : You have to apply inverse kinematics based on the geometry of the robot to get the corresponding waypoints in the joint-space and then generate the trajectory. While computing the joint-space waypoints, you have to eliminate the possibility of multiple solution!)

Consider the initial conditions as $q(0) = [0,0,0,0]^T$ and $\dot{q}(0) = [0,0,0,0]^T$, assuming $t = 0$ as the initial time and choose k_1 and k_2 such that the error converges close to zero.

The parameter values for simulation are as follows: $m_1 = 20kg$, $m_2 = m_3 = 15kg$, $m_4 = 0.5kg$, $l_0 = 0m$, $l_1 = 0.2m$, $l_2 = 0.25m$, $r_1 = 0.1m$, $r_2 = 0.125m$, $I_{z1} = 0.27kg-m^2$, $I_{z2} = 0.31kg-m^2$, $I_{z3} = 0.02kg-m^2$, and $I_{z4} = 0.0001kg-m^2$

Please note that q_4 represents the position of the end effector from a point that is at l_0 height with respect to the base frame.

What you have to submit :

1> Simulink file / matlab code for the implementation of the dynamics, desired trajectory generator and controller.

2> Plot of time evolution of the following variables for both the cases during the time-span $t \in [0,20]$.

a> e-vs-t,

b> $\dot{e}$-vs-t

c> τ-vs-t.

d> q_d-vs-t.

e> "$x - vs - y - vs - z$" 3D plot of the end-effector actual trajectory (Generate it from your code using forward kinematics. If your code is correct the actual trajectory should be close to the supplied waypoints!).

4> Prepare a report which includes all the relevant derivations (inverse and forward kinematics etc.) and results (above mentioned plots).

[1] Refer to "Trajectories" in the chapter "Robot Arm Kinematics" of peter Corke's book