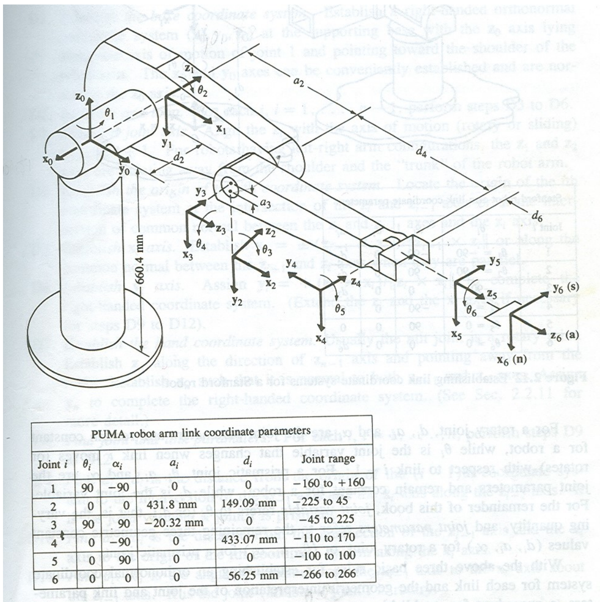
MECH564:
Fundamentals of Robot Mechanics and Controls
Final Project

1 Objective

Evaluate the task level robot control method for the first three joints of PUMA 560 robot (shown in Fig. 1) arm by computer simulations.



PUMA robot arm link coordinate parameters					
Joint i	θ_i	α_i	a_i	d_i	Joint range
1	90	−90	0	0	−160 to +160
2	0	0	431.8 mm	149.09 mm	−225 to 45
3	90	90	−20.32 mm	0	−45 to 225
4	0	−90	0	433.07 mm	−110 to 170
5	0	90	0	0	−100 to 100
6	0	0	0	56.25 mm	−266 to 266

Figure 1: Schematic and DH table for PUMA 560 robot

2 Dynamics Model of the Robot

The dynamics model of the first four links of PUMA 560 is

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

where

$$D(q) = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix}; \quad C(q,\dot{q}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}; \quad g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \quad q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}; \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

where the $(k,j)^{th}$ element of $C(q,\dot{q})$ is

$$c_{kj} = \sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i$$

The numeric forms of d_{kj} $(kg-m^2)$ are as follows:

$$
\begin{aligned}
d_{11} &= 2.4574+1.7181c_2c_2+0.4430s_{23}s_{23}-0.0324c_2c_{23}-0.0415c_{23}s_{23}+0.9378c_2s_{23}\\
d_{12} &= 2.2312s_2-0.0068s_{23}-0.1634c_{23}\\
d_{13} &= -0.0068s_{23}-0.1634c_{23}\\
\\
d_{21} &= d_{12}\\
d_{22} &= 5.1285+0.9378s_3-0.0324c_3\\
d_{23} &= 0.4424+0.4689s_3-0.0162c_3\\
\\
d_{31} &= d_{13}\\
d_{32} &= d_{23}\\
d_{33} &= 1.0236
\end{aligned}
$$

The numeric forms of c_{ijk} $(kg-m^2)$ are as follows:

$$
\begin{aligned}
c_{111} &= 0\\
c_{121} &= 0.0207-1.2752c_2s_2+0.4429c_3s_3-0.8859s_2s_3s_{23}\\
&\qquad +0.0325c_2s_{23}+0.4689c_2c_{23}-0.4689s_2s_{23}\\
&\qquad -0.0461c_{22}-0.0415c_{23}c_{23}-0.0163s_3\\
c_{131} &= 0.0207+0.4429c_2s_2+0.4429c_3s_3-0.8859s_2s_3s_{23}\\
&\qquad +0.0163c_2s_{23}+0.4689c_2c_{23}-0.0415c_{23}c_{23}\\
c_{211} &= c_{121}\\
c_{221} &= 1.8181c_2+0.1634s_{23}-0.0068c_{23}\\
c_{231} &= 0.1634s_{23}-0.0068c_{23}\\
c_{311} &= c_{131}\\
c_{321} &= c_{231}\\
c_{331} &= 0.1634s_{23}-0.0068c_{23}\\
\\
c_{112} &= -c_{121}\\
c_{122} &= 0\\
c_{132} &= 0\\
c_{212} &= c_{122}\\
c_{222} &= 0\\
c_{232} &= 0.4689c_3+0.0162s_3\\
c_{312} &= 0\\
c_{322} &= c_{232}\\
c_{332} &= 0.4689c_3+0.0162s_3\\
\\
c_{113} &= -c_{131}\\
c_{123} &= -c_{132}\\
c_{133} &= 0\\
c_{213} &= c_{123}\\
c_{223} &= -c_{232}\\
c_{233} &= 0\\
c_{313} &= c_{133}\\
c_{323} &= c_{233}\\
c_{333} &= 0
\end{aligned}
$$

The numeric forms of g_k $(kg - m^2 - sec^{-2})$ are as follows:

$$\begin{aligned} g_1 &= 0 \\ g_2 &= -48.5564c_2 + 1.0462s_2 + 0.3683c_{23} - 10.6528s_{23} \\ g_3 &= 0.3683c_{23} - 10.6528s_{23} \end{aligned}$$

The forward kinematics model of the first four links of PUMA 560

$$Y = h(q) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} h_1(q) \\ h_2(q) \\ h_3(q) \end{bmatrix}$$

where the robot task space consists of (x, y, z), and

$$h(q) = \begin{bmatrix} a_3c_1c_{23} + d_4c_1s_{23} + a_2c_1c_2 - d_2s_1 \\ a_3s_1c_{23} + d_4s_1s_{23} + a_2s_1c_2 + d_2c_1 \\ -a_3s_{23} + d_4c_{23} - a_2s_2 \end{bmatrix}$$

The link parameters in the above formula can be found in the table shown in Fig. 1.

3 The Task Level Control Method

The task level controller is given as

$$\tau = D(q)J_h(q)^{-1}(u - \dot{J}_h(q)\dot{q}) + C(q, \dot{q})\dot{q} + g(q)$$

where $u = \ddot{Y}^d + K_v(\dot{Y}^d - \dot{Y}) + K_p(Y^d - Y)$ and

$$J_h(q) = \begin{bmatrix} \frac{\partial h_1}{\partial q_1} & \frac{\partial h_1}{\partial q_2} & \frac{\partial h_1}{\partial q_3} \\ \frac{\partial h_2}{\partial q_1} & \frac{\partial h_2}{\partial q_2} & \frac{\partial h_2}{\partial q_3} \\ \frac{\partial h_3}{\partial q_1} & \frac{\partial h_3}{\partial q_2} & \frac{\partial h_3}{\partial q_3} \end{bmatrix}; \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}.$$

4 Required Tasks

4.1 Design of the Task Level Controller

(1) Derive the nonlinear controller to linearize and decouple the robot dynamics model in the task space.

(2) Write the linearized robot dynamics model in the state space form; use any design method that you are familiar with to design the linear feedback controller.

4.2 The Simulation

(1) Develop a software to simulate the robot dynamics model, i.e., numerically solve

$$\begin{cases} D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \\ Y = h(q) \end{cases}$$

You can write your own code to solve the ODE (e.g., Runge-Kutta 4th order method which can be found in "An Introduction to Numerical Analysis", Kendall E. Atkinson or other numerical analysis books). You can also directly use ODE solvers in Matlab (e.g., ode45, ode23), but then you need to rewrite the dynamics equation into the correct form that can be used by ODE solvers. With the implementation, test your dynamics model with an input $\tau = [1, 0, 0]^T$, plot the output $q_1(t)$, $q_2(t)$, and $q_3(t)$ in a single figure. Also plot the position x, y, and z in a single figure. Explain the plots you have generated.

(2) Numerically implement the nonlinear controller

$$\tau = D(q)J_h(q)^{-1}(u - \dot{J}_h(q)\dot{q}) + C(q,\dot{q})\dot{q} + g(q)$$

Plot x, y, z for the following three different inputs.

$$u = [1,0,0]^T, \quad u = [0,1,0]^T, \quad u = [0,0,1]^T$$

Explain the plots you have generated.

(3) Apply the linear controller in order to track a given desired trajectory as follows:

$$Y_d(t) = \begin{bmatrix} -0.866R\cos(\omega t) - 0.56 \\ R\sin(\omega t) \\ 0.5R\cos(\omega t) - 0.08 \end{bmatrix}$$

where $R = 0.25m$. Please plot the position error $Y_d - Y$ (one figure with three curves) and velocity error $\dot{Y}_d - \dot{Y}$ (one figure with three curves) for each of the following four cases:
1): $\omega = \pi/4$, $\dot{Y}(0) = 0$, and $Y(0) = [-0.7765m, 0.0m, 0.045m]^T$;
2): $\omega = \pi/4$, $\dot{Y}(0) = 0$, and $Y(0) = [-0.5m, -0.1m, 0.0m]^T$;
3): $\omega = \pi/2$, $\dot{Y}(0) = 0$, and $Y(0) = [-0.7765m, 0.0m, 0.045m]^T$
4): $\omega = \pi/2$, $\dot{Y}(0) = 0$, and $Y(0) = [-0.5m, -0.1m, 0.0m]^T$.
Explain your results.

4.3 Discussions

Please discuss the following questions based on the above simulation results.

(1) Explain the linearization and decoupling. How do you verify them?

(2) What kind of effects do the robot's initial position and the velocity of the desired trajectory have on the tracking errors ?

(3) Explain the effects of modeling errors on the robot tracking control.

5 Project Report

(1) The report must be typed, and turned in before the due date.

(2) In your report all tasks with proper explanations and discussions must be documented.

(3) The derivation and design steps have to be presented and clearly explained.

(4) The block diagram of all computer programs must be included in the report. All source codes of the computer programs must be clearly documented and submitted as Appendix at the end of the report.

(5) All plots and diagrams must have proper captions and be clearly explained.

(6) References must be provided in the report for any methods used in the project from sources other than the textbook and class notes.

6 Project Submission

All the documents including report and source codes should be submitted in a single Zip file on Canvas. Please name the zip file as FirstName_LastName.zip (e.g., Jianguo_Zhao.zip).