3.2 a)

Let $\theta_1(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$

$\theta_2(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3$

$t_0 = 0, \quad t_f = 10 \text{ sec}, \quad \theta_1(t_0) = \pi, \quad \theta_1(t_f) = 0, \quad \dot{\theta}_1(t_0) = \dot{\theta}_1(t_f) = 0$

$\theta_2(t_0) = \pi/2, \quad \theta_2(t_f) = 0, \quad \dot{\theta}_2(t_0) = \dot{\theta}_2(t_f) = 0$

```
clc;clear;close all;
syms theta1 theta2 theta1_dot theta2_dot a0 a1 a2 a3 b0 b1 b2 b3 t
%% 3.2(a) - Cubic Polynomial Tragectory
q1 = a0+ a1*t + a2*t^2 + a3*t^3; % Cubic Polynomial for 1st joint

q2 = b0+ b1*t + b2*t^2 + b3*t^3; % Cubic Polynomial for 2nd joint

q_dot = jacobian([q1,q2],t);      % Time Derivative

t0=0; tf =10; %Initial and final times

eq1 = subs(q1,[t],[t0]); %a0+ a1*t0 + a2*t0^2 + a3*t0^3
eq2 = subs(q1,[t],[tf]); %a0+ a1*tf + a2*tf^2 + a3*tf^3
eq3 = subs(q_dot(1),[t],[t0]);
eq4 = subs(q_dot(1),[t],[tf]);
eq5 = subs(q2,[t],[t0]); %b0+ b1*t0 + b2*t0^2 + b3*t0^3
eq6 = subs(q2,[t],[tf]); %b0+ b1*tf + b2*tf^2 + b3*tf^3
eq7 = subs(q_dot(2),[t],[t0]);
eq8 = subs(q_dot(2),[t],[tf]);

sol = solve([eq1==pi,eq2==0,eq3==0,eq4==0,eq5==pi/2,eq6==0,eq7==0,eq8==0],[a0,a1,a2,a3,b0,b1,b2,b3]);

cubic_1 = subs(q1,[a0,a1,a2,a3],[sol.a0,sol.a1,sol.a2,sol.a3]);
cubic_2 = subs(q2,[b0,b1,b2,b3],[sol.b0,sol.b1,sol.b2,sol.b3]);
disp("Cubic Polynomial for 1st Joint")
disp(cubic_1)
disp("Cubic Polynomial for 2nd Joint")
disp(cubic_2)
qd = [cubic_1;cubic_2];
qd_dot = jacobian(qd,t);
qd_ddot = jacobian(qd_dot,t);
```

```
Cubic Polynomial for 1st Joint
(pi*t^3)/500 - (3*pi*t^2)/100 + pi

Cubic Polynomial for 2nd Joint
(pi*t^3)/1000 - (3*pi*t^2)/200 + pi/2
```

Matlab script for calculating cubic polynomials

$$\theta_1(t) = \frac{\pi t^3}{500} - \frac{3\pi t^2}{100} + \pi$$

$$\theta_2(t) = \frac{\pi t^3}{1000} - \frac{3\pi t^2}{200} + \pi/2$$

3.2 b)

Equations of motions

$$\ddot{\theta}_2\left[m_2 r_2^2 + m_2 r_2 l_1 \cos\theta_2 + I_2\right] - \dot{\theta}_2\left[l_1 m_2 r_2 \dot{\theta}_1 \sin\theta_2 + l_1 m_2 r_2 \sin\theta_2 (\dot{\theta}_1 + \dot{\theta}_2)\right] + \ddot{\theta}_1\left[m_2 l_1^2 + 2 m_2 \cos\theta_2\, l_1 r_2 + m_1 r_1^2 + m_2 r_2^2 + I_1 + I_2\right] - g l_1 m_2 \sin\theta_1 - g m_1 r_1 \sin\theta_1 - m_2 g r_2 \sin(\theta_1 + \theta_2) = u_1$$

$$\ddot{\theta}_2\left[m_2 r_2^2 + I_2\right] + \ddot{\theta}_1\left[m_2 r_2^2 + m_2 l_1 r_2 \cos\theta_2 + I_2\right] - g m_2 r_2 \sin(\theta_1 + \theta_2) + l_1 m_2 r_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \sin\theta_2 - l_1 m_2 r_2 \dot{\theta}_1 \dot{\theta}_2 \sin\theta_2 = u_2$$

Manipulator equation form:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \qquad \text{where: } \tau = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$M = \begin{bmatrix} m_2 l_1^2 + 2 m_2 \cos\theta_2\, l_2 r_2 + m_1 r_1^2 + m_2 r_2^2 + I_1 + I_2 & m_2 r_2^2 + m_2 r_2 l_1 \cos\theta_2 + I_2 \\ m_2 r_2^2 + m_2 l_1 r_2 \cos\theta_2 + I_2 & m_2 r_2^2 + I_2 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & l_1 m_2 r_2 \dot{\theta}_1 \sin\theta_2 + l_1 m_2 r_2 \sin\theta_2 (\dot{\theta}_1 + \dot{\theta}_2) \\ l_1 m_2 r_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin\theta_2 - l_1 m_2 r_2 \dot{\theta}_2 \sin\theta_2 & 0 \end{bmatrix}$$

$$g = \begin{bmatrix} -g l_1 m_2 \sin\theta_1 - g m_1 r_1 \sin\theta_1 - m_2 g r_2 \sin(\theta_1 + \theta_2) \\ -g m_2 r_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

3.2 c)

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

$$M(q)\ddot{q} = \tau - C(q,\dot{q})\dot{q} - g(q) \quad \text{— ①}$$

Let us assume $\tau = M(q)v + C(q,\dot{q})\dot{q} + g(q)$, and substitute this in equation ①

we get $\boxed{\ddot{q} = v} \rightarrow$ new system

state vector $z = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$

state space $\rightarrow \dot{z} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} z_2 \\ v \end{bmatrix}$

$$\dot{z} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} v \quad \text{where } I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Control law for new system:

virtual control input $\boxed{v = -K \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}}$

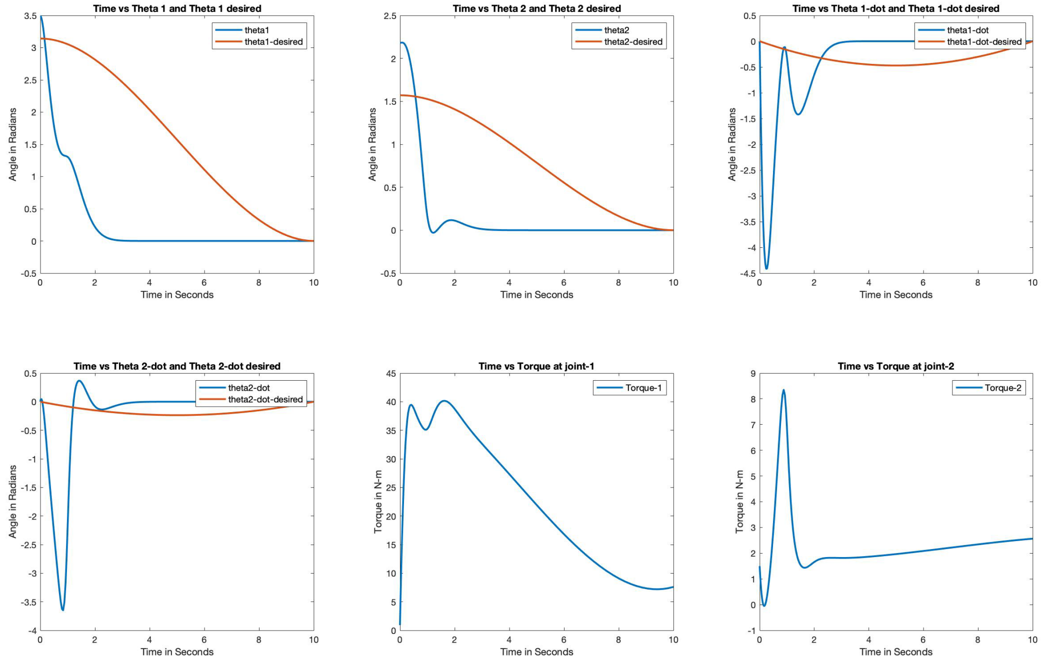
$\tau = M(q)v + C(q,\dot{q})\dot{q} + g(q)$ becomes

$$\tau = M(q)\left(-K\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right) + C(q,\dot{q})\dot{q} + g(q)$$

(or)

Final control law

$$\boxed{\tau = M(q)\left(-K\begin{bmatrix} q \\ \dot{q} \end{bmatrix}\right) + C(q,\dot{q})\dot{q} + g(q)}$$



System performance with feedback linearization alone

Eigen values are placed for virtual control input at : $[-3-2i, -3+2i, -3, -2.5]$

Gain is : $K = \begin{bmatrix} 8.4365 & 4.7619 & 5.7267 & 1.9061 \\ 5.9615 & 8.1920 & -1.9986 & 5.7733 \end{bmatrix}$

Note: To test this in matlab, open the file named "Tekumatla_ProgAssignment_3.m", comment out line 169 and uncomment line 168, and then run.

3.2 d) and 3.2 e)

To track a trajectory, we can not use the same virtual control input as above. virtual control input need to be changed to:

$$v = -K\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} z_{1d} \\ z_{2d} \end{bmatrix}\right) + \ddot{z}_d$$

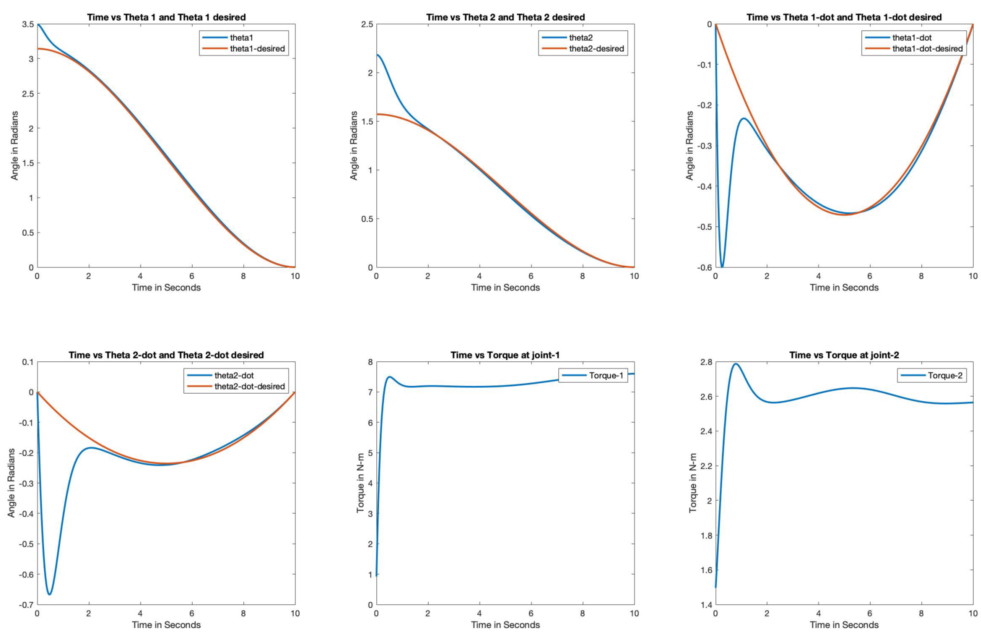
(or)

$$v = -K\left(\begin{bmatrix} q \\ \dot{q} \end{bmatrix} - \begin{bmatrix} q_d \\ \dot{q}_d \end{bmatrix}\right) + \ddot{q}_d$$ where q_d is desired trajectory

control law to track the Trajectory is:

$$\tau = M(q)\left(-K\left(\begin{bmatrix} q \\ \dot{q} \end{bmatrix} - \begin{bmatrix} q_d \\ \dot{q}_d \end{bmatrix}\right) + \ddot{q}_d\right) + C(q,\dot{q})\dot{q} + g(q)$$

Starting points: $\theta_1(0) = 200^\circ$, $\theta_2(0) = 125^\circ$, $\dot{\theta}_1 = 0$, $\dot{\theta}_2 = 0$

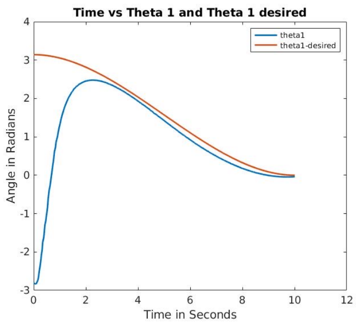


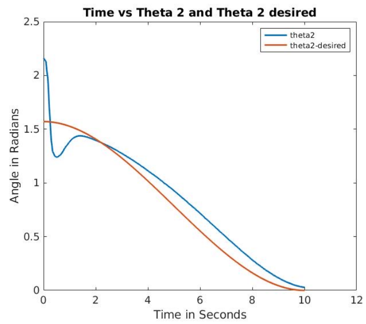
System performance with Trajectory tracking

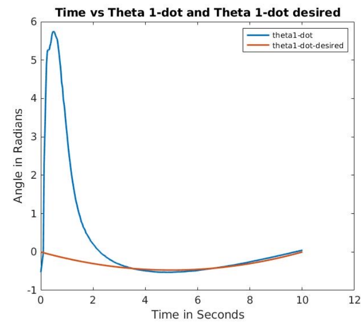
Note: To test this in matlab, open the file named "Tekumatla_ProgAssignment_3.m", comment out line 168 and uncomment line 169, and then run.

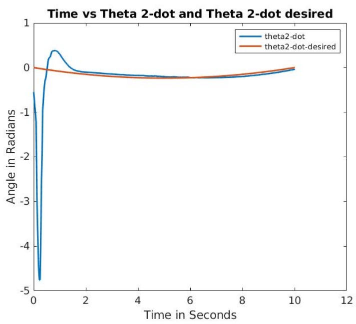
3.2 f)

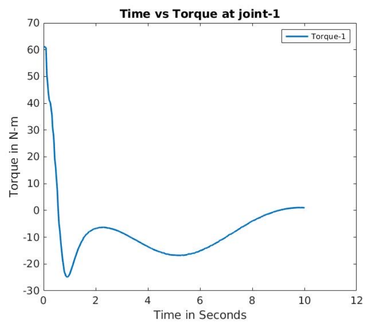
Plots of trajectories and control inputs from gazebo

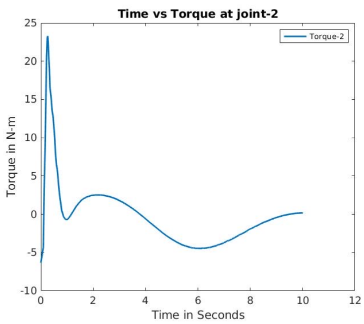












Neither the trajectories nor the control inputs are matching with those obtained in step e. This is because in the simulation, we are dealing with a physical model that has some properties such as friction which are ignored while modelling the dynamics.