MECH 6324 – Robot Control
Final Report

Submitted by: Abishek Chandrasekhar **Date: 11/30/16**
Net ID : axc157330

Problem 1:

1. **Dynamic equations of motion for a Planar Elbow manipulator: (question 1 and 2 of Problem 1)**
   a) **Function file from MATLAB**:

```
%%%Abishek Chandrasekhar
%%%Robot Control_Assignment #6
%%%Problem 1_1
%%function EOM
%%%Defining the state variables
%%% [x(1) x(2) x(3) x(4)] =  [q1 q1dot q2 q2dot]

function [xdot] = EOM(t,x)

l1 = 1; l2 = 1; m1 = 2; m2 = 1; lc1 = .5; lc2 = .5; I1 = .5; I2 =
.25; g =9.81; %%parameters
xdot = zeros (4,1); %%%%initializing xdot

%%Inertia Matrix D terms
d11 = m1*lc1^2 + m2*(l1^2 + lc2^2 + 2*l1*lc2^2 + 2*l1*lc2*cos(x(3)))
+ I1 + I2;
d12 = m2*(lc2^2 + l1*lc2*cos(x(3))) + I2;
d21 = d12;
d22 = m2*lc2^2 + I2;

D = [d11 d12; d21 d22];

%%Centrifugal and Coriolis C terms
h = -(m2*l1*lc2*sin(x(3)));
c111 = 0; c121 = h; c211 = h; c221 = h; c112 = -h; c122 = 0; c212 =
0; c222 = 0;

C = [(h*x(4)) (h*(x(4) + x(2))) ; (-h*x(1)) 0];

%%Gravity terms
phi1 = (m1*lc1 + m2*l1)*g*cos(x(1)) + m2*lc2*g*cos(x(1)+x(3));
phi2 = m2*lc2*g*cos(x(1)+x(3));


%%Acceleration vector
a1 = - c121*x(2)*x(4) - c211*x(4)*x(2) - c221*x(4)^2 - phi1;
a2 = - c112*x(2)^2 - phi2;
Dinv = inv(D);
Atemp = [a1; a2];
A  = Dinv*Atemp;
q1ddot = A(1);
q2ddot = A(2);
```

```
%%%Required vector xdot
%xdot = [q1dot q1ddot q2dot q2ddot]

xdot(1) = x(2);
xdot(2) = q1ddot;
xdot(3) = x(4);
xdot(4) = q2ddot;
```

b) Main program that uses ode 45: (question 3 of Problem 1)

```
%%%%Abishek Chandrasekhar
%%Robot Control_HW_#6
%%Problem 1_2
%%Main program

%%Time
tspan = [0 40];
x0 = [-1.5, 0, 0.1, 0]'; %%initial conditions

[t,x] = ode45('EOM', tspan, x0);
```

2. Plot of Link Responses: (question 4 of Problem 1)

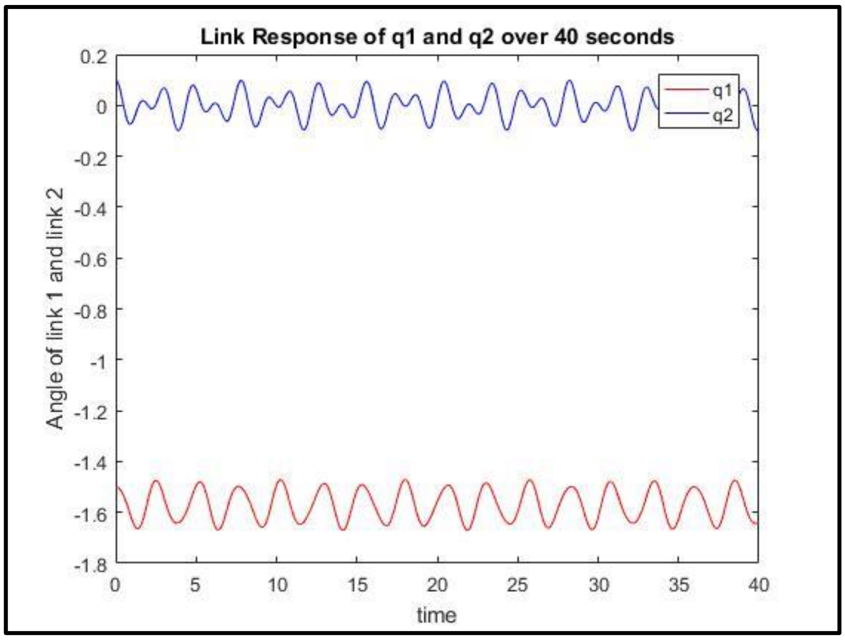

Figure 1 - Link responses of q1 and q2

The above **Figure 1** shows how the angles q1 and q2 change over time. The time span for the ode solver 45 is given as 40 seconds. The red plot represents the response of **'q1'** and the blue plot represents the response of **'q2'.**

3. Phase space trajectories: (question 5 of Problem 1)

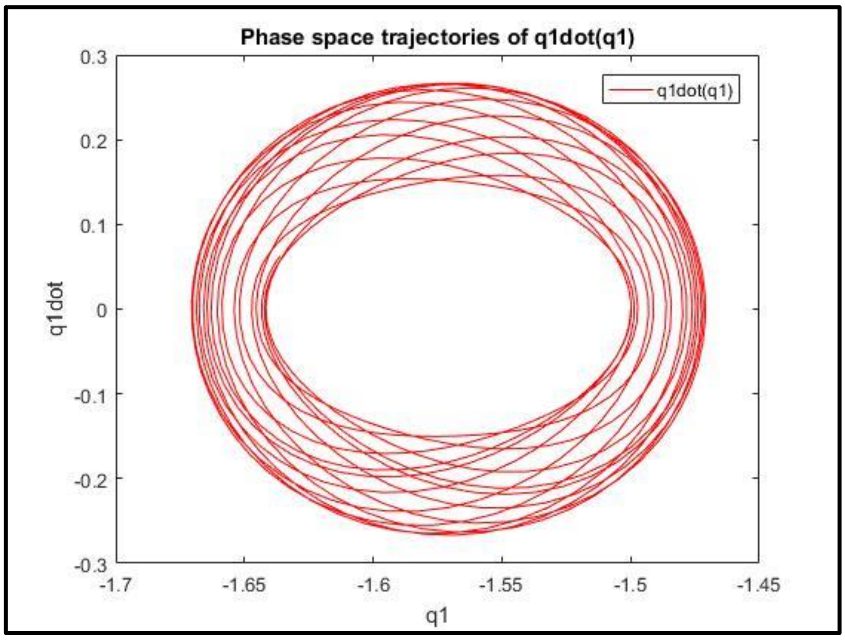

Figure 2 - q1dot(q1)

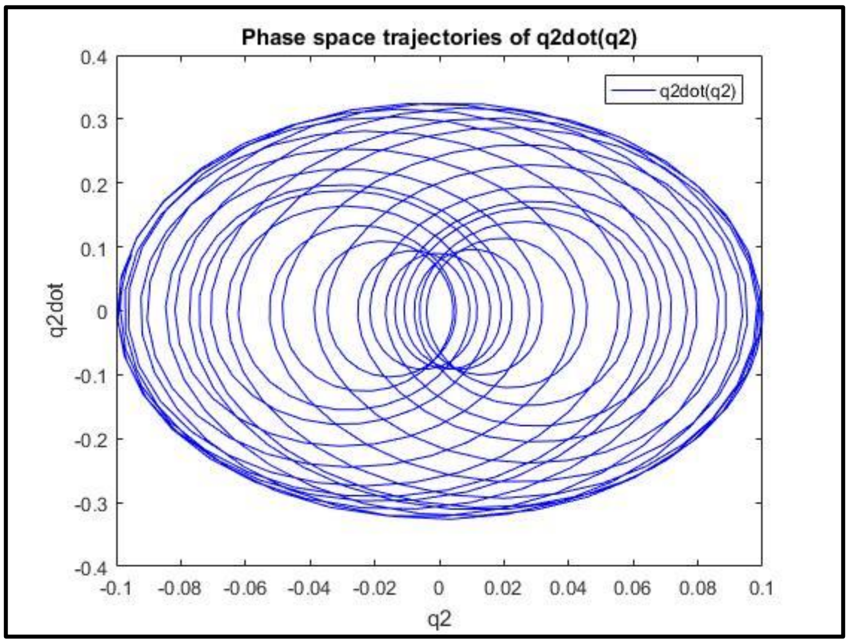

Figure 3 - q2dot(q2)

Figure 2 and **Figure 3** show the phase space trajectories of q1dot(q1) and q2dot(q2) respectively.

The supporting code for creating the above plots is given below:

(code for question 4 and 5)

```
tspan = [0 40]; %%Time
x0 = [-1.5, 0, 0.1, 0]'; %%initial conditions
[t,x] = ode45('EOM', tspan, x0);
figure
plot(t,x(:,1),'R') %%%% q1 response
hold on
plot(t,x(:,3),'B') %%%% q2 response
title('Link Response of q1 and q2 over 40 seconds');
xlabel('time');
ylabel('Angle of link 1 and link 2');
legend('q1', 'q2')

figure
plot(x(:,1),x(:,2),'R') %%%% q1dot(q1)
title('Phase space trajectories of q1dot(q1)');
xlabel('q1');
ylabel('q1dot');
legend('q1dot(q1)');

figure
plot(x(:,3),x(:,4),'B') %%%% q2dot(q2)
title('Phase space trajectories of q2dot(q2)');
xlabel('q2');
ylabel('q2dot');
legend('q2dot(q2)');
```

4. **The initial conditions are changed: (question 6 of Problem 1)**
   (Supporting code)

```
tspan = [0 40]; %%Time
x0 = [1, 0, 0, 0]'; %%initial conditions
[t,x] = ode45('EOM', tspan, x0);
```

5. **Link responses of q1 and q2 under new conditions: (question 7 of Problem 1)**

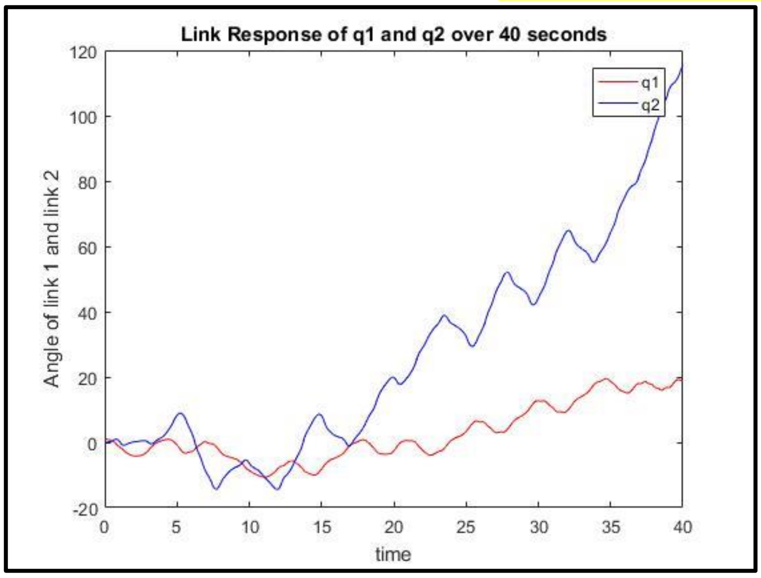

Figure 4 - Responses of q1 and q2 under new conditions

6. Phase space trajectories under new conditions: (question 7 of Problem 1)

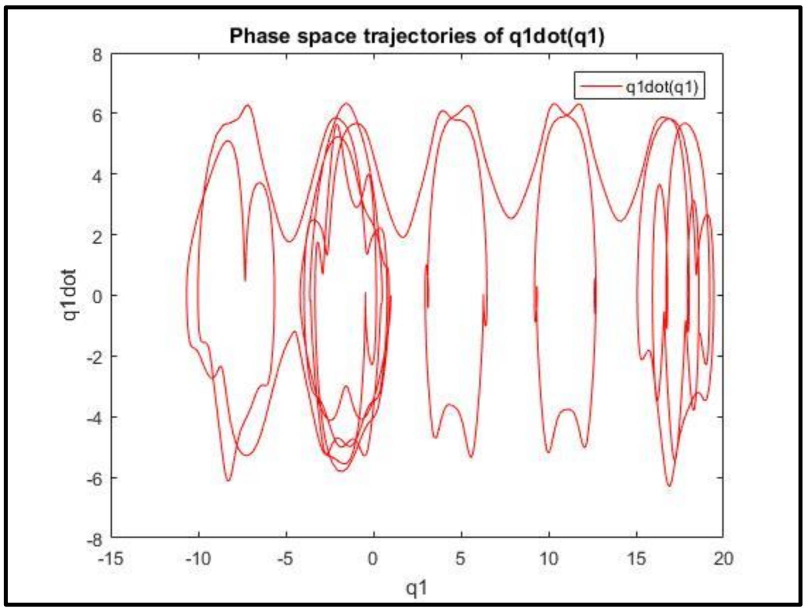

Figure 5 - q1dot(q1) under new conditions

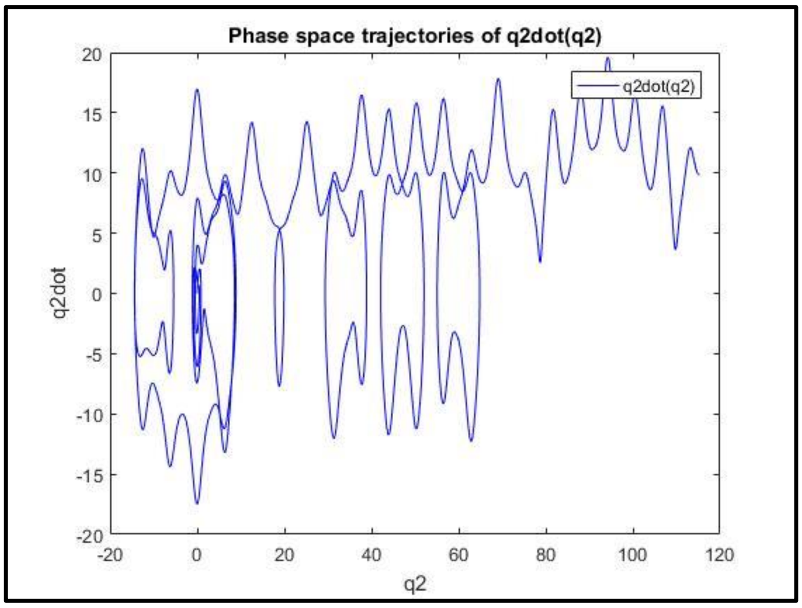

Figure 6 - q2dot(q2) under new conditions

Figure 4,**Figure 5** and **Figure 6** represent the plots of the link responses and the phase space trajectories under the new initial conditions.

The code for the above plots is the same as the one given above for question 5 and 6, with the exception of the initial conditions, i.e. x0.

It can be seen from the above plots that a small change in initial conditions leads to different end trajectories. This falls under a chaotic system.

Problem 2:

1. **Plant model (question 1 of Problem 2)**
   The Plant model is modified according to the parameters given in the question. The gravity terms are removed as the manipulator is horizontal. The input torque calculations are also included inside the function. The plant model is defined in MATLAB as follows

```
%%%Abishek Chandrasekhar
%%%Robot Control_Assignment #6
%%%Problem 2_1

%%function EOM
%%%Defining the state variables
%%% [x(1) x(2) x(3) x(4)] =  [q1 q1dot q2 q2dot]

function [xdot] = EOM1(t,x)

l1 = .3; m1 = 7.848; m2 = 4.49; lc1 = .1554; lc2 = .0341; I1 = .176;
I2 = .0411; %%parameters
kp1 = 100; kd1 = 20; %%%gains
kp2 = kp1; kd2 = kd1;

xdot = zeros (4,1); %%%%initializing xdot

q1d = pi/2*(t>=0 & t<=1);
q2d = pi/2*(t>=0 & t<=1);

%%Input Torque
tau1 = kp1*(q1d - x(1)) - kd1*x(2);
tau2 = kp2*(q2d - x(3)) - kd2*x(4);

%%Inertia Matrix D terms
d11 = m1*lc1^2 + m2*(l1^2 + lc2^2 + 2*l1*lc2^2 + 2*l1*lc2*cos(x(3)))
+ I1 + I2;
d12 = m2*(lc2^2 + l1*lc2*cos(x(3))) + I2;
d21 = d12;
d22 = m2*lc2^2 + I2;

D = [d11 d12; d21 d22];

%%Centrifugal and Coriolis C terms
h = -(m2*l1*lc2*sin(x(3)));
c111 = 0; c121 = h; c211 = h; c221 = h; c112 = -h; c122 = 0; c212 =
0; c222 = 0;

C = [(h*x(4)) (h*(x(4) + x(2))) ; (-h*x(1)) 0];

if tau1 >10
    tau1 = 10;
    t1 = tau1;
elseif tau1 < -10
    tau1 = -10;
    t1 = tau1;
else
    t1 = tau1;
end

if tau2 >10
```

```
        tau2 = 10;
        t2 = tau2;
    elseif tau2 < -10
        tau2 = -10;
        t2 = tau2;
    else
        t2 = tau2;
    end

    %%Acceleration vector
    a1 = t1 - c121*x(2)*x(4) - c211*x(4)*x(2) - c221*x(4)^2;
    a2 = t2 - c112*x(2)^2;
    Dinv = inv(D);
    Atemp = [a1; a2];
    A   = Dinv*Atemp;
    q1ddot = A(1);
    q2ddot = A(2);


    %%%Required vector xdot
    %xdot = [q1dot q1ddot q2dot q2ddot]

    xdot(1)  = x(2);
    xdot(2)  = q1ddot;
    xdot(3)  = x(4);
    xdot(4)  = q2ddot;
```

2. **PD control (question 1 of Problem 2)**
   a) The tracking errors and the reference inputs are defined as given in the question. The code is as follows:

```
%%%%Abishek Chandrasekhar
%%Robot Control_HW_#6
%%Problem 2_2
%%Main program

%%Time
tspan = [0 2];
x0 = [0.05, 0, 0, 0]'; %%%initial conditions

[t,x]=ode45('EOM1',tspan,x0)
[xr, xc] = size(x);
kp1 = 100; kd1 = 20; %%%gains
kp2 = kp1; kd2 = kd1;

for i = 1:xr
    qd(i,1) = pi/2*(t(i)>=0 & t(i)<=1); %%%%reference input
    qd(i,2) = pi/2*(t(i)>=0 & t(i)<=1);
end

for i =1:xr
    e(i,1) = x(i,1) - qd(i,1);
    e(i,2) = x(i,3) - qd(i,2);

    tau1 = kp1*(qd(i,1) - x(i,1)) - kd1*x(i,2);
    tau2 = kp2*(qd(i,2) - x(i,3)) - kd2*x(i,4);
    if tau1 >10
    tau1 = 10;
    %tau(i,1) = tau1;
    elseif tau1 < -10
    tau1 = -10;
    %tau(i,1) = tau1;
    %else

    end

    if tau2 >10
    tau2 = 10;
    %tau(i,2) = tau2;
    elseif tau2 < -10
    tau2 = -10;
    %tau(i,2) = tau2;

    end
    tau(i,1) = tau1;
    tau(i,2) = tau2;
end
```

b) The plot that was described in the question was obtained.

c) The input conditions were also entered. It is defined in the code mentioned above

d) The plots are:

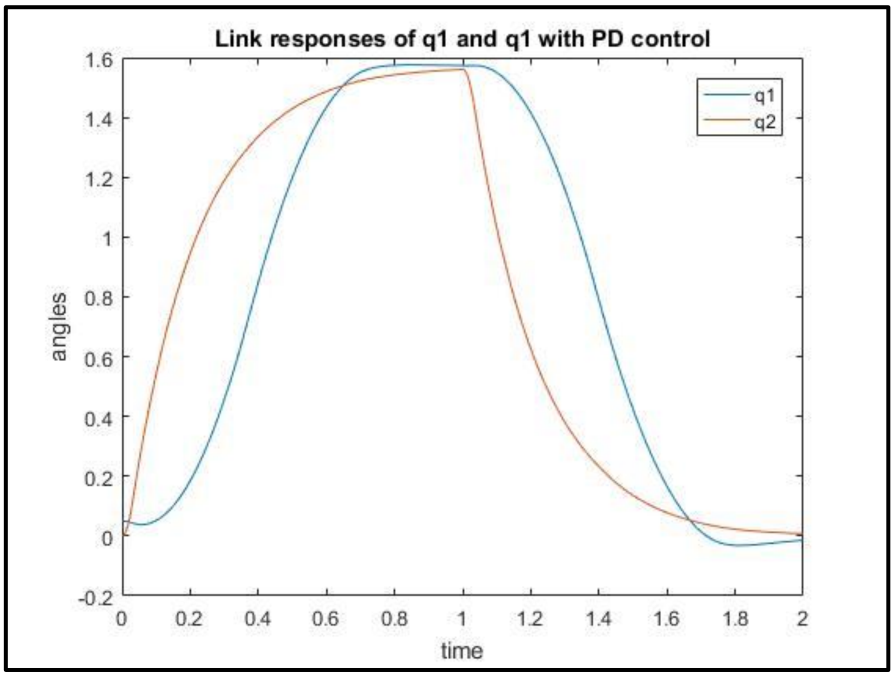

Figure 7- Link responses with PD control

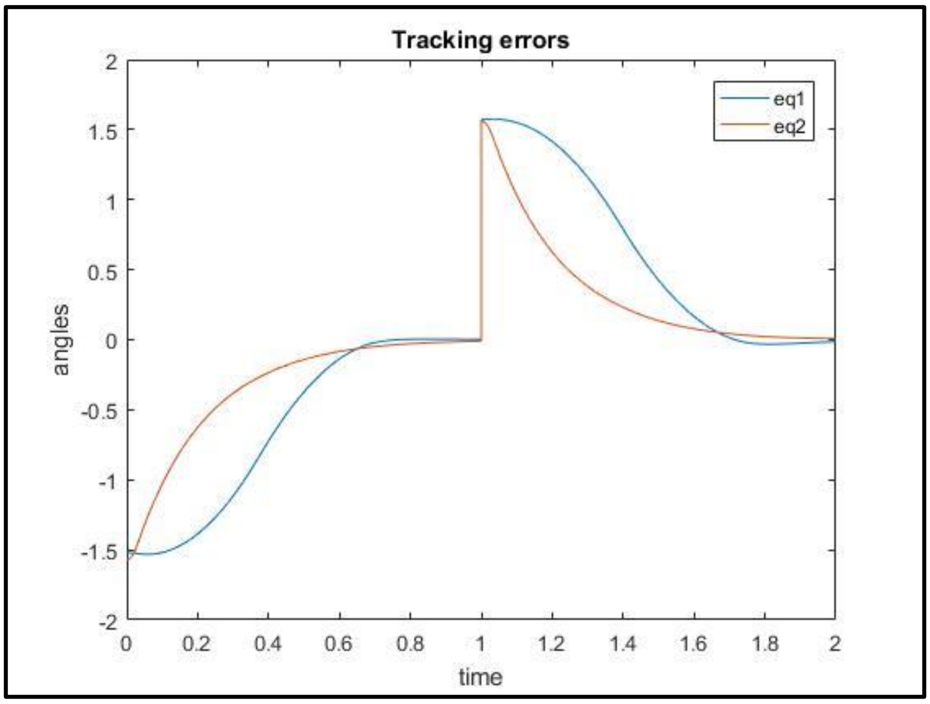

Figure 8 - Tracking errors

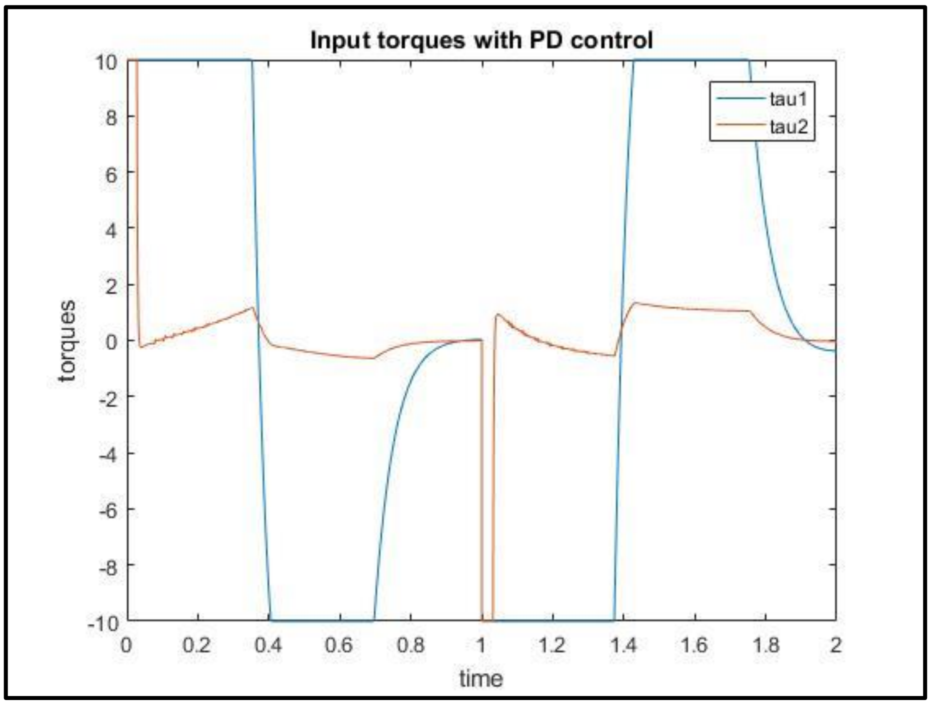

Figure 9 - Torques with PD control

e) The error in position t = 1 second changes the angle from 0 to pi/2 radians. At t = 2 seconds, brings it back to 0.

3. PD plus feedforward control: (question 2 of Problem 2)

a) Cubic polynomial

Plots:

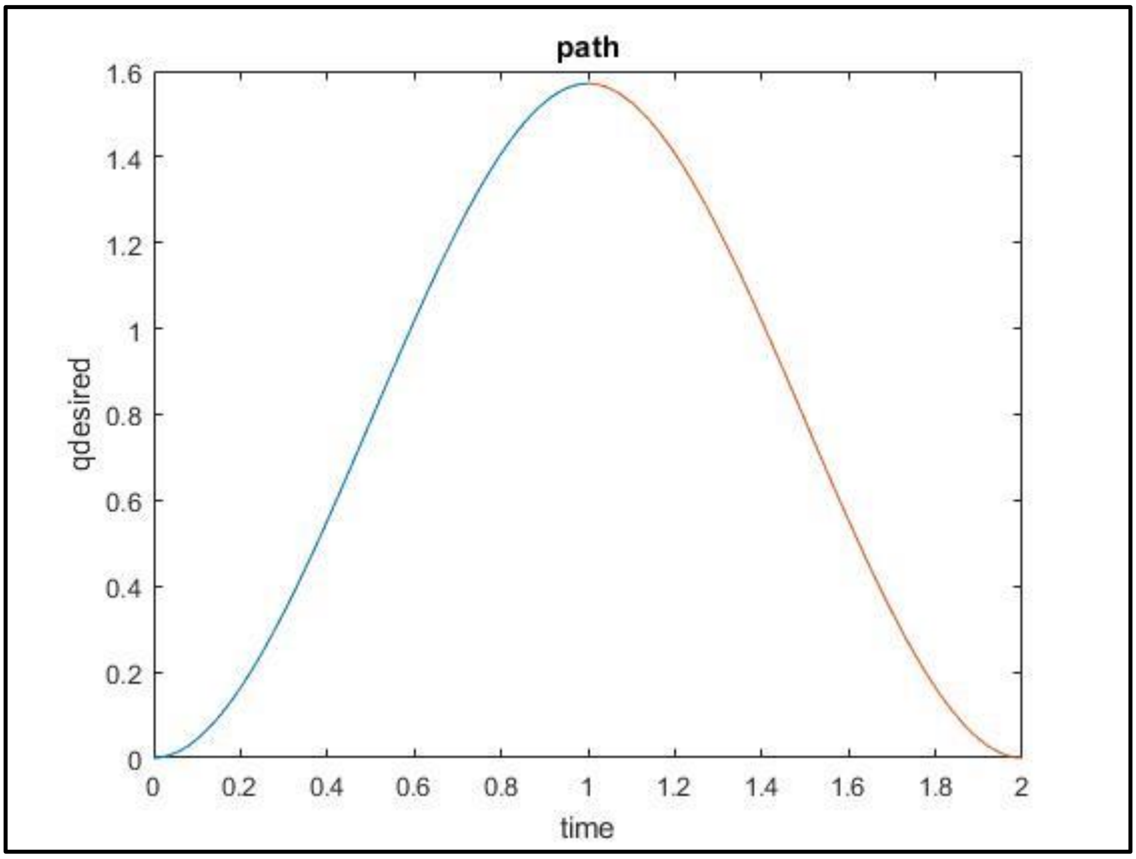

Figure 10 - Qdesired

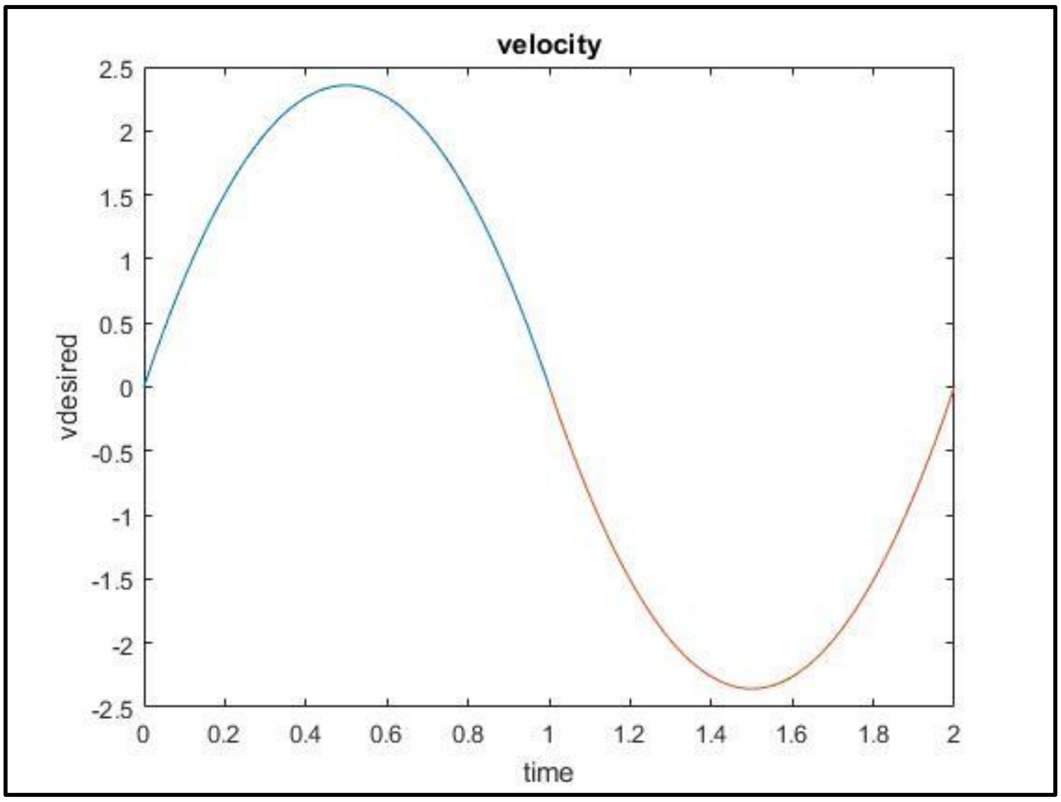

Figure 11 - Vdesired

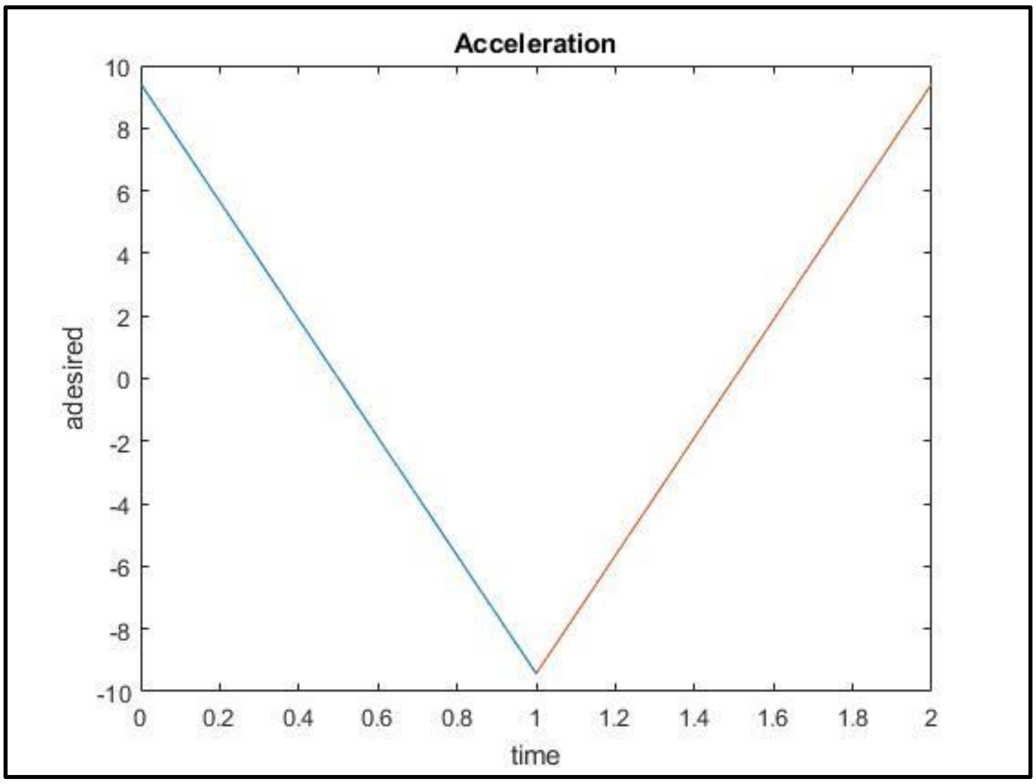

Figure 12 -A desired

MATLAB code for polynomial:

```
%%%%Abishek Chandrasekhar
%%Robot Control_Assignment #6
%%Problem2_2 - Generating the cubic polynomial
q0 = 0; qd0 = 0; t0 = 0; %%initial conditions
q1 = pi/2; qd1 = 0; t1 = 1; %at time t = 1
q2 = 0; qd2 = 0; t2 = 2; %%final time at t = 2
%%%coefficient matrix  for t = 0 to 1;
A1 = [1,  t0,  t0^2, t0^3;
     0,  1,   2*t0, 3*t0^2;
     1,  t1,  t1^2, t1^3;
     0,  1,   2*t1, 3*t1^2];
 A2=  [1,  t1,  t1^2, t1^3;
     0,  1,   2*t1, 3*t1^2;
     1,  t2,  t2^2, t2^3;
     0,  1,   2*t2, 3*t2^2];

 b1 = [q0; qd0; q1; qd1]; %%position and velocity at t = 0 and t = 1
 b2 = [q1; qd1; q2; qd2]; %%position and velocity at t = 1 and t = 2

%%Calculating the coefficients
a1 = inv(A1)*b1;
a2 = inv(A2)*b2;

ta = t0:0.001:t1;
tb = t1:0.001:t2;
q1  = a1(1) + a1(2)*ta + a1(3)*ta.^2 + a1(4)*ta.^3;
q1dot  = a1(2) + 2*a1(3)*ta + 3*a1(4)*ta.^2;

q2  = a2(1) + a2(2)*tb + a2(3)*tb.^2 + a2(4)*tb.^3;
q2dot  = a2(2) + 2*a2(3)*tb + 3*a2(4)*tb.^2;

q1ddot = 2*a1(3) + 6*a1(4)*ta; %%%%Acceleration from t = 0 to 1
q2ddot = 2*a2(3) + 6*a2(4)*tb; %%%%Acceleration from t = 1 to 2

figure
plot(ta,q1,tb,q2);
xlabel('time'); ylabel('qdesired');
title('path');
figure
plot(ta,q1dot,tb,q2dot);
xlabel('time'); ylabel('vdesired');
title('velocity');
figure
plot(ta,q1ddot,tb,q2ddot);
xlabel('time'); ylabel('adesired');
title('Acceleration');
```