ME 5263

MECHANICS AND CONTROL OF ROBOTIC MANIPULATORS

MINI PROJECT REPORT

GROUP MEMBERS

AJAY KRISHNAN V - 132314002
ABHINAND S S - 132001003
LABEEB NASSAR -132001019

INTRODUCTION

In this project we have chosen a 4R spatial manipulator. This manipulator has four degrees of freedom. This manipulator is designed for general purpose operation like assembly and pick and place.

CAD model of the robot

The Robot was designed in Autodesk Fusion 360. Dimensions have been suitably selected. The below figure shows the CAD model of the robot. Dimensions of the robot have been also marked.

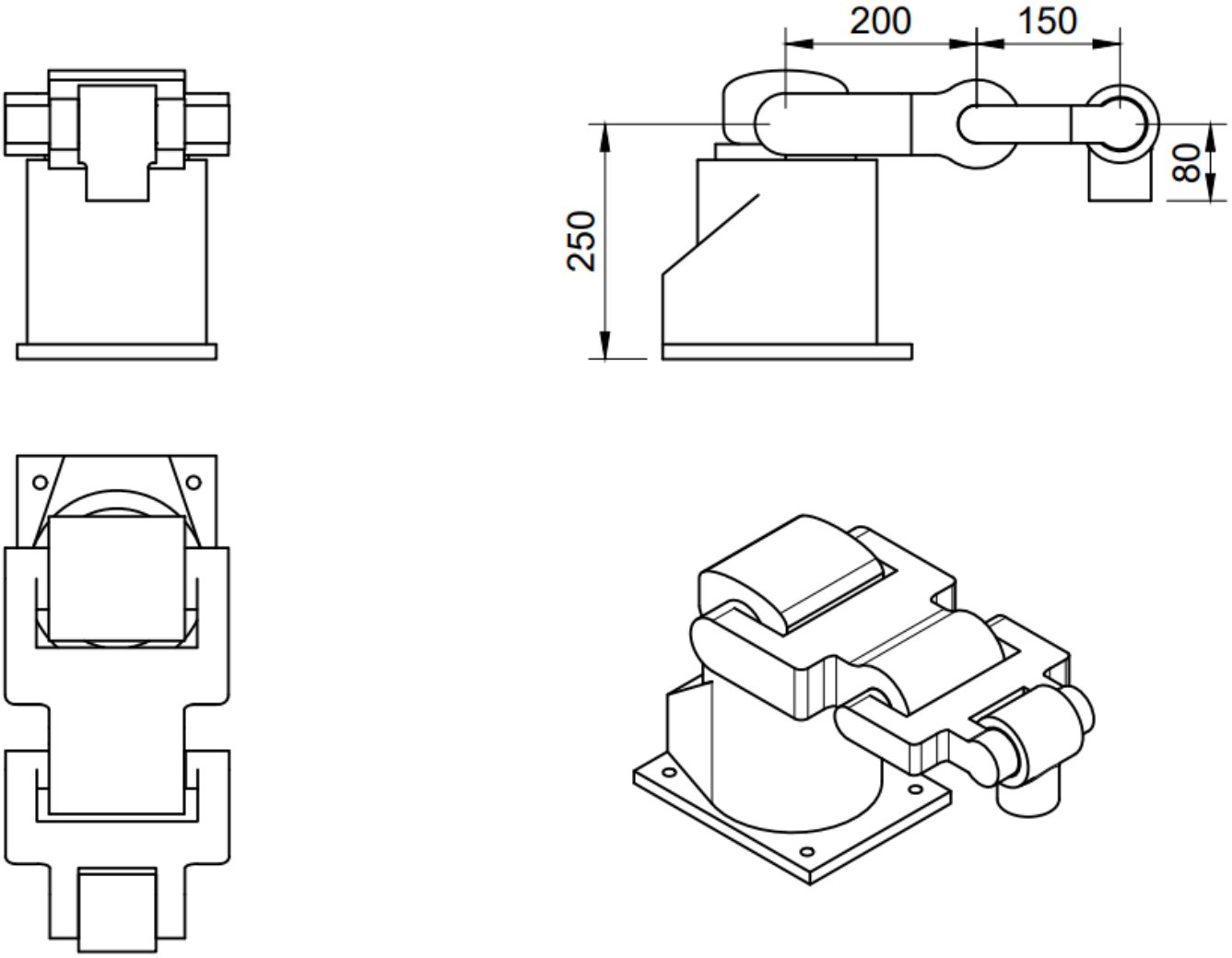

Fig.1 Standard views of the robot with dimensions and the isometric view

Frame assignment

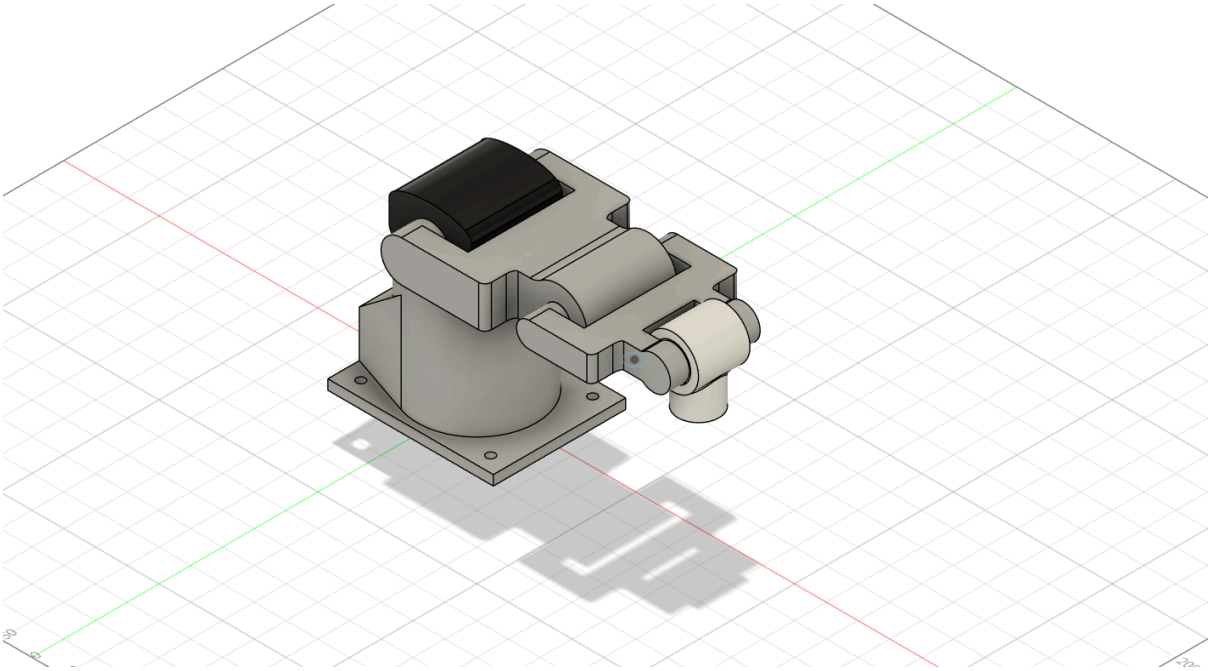

Fig. 2 CAD Model of the robot

Frame arrangement

The below figure shows the frame diagram of the robot. All the joints are revolute. Frames have been assigned based on the D-H Convention. Frame {0} is the base frame and frame {5} is the end effector frame. θ_1,θ_2,θ_3 and θ_4 are the joint variables. d1, l2, l3 and d5 are the link length and are constant. For this robot, d1 = .250m, l2 = .200m, l3 = .150m and d5 = .080m. All the frames have been attached following the D H Convention.

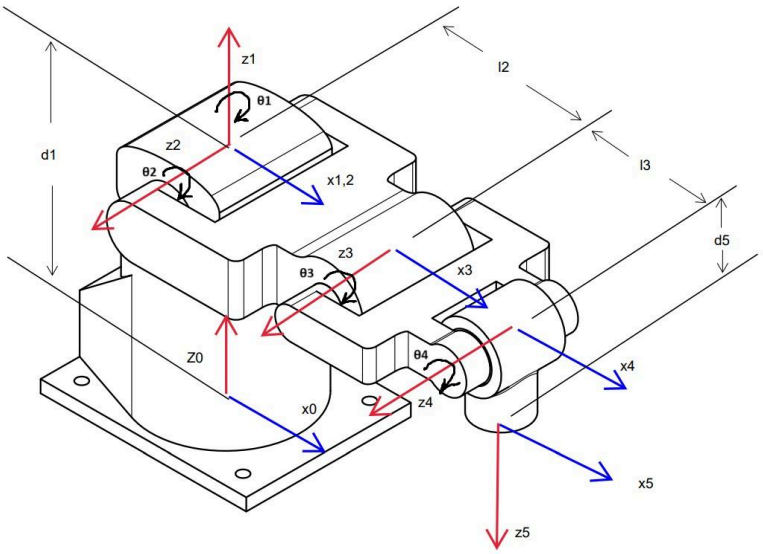

Fig.3 Frame diagram of the robot

Forward Kinematic Model

The forward kinematics of the 4R serial manipulator was derived using MATLAB. Denavit Hartenberg convention was followed and Frames were assigned as shown in Fig. 3. Table 3 shows the D-H Table for the robot.

Table 1 D H Table for the robot.

i	α_{i-1}	a_{i-1}	θ_i	d_i
1	0	0	θ_1	d_1
2	$\pi/2$	0	θ_2	0
3	0	l_2	θ_3	0
4	0	l_3	θ_4	0
5	$\pi/2$	0	0	d_5

The matlab code for deriving forward kinematics is given below.

```
clc;

disp("Forward Kinematics of a spatial 4R Manipulator")
```

```
Forward Kinematics of a spatial 4R Manipulator
```

```
%% Denavit-Hartenberg Parameters

syms theta alpha a d

% Variables

syms  theta1 theta2 theta3 theta4 l1 l2 l3 d1 d5 real

%% DH Table of the manipulator in the order of alpha,a,theta,d

disp("D H Table")
```

```
D H Table
```

```
DH=[0,0,theta1,d1;pi/2,0,theta2,0;0,l2,theta3,0;0,l3,theta4,0;pi/2,0,0,d5]
```

DH = $\begin{pmatrix} 0 & 0 & \theta_1 & d_1 \\ \frac{\pi}{2} & 0 & \theta_2 & 0 \\ 0 & l_2 & \theta_3 & 0 \\ 0 & l_3 & \theta_4 & 0 \\ \frac{\pi}{2} & 0 & 0 & d_5 \end{pmatrix}$

```
%% The general Denavit Hartenberg Transformation matrix

disp("Arm Matrix")
```

Arm Matrix

```
TDH=[cos(theta),-sin(theta),0,a;

   sin(theta)*cos(alpha),cos(theta)*cos(alpha),-sin(alpha),-d*sin(alpha);

   sin(theta)*sin(alpha),cos(theta)*sin(alpha),cos(alpha),d*cos(alpha);

        0,0,0,1]
```

TDH = $\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & a \\ \cos(\alpha)\sin(\theta) & \cos(\alpha)\cos(\theta) & -\sin(\alpha) & -d\sin(\alpha) \\ \sin(\alpha)\sin(\theta) & \sin(\alpha)\cos(\theta) & \cos(\alpha) & d\cos(\alpha) \\ 0 & 0 & 0 & 1 \end{pmatrix}$

```
%% No of frames excluding the base frame

N=5;

%% initialising array

A=cell(1,N);

%% subsituting DH parameters

for i=1:N

        alpha = DH(i,1);

        a = DH(i,2);

        theta = DH(i,3);

        d = DH(i,4);

        A{i} = subs(TDH);

end
```

```
%% intializing 4x4 identity matrix

T = eye(4);

%% multiplying the individual transformation matrices

for i=1:N

        T=T*A{i};

        T = simplify(T);

end

%% individual transformation matrices

T01=A{1};

T12=A{2};

T23=A{3};

T34=A{4};

T45=A{5};

%% Forward kinematics of the robot

disp("forward kinematics (T05)")
```

forward kinematics (T05)

```
disp(T)
```

$$\begin{pmatrix} \sigma_2\cos(\theta_1) & \sin(\theta_1) & \sigma_1\cos(\theta_1) & \cos(\theta_1)\,\sigma_3 + d_5\,\sigma_1\cos(\theta_1) \\ \sigma_2\sin(\theta_1) & -\cos(\theta_1) & \sigma_1\sin(\theta_1) & \sin(\theta_1)\,\sigma_3 + d_5\,\sigma_1\sin(\theta_1) \\ \sigma_1 & 0 & -\sigma_2 & d_1 + l_3\sin(\theta_2+\theta_3) + l_2\sin(\theta_2) - d_5\,\sigma_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \sin(\theta_2+\theta_3+\theta_4)$$

$$\sigma_2 = \cos(\theta_2+\theta_3+\theta_4)$$

$$\sigma_3 = l_3\cos(\theta_2+\theta_3) + l_2\cos(\theta_2)$$

This is the forward Kinematics of the chosen manipulator. Position vector i.e the position of the end -effector with respect to the base, the approach vector, sliding vector and normal vectors have been extracted from the base to end effector transformation matrix (forward kinematics) and is shown below

Position of end effector with respect to base

$$\begin{pmatrix} \cos(\theta_1)\,\sigma_1 + d_5 \sin(\theta_2+\theta_3+\theta_4)\cos(\theta_1) \\ \sin(\theta_1)\,\sigma_1 + d_5 \sin(\theta_2+\theta_3+\theta_4)\sin(\theta_1) \\ d_1 + l_3 \sin(\theta_2+\theta_3) + l_2 \sin(\theta_2) - d_5 \cos(\theta_2+\theta_3+\theta_4) \end{pmatrix}$$

where

P = $\sigma_1 = l_3 \cos(\theta_2+\theta_3) + l_2 \cos(\theta_2)$

From P vector, we have

$x = cos(\theta_1)(l_3 cos(\theta_2 + \theta_3) + l_2 cos(\theta_2)) + d_5 sin(\theta_2 + \theta_3 + \theta_4) cos(\theta_1)$

$y = sin(\theta_1)(l_3 cos(\theta_2 + \theta_3) + l_2 cos(\theta_2)) + d_5 sin(\theta_2 + \theta_3 + \theta_4) sin(\theta_1)$

$z = d_1 + (l_3 sin(\theta_2 + \theta_3) + l_2 sin(\theta_2) - d_5 cos(\theta_2 + \theta_3 + \theta_4)$

Putting θ_1, θ_2, θ_3 and θ_4 =0, we get the home position as

$x = l_3 + l_2$

$y = 0$

$z = d_1 - d_5$

Which gives the x, y and z position of end effector with respect to the base frame.

The normal approach and sliding vectors have been shown below.

Normal Vector

n = $\begin{pmatrix} \cos(\theta_2+\theta_3+\theta_4)\cos(\theta_1) \\ \cos(\theta_2+\theta_3+\theta_4)\sin(\theta_1) \\ \sin(\theta_2+\theta_3+\theta_4) \end{pmatrix}$

Sliding Vector

s = $\begin{pmatrix} \sin(\theta_1) \\ -\cos(\theta_1) \\ 0 \end{pmatrix}$

Approach Vector

$$a = \begin{pmatrix} \sin(\theta_2+\theta_3+\theta_4)\cos(\theta_1) \\ \sin(\theta_2+\theta_3+\theta_4)\sin(\theta_1) \\ -\cos(\theta_2+\theta_3+\theta_4) \end{pmatrix}$$

This is the forward kinematic model of the robot. Using forward kinematics, The home position plot of the robot has been done in MATLAB. The animation of the forward kinematics of the robot has also been developed.

The figure 4 shows the plot of home position of the robot

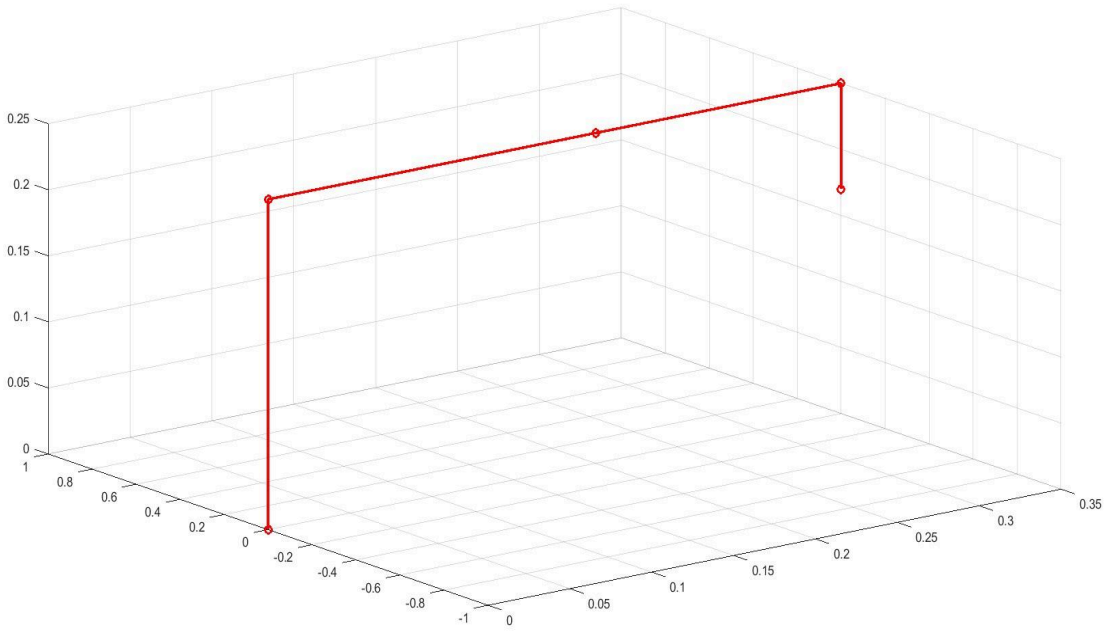

Fig.4 Home position of the robot using Forward Kinematics of the robot

The MATLAB code for forward kinematics animation is shown below.

```
clc;
close all; clear all;
%% Forward Kinematics animation
% joint angle range
theta1 = 0:0.1:pi/2;
theta2 = 0:0.1:pi/2;
theta3 = 0:0.1:pi/2;
theta4 = 0:0.1:pi/2;
```

```
% link lengths
d1 = 0.250;
l2 = 0.200;
l3 = 0.150;
d5 = 0.080;

figure;
for i=1:length(theta1)
x0 = zeros(1,i);
y0 = zeros(1,i);
z0 = zeros(1,i);

x1 = zeros(1,i);
y1 = zeros(1,i);
z1 = d1.*ones(1,i);

x2 = zeros(1,i);
y2 = zeros(1,i);
z2 = d1.*ones(1,i);

x3 = l2*cos(theta1(1,i))*cos(theta2(1,i));
y3 = l2*cos(theta2(1,i))*sin(theta1(1,i));
z3 = l2*sin(theta2(1,i)) + d1;

x4 = l2*cos(theta1(1,i))*cos(theta2(1,i)) -
l3*cos(theta1(1,i))*sin(theta2(1,i))*sin(theta3(1,i)) +
l3*cos(theta1(1,i))*cos(theta2(1,i))*cos(theta3(1,i));
y4 = l2*cos(theta2(1,i))*sin(theta1(1,i)) -
l3*sin(theta1(1,i))*sin(theta2(1,i))*sin(theta3(1,i)) +
l3*cos(theta2(1,i))*cos(theta3(1,i))*sin(theta1(1,i));
z4 = l2*sin(theta2(1,i)) + l3*cos(theta2(1,i))*sin(theta3(1,i)) +
l3*cos(theta3(1,i))*sin(theta2(1,i)) + d1;

x5 = cos(theta1(1,i))*(l3*cos(theta2(1,i) + theta3(1,i)) +
l2*cos(theta2(1,i))) + d5*sin(theta2(1,i) + theta3(1,i) +
theta4(1,i))*cos(theta1(1,i));
y5 = sin(theta1(1,i))*(l3*cos(theta2(1,i) + theta3(1,i)) +
l2*cos(theta2(1,i))) + d5*sin(theta2(1,i) + theta3(1,i) +
theta4(1,i))*sin(theta1(1,i));
z5 = d1 + l3*sin(theta2(1,i) + theta3(1,i)) + l2*sin(theta2(1,i)) -
d5*cos(theta2(1,i) + theta3(1,i) + theta4(1,i));

plot3([x0,x1,x2,x3,x4,x5],[y0,y1,y2,y3,y4,y5],[z0,z1,z2,z3,z4,z5],'r-o',
LineWidth=1.5)
xlim([-0.600,0.600]);
ylim([-0.600,0.600]);
zlim([0,1]);
```

```
title("forward kinematics simulation")

grid on;
hold on;
pause(.1);
hold off;

end
```

Inverse Kinematic Model

The Iterative Newton Raphson method was used to model the inverse kinematics. The forward kinematics is available with us. With inverse kinematics, we have to find the joint variables $\theta_1, \theta_2, \theta_3$ and θ_4 corresponding to the desired end effector positions (task space variables).

For iterative Newton Raphson method of inverse kinematics, we use the following formula

$$q_{i+1} = q_i + J(q_i)^{-1}\delta\mu(q_i)$$

Where q_{i+1} is the joint variable obtained after i^{th} iteration,

q_i is initial guess for joint variable,

$J(q_i)$ is the jacobian matrix.

The initial angular position guess was given and the final values were found out in iterations.

The matlab code for inverse kinematics is as follows

```
clc; close all; clear all;
%% Inverse Kinematics (RRRR Spatial)

% Physical Parameters
d1 = 0.250;
l2 = 0.200;
l3 = 0.150;
d5 = 0.080;

%%Goal position
% mu_a = [cos(theta1)*(l3*cos(theta2 + theta3) + l2*cos(theta2)) +
d5*sin(theta2 + theta3 + theta4)*cos(theta1);
%     sin(theta1)*(l3*cos(theta2 + theta3) + l2*cos(theta2)) +
d5*sin(theta2 + theta3 + theta4)*sin(theta1);
%     d1 + l3*sin(theta2 + theta3) + l2*sin(theta2) - d5*cos(theta2 +
theta3 + theta4)];
```

```
mu_a = [0.2;0.2;0.3]; % giving desired end-effector goal position
%Initial Guess
q = [0;0;0;0];

% iteration time

%Iteration
for i = 1:1000
th1 = q(1);
th2 = q(2);
th3 = q(3);
th4 = q(4);

%Jacobian
J =  [- sin(th1)*(l3*cos(th2 + th3) + l2*cos(th2)) - d5*sin(th2 + th3 +
th4)*sin(th1), d5*cos(th2 + th3 + th4)*cos(th1) - cos(th1)*(l3*sin(th2 +
th3) + l2*sin(th2)), d5*cos(th2 + th3 + th4)*cos(th1) - l3*sin(th2 +
th3)*cos(th1), d5*cos(th2 + th3 + th4)*cos(th1);
    cos(th1)*(l3*cos(th2 + th3) + l2*cos(th2)) + d5*sin(th2 + th3 +
th4)*cos(th1), d5*cos(th2 + th3 + th4)*sin(th1) - sin(th1)*(l3*sin(th2 +
th3) + l2*sin(th2)), d5*cos(th2 + th3 + th4)*sin(th1) - l3*sin(th2 +
th3)*sin(th1), d5*cos(th2 + th3 + th4)*sin(th1);
    0, l3*cos(th2 + th3) + l2*cos(th2) + d5*sin(th2 + th3 + th4),
l3*cos(th2 + th3) + d5*sin(th2 + th3 + th4), d5*sin(th2 + th3 + th4)];
%Estimated position
mu_e = [cos(th1)*(l3*cos(th2 + th3) + l2*cos(th2)) + d5*sin(th2 + th3 +
th4)*cos(th1);
    sin(th1)*(l3*cos(th2 + th3) + l2*cos(th2)) + d5*sin(th2 + th3 +
th4)*sin(th1);
    d1 + l3*sin(th2 + th3) + l2*sin(th2) - d5*cos(th2 + th3 + th4)];
%error
delta = mu_a - mu_e;

if abs(delta) < 1e-10
    break;
end

%updating the joint angles
q = q + pinv(J)*delta;

% animation
x0 = 0.0;
y0 = 0.0;
z0 = 0.0;

x1 = 0.0;
```

```
y1 = 0.0;
z1 = d1;

x2 = 0.0;
y2 = 0;
z2 = d1;

x3 = l2*cos(th1)*cos(th2);
y3 = l2*cos(th2)*sin(th1);
z3 = l2*sin(th2) + d1;

x4 = l2*cos( th1 )*cos( th2 ) - l3*cos( th1 )*sin( th2 )*sin( th3 ) +
l3*cos( th1 )*cos( th2 )*cos( th3 );
y4 = l2*cos( th2 )*sin( th1 ) - l3*sin( th1 )*sin( th2 )*sin( th3 ) +
l3*cos( th2 )*cos( th3 )*sin( th1 );
z4 = l2*sin( th2 ) + l3*cos( th2 )*sin( th3 ) + l3*cos( th3 )*sin( th2 )
+ d1;

plot3(mu_a(1), mu_a(2), mu_a(3), 'kx')
hold on
plot3([x0,x1,x2,x3,x4,mu_e(1)],[y0,y1,y2,y3,y4,mu_e(2)],[z0,z1,z2,z3,z4,
mu_e(3)],'r-o',LineWidth=1)
grid on
xlim([-0.6,0.6]);
ylim([-0.6,0.6]);
zlim([0,0.6]);
view([-9 33])
hold on
title("Inverse kinematics animation")

pause(0.5)
hold off
end
```

Workspace Analysis

Workspace analysis was done using MATLAB using the forward kinematics simulation by plotting the end effector location for looped values of different angles of the joints.

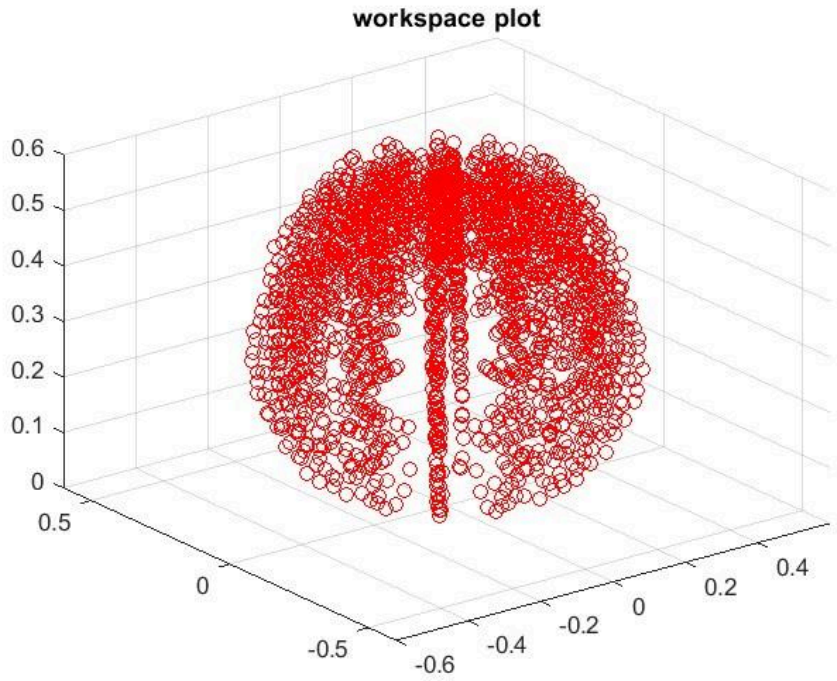

Fig.5 Workspace of the robot

Velocity analysis

Velocity analysis of the selected robot was done in MATLAB. The linear velocity and angular velocity of the base of the robot was taken as 0. The velocity was then propagated from base to end effector. The MATLAB code for velocity analysis is shown below.

```
% Individual rotation matrix and position vectors
R01 = A{1}(1:3,1:3);
P01 = A{1}(1:3,4);
R12= A{2}(1:3,1:3);
P12 =A{2}(1:3,4);
R23= A{3}(1:3,1:3);
P23 =A{3}(1:3,4);
R34= A{4}(1:3,1:3);
P34 =A{4}(1:3,4);
R45= A{5}(1:3,1:3);
P45 =A{5}(1:3,4);
%position of frame wrt base
T02 = T01*T12;
T03 = T01*T12*T23;
T04 = T01*T12*T23*T34;
T05 = T;
%position of each frame wrt to base
P01;
P02 = T02(1:3,4);
P03 = T03(1:3,4);
P04 = T04(1:3,4);
P05 = P;
%angular velocity of the individual joints
```

```
syms theta1dot theta2dot theta3dot theta4dot theta1ddot theta2ddot theta3ddot
theta4ddot g real
% angular velocity propagation
w0 = [0;0;0];
w1 = R01'*w0 + [0;0;theta1dot];
w2 = R12'*w1 + [0;0;theta2dot];
w3 = R23'*w2 + [0;0;theta3dot];
w4 = R34'*w3 + [0;0;theta4dot];
w5 = R45'*w4;
% End effector angular velocity w.r.t base frame
w05 = R01*R12*R23*R34*R45 * w5;
%% Linear velocity propagation
v0 = [0;0;0];
v1 = R01'*(v0 + cross(w0,P01));
v2 = R12'*(v1 + cross(w1,P12));
v3 = R23'*(v2 + cross(w1,P23));
v4 = R34'*(v3 + cross(w1,P34));
v5 = R45'*(v4 + cross(w1,P45));
% End effector linear velocity w.r.t base frame
v05 = simplify (R01*R12*R23*R34*R45*v5);
% Jacobian Matrix
% J=equationsToMatrix(v05,[theta1dot;theta2dot;theta3dot;theta4dot])
%Jacobian by partially differentiating position vector
J1 =[diff(P(1),theta1),diff(P(1),theta2),diff(P(1),theta3),diff(P(1),theta4);
   diff(P(2),theta1),diff(P(2),theta2),diff(P(2),theta3),diff(P(2),theta4);
   diff(P(3),theta1),diff(P(3),theta2),diff(P(3),theta3),diff(P(3),theta4)];
simplify(J1)
```

Jacobian Matrix, $\mathbb{J}$(q) =

$$\begin{pmatrix} -\sin(\theta_1)\,\sigma_5 - d_5\,\sigma_2\,\sin(\theta_1) & \sigma_4 - \cos(\theta_1)\,\sigma_1 & \sigma_4 - l_3\,\sin(\theta_2+\theta_3)\,\cos(\theta_1) & \sigma_4 \\ \cos(\theta_1)\,\sigma_5 + d_5\,\sigma_2\,\cos(\theta_1) & \sigma_3 - \sin(\theta_1)\,\sigma_1 & \sigma_3 - l_3\,\sin(\theta_2+\theta_3)\,\sin(\theta_1) & \sigma_3 \\ 0 & \sigma_7 + l_2\,\cos(\theta_2) + d_5\,\sigma_2 & \sigma_7 + d_5\,\sigma_2 & d_5\,\sigma_2 \end{pmatrix}$$

where

$$\sigma_1 = l_3\,\sin(\theta_2+\theta_3) + l_2\,\sin(\theta_2)$$

$$\sigma_2 = \sin(\theta_2+\theta_3+\theta_4)$$

$$\sigma_3 = d_5\,\sigma_6\,\sin(\theta_1)$$

$$\sigma_4 = d_5\,\sigma_6\,\cos(\theta_1)$$

$$\sigma_5 = \sigma_7 + l_2\,\cos(\theta_2)$$

$$\sigma_6 = \cos(\theta_2+\theta_3+\theta_4)$$

$$\sigma_7 = l_3\,\cos(\theta_2+\theta_3)$$

Dynamic Model

To obtain the dynamic model, Newton Euler method was used. The code for deriving the dynamic model is shown below. The position of center of mass of each link and the inertia tensor of each link was obtained from the cad model.

```
% Dynamic model
%angular acceleration
al0 = [0;0;0];
al1 = R01'*(al0+cross(w0,[0;0;theta1dot]))+[0;0;theta1ddot];
al2 = R12'*(al1+cross(w1,[0;0;theta2dot]))+[0;0;theta2ddot];
al3 = R23'*(al2+cross(w2,[0;0;theta3dot]))+[0;0;theta3ddot];
al4 = R34'*(al3+cross(w3,[0;0;theta4dot]))+[0;0;theta4ddot];
al5 = R45'*(al4);
% End effector angular accelaration w.r.t base frame
al05 = R01*R12*R23*R34*R45* al5;
%Linear accelaration
a0 = [0;0;g];
a1 = R01'*(a0+cross(al0,P01)+cross(w0,cross(w0,P01)));
a2 = R12'*(a1+cross(al1,P12)+cross(w1,cross(w1,P12)));
a3 = R23'*(a2+cross(al2,P23)+cross(w2,cross(w2,P23)));
a4 = R34'*(a3+cross(al3,P34)+cross(w3,cross(w3,P34)));
a5 = R45'*(a4+cross(al4,P45)+cross(w4,cross(w4,P45)));
%Linear acceleration wrt base
a05 = simplify(R01*R12*R23*R34*R45* a5);
%Centre of mass,mass and Inertia of each link
syms xc1 yc1 zc1 Ix1 Iy1 Iz1 xc2 yc2 zc2 Ix2 Iy2 Iz2 xc3 yc3 zc3 Ix3 Iy3 Iz3 xc4
yc4 zc4 Ix4 Iy4 Iz4 xc5 yc5 zc5 Ix5 Iy5 Iz5 mc1 mc2 mc3 mc4
Pc1= [xc1;yc1;zc1];
Ic1 = [Ix1,0,0;0,Iy1,0;0,0,Iz1];
Pc2 = [xc2;yc2;zc2];
Ic2 = [Ix2,0,0;0,Iy2,0;0,0,Iz2];
Pc3 = [xc3;yc3;zc3];
Ic3 = [Ix3,0,0;0,Iy3,0;0,0,Iz3];
Pc4 = [xc4;yc4;zc4];
Ic4 = [Ix4,0,0;0,Iy4,0;0,0,Iz4];
Pc5 = [xc5;yc5;zc5];
Ic5 = [Ix5,0,0;0,Iy5,0;0,0,Iz5];
%Acceleration of the center of mass of links
ac1 = a1 +cross(al1,Pc1)+cross(w1,cross(w1,Pc1));
ac2 = a2 +cross(al2,Pc2)+cross(w2,cross(w2,Pc2));
ac3 = a3 +cross(al3,Pc3)+cross(w3,cross(w3,Pc3));
ac4 = a4 +cross(al4,Pc4)+cross(w4,cross(w4,Pc4));
%link forces
F1 = mc1*ac1;
F2 = mc2*ac1;
F3 = mc3*ac3;
F4 = mc4*ac4;
%link moments
N1 = Ic1*al1 + cross(w1,Ic1*w1);
N2 = Ic2*al2 + cross(w2,Ic2*w2);
N3 = Ic3*al3 + cross(w3,Ic3*w3);
N4 = Ic4*al4 + cross(w4,Ic4*w4);
%Force and Moment propagation
```

```
%force and moment of end effector
f5 = [0;0;0];
n5 = [0;0;0];
f4 = R45*f5 + F4;
n4 = R45*n5 + N4 + cross(Pc4,F4) + cross(P45, R45*f5);
f3 = R34*f4 + F3;
n3 = R34*n4 + N3 + cross(Pc3,F3) + cross(P34, R34*f4);
f2 = R23*f3 + F2;
n2 = R23*n3 + N2 + cross(Pc2,F2) + cross(P23, R23*f4);
f1 = R12*f2 + F1;
n1 = R12*n2 + N1 + cross(Pc1,F1) + cross(P12, R12*f3);
f0 = R01*f1;
n0 = R01*n1 + cross(P01, R01*f1);
% joint torques
tau1 = n1(3);
tau2 = n2(3);
tau3 = n3(3);
tau4 = n4(3);
% inertia matrix parameter
m_11 = (mc4*l2^2 + 2*mc4*l2*l3 + mc3*l2*xc3 + 2*mc4*l2*xc4 + mc4*l3^2 +
2*mc4*l3*xc4 + mc1*xc1^2 + mc3*xc3^2 + mc4*xc4^2 + mc1*yc1^2 - mc2*zc2*yc1 +
mc3*zc3^2 + mc4*zc4^2 + Iy2 + Iy3 + Iy4 + Iz1);
m_12 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_13 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_14 =  -(mc4*yc4*zc4);
m_21 = (mc2*xc1*xc2 + mc2*yc1*yc2 - mc3*yc3*zc3 - mc4*yc4*zc4);
m_22 = (mc4*l2^2 + 2*mc4*l2*l3 + mc3*l2*xc3 + 2*mc4*l2*xc4 + mc4*l3^2 +
2*mc4*l3*xc4 + mc3*xc3^2 + mc4*xc4^2 + mc3*yc3^2 + mc4*yc4^2 + Iz2 + Iz3 + Iz4);
m_23 = (mc4*l3^2 + 2*mc4*l3*xc4 + l2*mc4*l3 + mc3*xc3^2 + mc4*xc4^2 + l2*mc4*xc4 +
mc3*yc3^2 + mc4*yc4^2 + Iz3 + Iz4);
m_24 = (Iz4 + mc4*xc4^2 + mc4*yc4^2 + l2*mc4*xc4 + l3*mc4*xc4);
m_31 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_32 = (mc4*l3^2 + 2*mc4*l3*xc4 + l2*mc4*l3 + mc3*xc3^2 + l2*mc3*xc3 + mc4*xc4^2 +
l2*mc4*xc4 + mc3*yc3^2 + mc4*yc4^2 + Iz3 + Iz4);
m_33 = (mc4*l3^2 + 2*mc4*l3*xc4 + mc3*xc3^2 + mc4*xc4^2 + mc3*yc3^2 + mc4*yc4^2 +
Iz3 + Iz4);
m_34 = (mc4*xc4^2 + l3*mc4*xc4 + mc4*yc4^2 + Iz4);
m_41 = -(mc4*yc4*zc4);
m_42 = (mc4*yc4^2+Iz4+mc4*xc4*(l2+l3+xc4));
m_43 = (mc4*xc4^2 + l3*mc4*xc4 + mc4*yc4^2 + Iz4);
m_44 = (mc4*xc4^2 + mc4*yc4^2 + Iz4);
M = [m_11, m_12, m_13, m_14;
   m_21, m_22, m_23, m_24;
   m_31, m_32, m_33, m_34;
   m_41, m_42, m_43, m_44];
% other effects
oe_v1 = (Ix3*theta1dot*theta2dot*sin(2*theta2 + 2*theta3))/2 +
Ix3*theta1dot*theta3dot*sin(2*theta2 + 2*theta3) -
(Iy3*theta1dot*theta2dot*sin(2*theta2 + 2*theta3))/2 -
Iy3*theta1dot*theta3dot*sin(2*theta2 + 2*theta3) +
(Ix2*theta1dot*theta2dot*sin(2*theta2))/2 -
(Iy2*theta1dot*theta2dot*sin(2*theta2))/2 + (Ix4*theta1dot*theta2dot*sin(2*theta2
+ 2*theta3 + 2*theta4))/2 + Ix4*theta1dot*theta3dot*sin(2*theta2 + 2*theta3 +
```

```
2*theta4) + Ix4*theta1dot*theta4dot*sin(2*theta2 + 2*theta3 + 2*theta4) -
(Iy4*theta1dot*theta2dot*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
Iy4*theta1dot*theta3dot*sin(2*theta2 + 2*theta3 + 2*theta4) -
Iy4*theta1dot*theta4dot*sin(2*theta2 + 2*theta3 + 2*theta4) -
(mc3*theta1dot*theta2dot*xc3^2*sin(2*theta2 + 2*theta3))/2 -
mc3*theta1dot*theta3dot*xc3^2*sin(2*theta2 + 2*theta3) +
(mc3*theta1dot*theta2dot*yc3^2*sin(2*theta2 + 2*theta3))/2 +
mc3*theta1dot*theta3dot*yc3^2*sin(2*theta2 + 2*theta3) -
l3*mc4*theta2dot^2*zc4*cos(theta2 + theta3) - l3*mc4*theta3dot^2*zc4*cos(theta2 +
theta3) - mc3*theta2dot^2*xc3*zc3*cos(theta2 + theta3) -
mc3*theta3dot^2*xc3*zc3*cos(theta2 + theta3) + mc3*theta2dot^2*yc3*zc3*sin(theta2
+ theta3) + mc3*theta3dot^2*yc3*zc3*sin(theta2 + theta3) -
l2*mc3*theta1dot^2*zc3*cos(theta2) - l2*mc3*theta2dot^2*zc3*cos(theta2) -
l2*mc4*theta2dot^2*zc4*cos(theta2) - mc2*theta1dot^2*xc1*zc2*cos(theta2) +
mc2*theta1dot^2*yc1*zc2*sin(theta2) - mc4*theta2dot^2*xc4*zc4*cos(theta2 + theta3
+ theta4) - mc4*theta3dot^2*xc4*zc4*cos(theta2 + theta3 + theta4) -
mc4*theta4dot^2*xc4*zc4*cos(theta2 + theta3 + theta4) +
mc4*theta2dot^2*yc4*zc4*sin(theta2 + theta3 + theta4) +
mc4*theta3dot^2*yc4*zc4*sin(theta2 + theta3 + theta4) +
mc4*theta4dot^2*yc4*zc4*sin(theta2 + theta3 + theta4) -
(l2^2*mc4*theta1dot*theta2dot*sin(2*theta2))/2 -
(mc4*theta1dot*theta2dot*xc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
mc4*theta1dot*theta3dot*xc4^2*sin(2*theta2 + 2*theta3 + 2*theta4) -
mc4*theta1dot*theta4dot*xc4^2*sin(2*theta2 + 2*theta3 + 2*theta4) +
(mc4*theta1dot*theta2dot*yc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 +
mc4*theta1dot*theta3dot*yc4^2*sin(2*theta2 + 2*theta3 + 2*theta4) +
mc4*theta1dot*theta4dot*yc4^2*sin(2*theta2 + 2*theta3 + 2*theta4) -
(l3^2*mc4*theta1dot*theta2dot*sin(2*theta2 + 2*theta3))/2 -
l3^2*mc4*theta1dot*theta3dot*sin(2*theta2 + 2*theta3) -
mc3*theta1dot*theta2dot*xc3*yc3*cos(2*theta2 + 2*theta3) -
2*mc3*theta1dot*theta3dot*xc3*yc3*cos(2*theta2 + 2*theta3) -
l2*mc4*theta1dot*theta2dot*yc4*cos(2*theta2 + theta3 + theta4) -
l2*mc4*theta1dot*theta3dot*yc4*cos(2*theta2 + theta3 + theta4) -
l2*mc4*theta1dot*theta4dot*yc4*cos(2*theta2 + theta3 + theta4) -
l2*mc4*theta1dot*theta2dot*xc4*sin(2*theta2 + theta3 + theta4) -
l2*mc4*theta1dot*theta3dot*xc4*sin(2*theta2 + theta3 + theta4) -
l2*mc4*theta1dot*theta4dot*xc4*sin(2*theta2 + theta3 + theta4) -
l2*mc4*theta1dot*theta3dot*yc4*cos(theta3 + theta4) -
l2*mc4*theta1dot*theta4dot*yc4*cos(theta3 + theta4) -
2*l3*mc4*theta2dot*theta3dot*zc4*cos(theta2 + theta3) -
2*mc3*theta2dot*theta3dot*xc3*zc3*cos(theta2 + theta3) -
l2*mc4*theta1dot*theta3dot*xc4*sin(theta3 + theta4) -
l2*mc4*theta1dot*theta4dot*xc4*sin(theta3 + theta4) +
2*mc3*theta2dot*theta3dot*yc3*zc3*sin(theta2 + theta3) +
(l2*mc3*theta1dot*theta2dot*yc3*cos(theta3))/2 -
l3*mc4*theta1dot*theta4dot*yc4*cos(theta4) -
l2*l3*mc4*theta1dot*theta3dot*sin(theta3) -
l3*mc4*theta1dot*theta2dot*yc4*cos(2*theta2 + 2*theta3 + theta4) -
2*l3*mc4*theta1dot*theta3dot*yc4*cos(2*theta2 + 2*theta3 + theta4) -
l3*mc4*theta1dot*theta4dot*yc4*cos(2*theta2 + 2*theta3 + theta4) +
(l2*mc3*theta1dot*theta2dot*xc3*sin(theta3))/2 -
l3*mc4*theta1dot*theta4dot*xc4*sin(theta4) -
l3*mc4*theta1dot*theta2dot*xc4*sin(2*theta2 + 2*theta3 + theta4) -
```

```
2*l3*mc4*theta1dot*theta3dot*xc4*sin(2*theta2 + 2*theta3 + theta4) -
l3*mc4*theta1dot*theta4dot*xc4*sin(2*theta2 + 2*theta3 + theta4) -
(l2*mc3*theta1dot*theta2dot*yc3*cos(2*theta2 + theta3))/2 -
l2*l3*mc4*theta1dot*theta2dot*sin(2*theta2 + theta3) -
l2*l3*mc4*theta1dot*theta3dot*sin(2*theta2 + theta3) -
(l2*mc3*theta1dot*theta2dot*xc3*sin(2*theta2 + theta3))/2 -
2*mc4*theta2dot*theta3dot*xc4*zc4*cos(theta2 + theta3 + theta4) -
2*mc4*theta2dot*theta4dot*xc4*zc4*cos(theta2 + theta3 + theta4) -
2*mc4*theta3dot*theta4dot*xc4*zc4*cos(theta2 + theta3 + theta4) +
2*mc4*theta2dot*theta3dot*yc4*zc4*sin(theta2 + theta3 + theta4) +
2*mc4*theta2dot*theta4dot*yc4*zc4*sin(theta2 + theta3 + theta4) +
2*mc4*theta3dot*theta4dot*yc4*zc4*sin(theta2 + theta3 + theta4) -
mc4*theta1dot*theta2dot*xc4*yc4*cos(2*theta2 + 2*theta3 + 2*theta4) -
2*mc4*theta1dot*theta3dot*xc4*yc4*cos(2*theta2 + 2*theta3 + 2*theta4) -
2*mc4*theta1dot*theta4dot*xc4*yc4*cos(2*theta2 + 2*theta3 + 2*theta4);
oe_v2 = (Iy2*theta1dot^2*sin(2*theta2))/2 - (Ix2*theta1dot^2*sin(2*theta2))/2 -
(Ix4*theta1dot^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 +
(Iy4*theta1dot^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
(Ix3*theta1dot^2*sin(2*theta2 + 2*theta3))/2 + (Iy3*theta1dot^2*sin(2*theta2 +
2*theta3))/2 + (l2^2*mc4*theta1dot^2*sin(theta4))/2 +
l2^2*mc4*theta2dot^2*sin(theta4) + (l2^2*mc4*theta1dot^2*sin(2*theta2 + theta4))/2
+ mc2*theta1dot^2*xc1*yc2 - mc2*theta1dot^2*xc2*yc1 +
(mc4*theta1dot^2*xc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
(mc4*theta1dot^2*yc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 +
(l3^2*mc4*theta1dot^2*sin(2*theta2 + 2*theta3))/2 +
(mc3*theta1dot^2*xc3^2*sin(2*theta2 + 2*theta3))/2 -
(mc3*theta1dot^2*yc3^2*sin(2*theta2 + 2*theta3))/2 +
(l2*mc4*theta1dot^2*yc4*cos(2*theta2 + theta3 + theta4))/2 +
(l2*l3*mc4*theta1dot^2*sin(2*theta2 + theta3 + theta4))/2 +
(l2*mc4*theta1dot^2*xc4*sin(2*theta2 + theta3 + theta4))/2 +
(l2*mc4*theta1dot^2*yc4*cos(theta3 + theta4))/2 +
l2*mc4*theta2dot^2*yc4*cos(theta3 + theta4) + (l2*mc4*theta1dot^2*xc4*sin(theta3 +
theta4))/2 + l2*mc4*theta2dot^2*xc4*sin(theta3 + theta4) +
(l2*mc3*theta1dot^2*yc3*cos(theta3))/2 + l2*mc3*theta2dot^2*yc3*cos(theta3) -
(l2*mc4*theta1dot^2*yc4*cos(theta3))/2 - l2*mc4*theta2dot^2*yc4*cos(theta3) -
l2*mc4*theta3dot^2*yc4*cos(theta3) - l2*mc4*theta4dot^2*yc4*cos(theta3) -
l3*mc4*theta4dot^2*yc4*cos(theta4) + (l2*l3*mc4*theta1dot^2*sin(theta3))/2 +
l2*l3*mc4*theta2dot^2*sin(theta3) + (l2*mc4*theta1dot^2*yc4*cos(2*theta2 + theta3
+ 2*theta4))/2 + l3*mc4*theta1dot^2*yc4*cos(2*theta2 + 2*theta3 + theta4) +
(l2*mc3*theta1dot^2*xc3*sin(theta3))/2 + l2*mc3*theta2dot^2*xc3*sin(theta3) -
(l2*mc4*theta1dot^2*xc4*sin(theta3))/2 - l2*mc4*theta2dot^2*xc4*sin(theta3) -
l2*mc4*theta3dot^2*xc4*sin(theta3) - l2*mc4*theta4dot^2*xc4*sin(theta3) -
l3*mc4*theta4dot^2*xc4*sin(theta4) + (l2*mc4*theta1dot^2*xc4*sin(2*theta2 + theta3
+ 2*theta4))/2 + l3*mc4*theta1dot^2*xc4*sin(2*theta2 + 2*theta3 + theta4) +
(l2*mc3*theta1dot^2*yc3*cos(2*theta2 + theta3))/2 +
(l2*l3*mc4*theta1dot^2*sin(2*theta2 + theta3))/2 -
(l2*l3*mc4*theta1dot^2*sin(theta3 - theta4))/2 - l2*l3*mc4*theta2dot^2*sin(theta3
- theta4) - l2*l3*mc4*theta3dot^2*sin(theta3 - theta4) +
(l2*mc3*theta1dot^2*xc3*sin(2*theta2 + theta3))/2 +
mc4*theta1dot^2*xc4*yc4*cos(2*theta2 + 2*theta3 + 2*theta4) +
mc3*theta1dot^2*xc3*yc3*cos(2*theta2 + 2*theta3) +
l2*mc4*theta1dot*theta2dot*zc4*cos(theta2 + theta4) +
l3*mc4*theta1dot*theta2dot*zc4*cos(theta2 + theta3) +
```

```
mc3*theta1dot*theta2dot*xc3*zc3*cos(theta2 + theta3) -
mc3*theta1dot*theta2dot*yc3*zc3*sin(theta2 + theta3) -
2*l2*mc4*theta2dot*theta3dot*yc4*cos(theta3) -
2*l2*mc4*theta2dot*theta4dot*yc4*cos(theta3) -
2*l2*mc4*theta3dot*theta4dot*yc4*cos(theta3) -
2*l3*mc4*theta2dot*theta4dot*yc4*cos(theta4) -
2*l3*mc4*theta3dot*theta4dot*yc4*cos(theta4) -
2*l2*mc4*theta2dot*theta3dot*xc4*sin(theta3) -
2*l2*mc4*theta2dot*theta4dot*xc4*sin(theta3) -
2*l2*mc4*theta3dot*theta4dot*xc4*sin(theta3) -
2*l3*mc4*theta2dot*theta4dot*xc4*sin(theta4) -
2*l3*mc4*theta3dot*theta4dot*xc4*sin(theta4) -
2*l2*l3*mc4*theta2dot*theta3dot*sin(theta3 - theta4) +
mc4*theta1dot*theta2dot*xc4*zc4*cos(theta2 + theta3 + theta4) -
mc4*theta1dot*theta2dot*yc4*zc4*sin(theta2 + theta3 + theta4);
oe_v3 = (Iy4*theta1dot^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
(Ix4*theta1dot^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
(Ix3*theta1dot^2*sin(2*theta2 + 2*theta3))/2 + (Iy3*theta1dot^2*sin(2*theta2 +
2*theta3))/2 + (mc4*theta1dot^2*xc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
(mc4*theta1dot^2*yc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 +
(l3^2*mc4*theta1dot^2*sin(2*theta2 + 2*theta3))/2 +
(mc3*theta1dot^2*xc3^2*sin(2*theta2 + 2*theta3))/2 -
(mc3*theta1dot^2*yc3^2*sin(2*theta2 + 2*theta3))/2 +
(l2*mc4*theta1dot^2*yc4*cos(2*theta2 + theta3 + theta4))/2 +
(l2*mc4*theta1dot^2*xc4*sin(2*theta2 + theta3 + theta4))/2 +
(l2*mc4*theta1dot^2*yc4*cos(theta3 + theta4))/2 +
l2*mc4*theta2dot^2*yc4*cos(theta3 + theta4) + (l2*mc4*theta1dot^2*xc4*sin(theta3 +
theta4))/2 + l2*mc4*theta2dot^2*xc4*sin(theta3 + theta4) +
(l2*mc3*theta1dot^2*yc3*cos(theta3))/2 + l2*mc3*theta2dot^2*yc3*cos(theta3) -
l3*mc4*theta4dot^2*yc4*cos(theta4) + (l2*l3*mc4*theta1dot^2*sin(theta3))/2 +
l2*l3*mc4*theta2dot^2*sin(theta3) + l3*mc4*theta1dot^2*yc4*cos(2*theta2 + 2*theta3
+ theta4) + (l2*mc3*theta1dot^2*xc3*sin(theta3))/2 +
l2*mc3*theta2dot^2*xc3*sin(theta3) - l3*mc4*theta4dot^2*xc4*sin(theta4) +
l3*mc4*theta1dot^2*xc4*sin(2*theta2 + 2*theta3 + theta4) +
(l2*mc3*theta1dot^2*yc3*cos(2*theta2 + theta3))/2 +
(l2*l3*mc4*theta1dot^2*sin(2*theta2 + theta3))/2 +
(l2*mc3*theta1dot^2*xc3*sin(2*theta2 + theta3))/2 +
mc4*theta1dot^2*xc4*yc4*cos(2*theta2 + 2*theta3 + 2*theta4) +
mc3*theta1dot^2*xc3*yc3*cos(2*theta2 + 2*theta3) +
l3*mc4*theta1dot*theta2dot*zc4*cos(theta2 + theta3) +
mc3*theta1dot*theta2dot*xc3*zc3*cos(theta2 + theta3) -
mc3*theta1dot*theta2dot*yc3*zc3*sin(theta2 + theta3) -
2*l3*mc4*theta2dot*theta4dot*yc4*cos(theta4) -
2*l3*mc4*theta3dot*theta4dot*yc4*cos(theta4) -
2*l3*mc4*theta2dot*theta4dot*xc4*sin(theta4) -
2*l3*mc4*theta3dot*theta4dot*xc4*sin(theta4) +
mc4*theta1dot*theta2dot*xc4*zc4*cos(theta2 + theta3 + theta4) -
mc4*theta1dot*theta2dot*yc4*zc4*sin(theta2 + theta3 + theta4);
oe_v4 = (Iy4*theta1dot^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
(Ix4*theta1dot^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 +
(mc4*theta1dot^2*xc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 -
(mc4*theta1dot^2*yc4^2*sin(2*theta2 + 2*theta3 + 2*theta4))/2 +
(l2*mc4*theta1dot^2*yc4*cos(2*theta2 + theta3 + theta4))/2 +
```

```
(l2*mc4*theta1dot^2*xc4*sin(2*theta2 + theta3 + theta4))/2 +
(l2*mc4*theta1dot^2*yc4*cos(theta3 + theta4))/2 +
l2*mc4*theta2dot^2*yc4*cos(theta3 + theta4) + (l2*mc4*theta1dot^2*xc4*sin(theta3 +
theta4))/2 + l2*mc4*theta2dot^2*xc4*sin(theta3 + theta4) +
(l3*mc4*theta1dot^2*yc4*cos(theta4))/2 + l3*mc4*theta2dot^2*yc4*cos(theta4) +
l3*mc4*theta3dot^2*yc4*cos(theta4) + (l3*mc4*theta1dot^2*yc4*cos(2*theta2 +
2*theta3 + theta4))/2 + (l3*mc4*theta1dot^2*xc4*sin(theta4))/2 +
l3*mc4*theta2dot^2*xc4*sin(theta4) + l3*mc4*theta3dot^2*xc4*sin(theta4) +
(l3*mc4*theta1dot^2*xc4*sin(2*theta2 + 2*theta3 + theta4))/2 +
mc4*theta1dot^2*xc4*yc4*cos(2*theta2 + 2*theta3 + 2*theta4) +
2*l3*mc4*theta2dot*theta3dot*yc4*cos(theta4) +
2*l3*mc4*theta2dot*theta3dot*xc4*sin(theta4) +
mc4*theta1dot*theta2dot*xc4*zc4*cos(theta2 + theta3 + theta4) -
mc4*theta1dot*theta2dot*yc4*zc4*sin(theta2 + theta3 + theta4);
oe_v = [oe_v1;oe_v2;oe_v3;oe_v4];

% gravitational compensation
g_v1 = -g*mc2*(xc2*cos(theta2) - yc2*sin(theta2));
g_v2 = g*(mc4*xc4*cos(theta2 + theta3 + theta4) - mc3*yc3*sin(theta2 + theta3) -
mc4*yc4*sin(theta2 + theta3 + theta4) + l2*mc4*cos(theta2 + theta4) +
l3*mc4*cos(theta2 + theta3) + mc3*xc3*cos(theta2 + theta3));
g_v3 = g*l3*mc4*cos(theta2 + theta3) + g*mc3*xc3*cos(theta2 + theta3) -
g*mc3*yc3*sin(theta2 + theta3) + g*mc4*xc4*cos(theta2 + theta3 + theta4) -
g*mc4*yc4*sin(theta2 + theta3 + theta4);
g_v4 = g*mc4*xc4*cos(theta2 + theta3 + theta4) - g*mc4*yc4*sin(theta2 + theta3 +
theta4);
g_v = [g_v1;g_v2;g_v3;g_v4];
```

Open-Loop Dynamic Control

The dynamic model was used for open loop dynamic control of the robot. Open loop dynamic control was simulated using MATLAB and the joint angle error was plotted against the time.

```
clc; close all; clear all;
%% open loop dynamic control
% simulation parameters
dt = 0.01;
ts = 20;
t = 0:dt:ts;

% initial conditions
q(:,1) = [0;0;0;0];
q_dot(:,1) = [0;0;0;0];

% physical parameters of manipulator
d1 = 0.250;
l2 = 0.200;
l3 = 0.150;
```

```
d5 = 0.080;

xc1 = -8.674e-3; yc1 = 0; zc1 = 94.921e-3;
Ix1 = 6.112e-9; Iy1 = 4.879e-9; Iz1 = 5.331e-9;
mc1 = 8.71;

xc2 = -1.667e-3; yc2 = -3.852e-3; zc2 = 3.638e-3;
Ix2 = 3e-9; Iy2 = 4.486e-9; Iz2 = 3.513e-9;
mc2 = 1.53;

xc3 = 124.38e-3; yc3 = -4.295e-3; zc3 = 0;
Ix3 = 4.64e-9; Iy3 = 6.338e-9; Iz3 = 2.143e-9;
mc3 = 6.8;

xc4 = 74.262e-3; yc4 = 4.295e-3; zc4 = -0.642e-3;
Ix4 = 2.41e-9; Iy4 = 8.08e-9; Iz4 = 5.99e-9;
mc4 = 1.07;

% gravity
g = 9.81;
% damping coeff
b1 = 0.5; b2 = 0.5; b3 = 0.5; b4 = 0.5;
c1 = 0.6; c2 = 0.6; c3 = 0.6; c4 = 0.6;

% inverse dynamics (configuration space)
for i = 1:length(t)
% desired joint positions, velocities, accelerations
q_des(:,i) = [-pi/6;0*pi/3;pi/6*sin(0.2*t(i));pi/3*sin(0.1*t(i))];
q(:,1) = q_des(:,1); % initial conditions joint pose
q_dot_des(:,i) = [0;0;0.2*pi/6*cos(0.2*t(i));0.1*pi/3*cos(0.01*t(i))];
q_dot(:,1) = q_dot_des(:,1); % initial conditions joint angular
velocities
q_ddot_des(:,i) =
[0;0;-0.2*0.2*pi/6*sin(0.2*t(i));-0.1*0.1*pi/3*sin(0.1*t(i))];

% q_des(:,i) = [0;0;0;0];
% q(:,1) = q_des(:,1); % initial conditions joint pose
% q_dot_des(:,i) = [0;0;0;0];
% q_dot(:,1) = q_dot_des(:,1); % initial conditions joint angular
velocities
% q_ddot_des(:,i) = [0;0;0;0];

m_11 = (mc4*l2^2 + 2*mc4*l2*l3 + mc3*l2*xc3 + 2*mc4*l2*xc4 + mc4*l3^2 +
2*mc4*l3*xc4 + mc1*xc1^2 + mc3*xc3^2 + mc4*xc4^2 + mc1*yc1^2 -
mc2*zc2*yc1 + mc3*zc3^2 + mc4*zc4^2 + Iy2 + Iy3 + Iy4 + Iz1);
m_12 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_13 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
```

```
m_14 =  -(mc4*yc4*zc4);

m_21 = (mc2*xc1*xc2 + mc2*yc1*yc2 - mc3*yc3*zc3 - mc4*yc4*zc4);
m_22 = (mc4*l2^2 + 2*mc4*l2*l3 + mc3*l2*xc3 + 2*mc4*l2*xc4 + mc4*l3^2 +
2*mc4*l3*xc4 + mc3*xc3^2 + mc4*xc4^2 + mc3*yc3^2 + mc4*yc4^2 + Iz2 + Iz3
+ Iz4);
m_23 = (mc4*l3^2 + 2*mc4*l3*xc4 + l2*mc4*l3 + mc3*xc3^2 + mc4*xc4^2 +
l2*mc4*xc4 + mc3*yc3^2 + mc4*yc4^2 + Iz3 + Iz4);
m_24 = (Iz4 + mc4*xc4^2 + mc4*yc4^2 + l2*mc4*xc4 + l3*mc4*xc4);

m_31 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_32 = (mc4*l3^2 + 2*mc4*l3*xc4 + l2*mc4*l3 + mc3*xc3^2 + l2*mc3*xc3 +
mc4*xc4^2 + l2*mc4*xc4 + mc3*yc3^2 + mc4*yc4^2 + Iz3 + Iz4);
m_33 = (mc4*l3^2 + 2*mc4*l3*xc4 + mc3*xc3^2 + mc4*xc4^2 + mc3*yc3^2 +
mc4*yc4^2 + Iz3 + Iz4);
m_34 = (mc4*xc4^2 + l3*mc4*xc4 + mc4*yc4^2 + Iz4);

m_41 = -(mc4*yc4*zc4);
m_42 = (mc4*yc4^2+Iz4+mc4*xc4*(l2+l3+xc4));
m_43 = (mc4*xc4^2 + l3*mc4*xc4 + mc4*yc4^2 + Iz4);
m_44 = (mc4*xc4^2 + mc4*yc4^2 + Iz4);

M = [m_11, m_12, m_13, m_14;
    m_21, m_22, m_23, m_24;
    m_31, m_32, m_33, m_34;
    m_41, m_42, m_43, m_44];

% other effects
oe_v1 = (Ix3*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + Ix3*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) +
2*q_des(3,i)) - (Iy3*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 - Iy3*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) +
2*q_des(3,i)) + (Ix2*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i)))/2
- (Iy2*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i)))/2 +
(Ix4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 + Ix4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) +
Ix4*q_dot_des(1,i)*q_dot_des(4,i)*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)) - (Iy4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 -
Iy4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)) - Iy4*q_dot_des(1,i)*q_dot_des(4,i)*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) -
(mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 -
mc3*q_dot_des(1,i)*q_dot_des(3,i)*xc3^2*sin(2*q_des(2,i) + 2*q_des(3,i))
+ (mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3^2*sin(2*q_des(2,i) +
```

```
2*q_des(3,i)))/2 +
mc3*q_dot_des(1,i)*q_dot_des(3,i)*yc3^2*sin(2*q_des(2,i) + 2*q_des(3,i))
- l3*mc4*q_dot_des(2,i)^2*zc4*cos(q_des(2,i) + q_des(3,i)) -
l3*mc4*q_dot_des(3,i)^2*zc4*cos(q_des(2,i) + q_des(3,i)) -
mc3*q_dot_des(2,i)^2*xc3*zc3*cos(q_des(2,i) + q_des(3,i)) -
mc3*q_dot_des(3,i)^2*xc3*zc3*cos(q_des(2,i) + q_des(3,i)) +
mc3*q_dot_des(2,i)^2*yc3*zc3*sin(q_des(2,i) + q_des(3,i)) +
mc3*q_dot_des(3,i)^2*yc3*zc3*sin(q_des(2,i) + q_des(3,i)) -
l2*mc3*q_dot_des(1,i)^2*zc3*cos(q_des(2,i)) -
l2*mc3*q_dot_des(2,i)^2*zc3*cos(q_des(2,i)) -
l2*mc4*q_dot_des(2,i)^2*zc4*cos(q_des(2,i)) -
mc2*q_dot_des(1,i)^2*xc1*zc2*cos(q_des(2,i)) +
mc2*q_dot_des(1,i)^2*yc1*zc2*sin(q_des(2,i)) -
mc4*q_dot_des(2,i)^2*xc4*zc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*q_dot_des(3,i)^2*xc4*zc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*q_dot_des(4,i)^2*xc4*zc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) +
mc4*q_dot_des(2,i)^2*yc4*zc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)) +
mc4*q_dot_des(3,i)^2*yc4*zc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)) +
mc4*q_dot_des(4,i)^2*yc4*zc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
(l2^2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i)))/2 -
(mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 -
mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) -
mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) +
(mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 +
mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) +
mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) -
(l3^2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 -
l3^2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) + 2*q_des(3,i))
- mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*yc3*cos(2*q_des(2,i) +
2*q_des(3,i)) -
2*mc3*q_dot_des(1,i)*q_dot_des(3,i)*xc3*yc3*cos(2*q_des(2,i) +
2*q_des(3,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*cos(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4*cos(2*q_des(2,i)
+ q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*sin(2*q_des(2,i)
+ q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(2*q_des(2,i)
```

```
+ q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4*cos(q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(q_des(3,i) + q_des(4,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*zc4*cos(q_des(2,i) + q_des(3,i))
- 2*mc3*q_dot_des(2,i)*q_dot_des(3,i)*xc3*zc3*cos(q_des(2,i) +
q_des(3,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*sin(q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(q_des(3,i) +
q_des(4,i)) + 2*mc3*q_dot_des(2,i)*q_dot_des(3,i)*yc3*zc3*sin(q_des(2,i)
+ q_des(3,i)) +
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*cos(q_des(3,i)))/2 -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
l2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*sin(q_des(3,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*cos(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) -
2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*sin(q_des(3,i)))/2 -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*sin(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) -
2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*sin(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) -
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*cos(2*q_des(2,i) +
q_des(3,i)))/2 -
l2*l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) + q_des(3,i)) -
l2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) + q_des(3,i)) -
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*sin(2*q_des(2,i) +
q_des(3,i)))/2 -
2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) -
2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) -
2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) +
2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*yc4*zc4*sin(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) +
2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*zc4*sin(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) +
2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*zc4*sin(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) -
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) -
2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*yc4*cos(2*q_des(2,i) +
```

```
2*q_des(3,i) + 2*q_des(4,i)) -
2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i));
oe_v2 = (Iy2*q_dot_des(1,i)^2*sin(2*q_des(2,i)))/2 -
(Ix2*q_dot_des(1,i)^2*sin(2*q_des(2,i)))/2 -
(Ix4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) + 2*q_des(4,i)))/2
+ (Iy4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 - (Ix3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (Iy3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (l2^2*mc4*q_dot_des(1,i)^2*sin(q_des(4,i)))/2 +
l2^2*mc4*q_dot_des(2,i)^2*sin(q_des(4,i)) +
(l2^2*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + q_des(4,i)))/2 +
mc2*q_dot_des(1,i)^2*xc1*yc2 - mc2*q_dot_des(1,i)^2*xc2*yc1 +
(mc4*q_dot_des(1,i)^2*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 - (mc4*q_dot_des(1,i)^2*yc4^2*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 +
(l3^2*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i)))/2 +
(mc3*q_dot_des(1,i)^2*xc3^2*sin(2*q_des(2,i) + 2*q_des(3,i)))/2 -
(mc3*q_dot_des(1,i)^2*yc3^2*sin(2*q_des(2,i) + 2*q_des(3,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*l3*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
q_des(3,i) + q_des(4,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)) + (l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*yc3*cos(q_des(3,i)))/2 +
l2*mc3*q_dot_des(2,i)^2*yc3*cos(q_des(3,i)) -
(l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i)))/2 -
l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i)) -
l2*mc4*q_dot_des(3,i)^2*yc4*cos(q_des(3,i)) -
l2*mc4*q_dot_des(4,i)^2*yc4*cos(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*yc4*cos(q_des(4,i)) +
(l2*l3*mc4*q_dot_des(1,i)^2*sin(q_des(3,i)))/2 +
l2*l3*mc4*q_dot_des(2,i)^2*sin(q_des(3,i)) +
(l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + q_des(3,i) +
2*q_des(4,i)))/2 + l3*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)^2*xc3*sin(q_des(3,i)))/2 +
l2*mc3*q_dot_des(2,i)^2*xc3*sin(q_des(3,i)) -
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i)))/2 -
l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i)) -
l2*mc4*q_dot_des(3,i)^2*xc4*sin(q_des(3,i)) -
l2*mc4*q_dot_des(4,i)^2*xc4*sin(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*xc4*sin(q_des(4,i)) +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
```

```
2*q_des(4,i)))/2 + l3*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)^2*yc3*cos(2*q_des(2,i) + q_des(3,i)))/2 +
(l2*l3*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + q_des(3,i)))/2 -
(l2*l3*mc4*q_dot_des(1,i)^2*sin(q_des(3,i) - q_des(4,i)))/2 -
l2*l3*mc4*q_dot_des(2,i)^2*sin(q_des(3,i) - q_des(4,i)) -
l2*l3*mc4*q_dot_des(3,i)^2*sin(q_des(3,i) - q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)^2*xc3*sin(2*q_des(2,i) + q_des(3,i)))/2 +
mc4*q_dot_des(1,i)^2*xc4*yc4*cos(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)) + mc3*q_dot_des(1,i)^2*xc3*yc3*cos(2*q_des(2,i) +
2*q_des(3,i)) + l2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*zc4*cos(q_des(2,i)
+ q_des(4,i)) + l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*zc4*cos(q_des(2,i)
+ q_des(3,i)) + mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*zc3*cos(q_des(2,i)
+ q_des(3,i)) - mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*zc3*sin(q_des(2,i)
+ q_des(3,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*yc4*cos(q_des(3,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*cos(q_des(3,i)) -
2*l2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*cos(q_des(3,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*xc4*sin(q_des(3,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*sin(q_des(3,i)) -
2*l2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*sin(q_des(3,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
2*l2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*sin(q_des(3,i) - q_des(4,i)) +
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i) +
q_des(4,i)) - mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*zc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i));
oe_v3 = (Iy4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 - (Ix4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 - (Ix3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (Iy3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (mc4*q_dot_des(1,i)^2*xc4^2*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 -
(mc4*q_dot_des(1,i)^2*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 + (l3^2*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (mc3*q_dot_des(1,i)^2*xc3^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 - (mc3*q_dot_des(1,i)^2*yc3^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) +
q_des(3,i) + q_des(4,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)) + (l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*yc3*cos(q_des(3,i)))/2 +
```

```
l2*mc3*q_dot_des(2,i)^2*yc3*cos(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*yc4*cos(q_des(4,i)) +
(l2*l3*mc4*q_dot_des(1,i)^2*sin(q_des(3,i)))/2 +
l2*l3*mc4*q_dot_des(2,i)^2*sin(q_des(3,i)) +
l3*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*xc3*sin(q_des(3,i)))/2 +
l2*mc3*q_dot_des(2,i)^2*xc3*sin(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*xc4*sin(q_des(4,i)) +
l3*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*yc3*cos(2*q_des(2,i) +
q_des(3,i)))/2 + (l2*l3*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
q_des(3,i)))/2 + (l2*mc3*q_dot_des(1,i)^2*xc3*sin(2*q_des(2,i) +
q_des(3,i)))/2 + mc4*q_dot_des(1,i)^2*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) +
mc3*q_dot_des(1,i)^2*xc3*yc3*cos(2*q_des(2,i) + 2*q_des(3,i)) +
l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*zc4*cos(q_des(2,i) + q_des(3,i)) +
mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*zc3*cos(q_des(2,i) + q_des(3,i)) -
mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*zc3*sin(q_des(2,i) + q_des(3,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) +
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i) +
q_des(4,i)) - mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*zc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i));
oe_v4 = (Iy4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 - (Ix4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 + (mc4*q_dot_des(1,i)^2*xc4^2*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 -
(mc4*q_dot_des(1,i)^2*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) +
q_des(3,i) + q_des(4,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)) + (l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)) + (l3*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(4,i)))/2 +
l3*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(4,i)) +
l3*mc4*q_dot_des(3,i)^2*yc4*cos(q_des(4,i)) +
(l3*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)))/2 + (l3*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(4,i)))/2 +
l3*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(4,i)) +
l3*mc4*q_dot_des(3,i)^2*xc4*sin(q_des(4,i)) +
(l3*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)))/2 + mc4*q_dot_des(1,i)^2*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) +
```

```
2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*yc4*cos(q_des(4,i)) +
2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*xc4*sin(q_des(4,i)) +
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i) +
q_des(4,i)) - mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*zc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i));

oe_v = [oe_v1;oe_v2;oe_v3;oe_v4];

% gravitational compensation
g_v1 = -g*mc2*(xc2*cos(q_des(2,i)) - yc2*sin(q_des(2,i)));
g_v2 = g*(mc4*xc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc3*yc3*sin(q_des(2,i) + q_des(3,i)) - mc4*yc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i)) + l2*mc4*cos(q_des(2,i) + q_des(4,i)) +
l3*mc4*cos(q_des(2,i) + q_des(3,i)) + mc3*xc3*cos(q_des(2,i) +
q_des(3,i)));
g_v3 = g*(l3*mc4*cos(q_des(2,i) + q_des(3,i)) + mc3*xc3*cos(q_des(2,i) +
q_des(3,i)) - mc3*yc3*sin(q_des(2,i) + q_des(3,i)) +
mc4*xc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*yc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)));
g_v4 = g*(mc4*xc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*yc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)));

g_v = [g_v1;g_v2;g_v3;g_v4];

% joint friction
fr =
[b1*q_dot(1,i)+c1*sign(q_dot(1,i));b2*q_dot(2,i)+c2*sign(q_dot(2,i));b3*
q_dot(3,i)+c3*sign(q_dot(3,i));b4*q_dot(4,i)+c4*sign(q_dot(4,i))];

% desired input torque
tau_des(:,i) = M*(q_ddot_des(:,i)) + g_v + oe_v;
% tau_des(:,i) = [0;0;0;0];

q_ddot(:,i) = inv(M)*(tau_des(:,i) - (g_v + oe_v + 0*fr));
q_dot(:,i+1) = q_dot(:,i) + dt*q_ddot(:,i);
q(:,i+1) = q(:,i) + dt*q_dot(:,i) + (1/2)*dt*dt*(q_ddot(:,i));

end


% inverse dynamics animation
figure

for i = 1:length(t)

x0 = zeros(1,i);
y0 = zeros(1,i);
```

```
z0 = zeros(1,i);

x1 = zeros(1,i);
y1 = zeros(1,i);
z1 = d1.*ones(1,i);

x2 = zeros(1,i);
y2 = zeros(1,i);
z2 = d1.*ones(1,i);

x3 = l2*cos(q(1,i))*cos(q(2,i));
y3 = l2*cos(q(2,i))*sin(q(1,i));
z3 = l2*sin(q(2,i)) + d1;

x4 = l2*cos(q(1,i))*cos(q(2,i)) - l3*cos(q(1,i))*sin(q(2,i))*sin(q(3,i))
+ l3*cos(q(1,i))*cos(q(2,i))*cos(q(3,i));
y4 = l2*cos(q(2,i))*sin(q(1,i)) - l3*sin(q(1,i))*sin(q(2,i))*sin(q(3,i))
+ l3*cos(q(2,i))*cos(q(3,i))*sin(q(1,i));
z4 = l2*sin(q(2,i)) + l3*cos(q(2,i))*sin(q(3,i)) +
l3*cos(q(3,i))*sin(q(2,i)) + d1;

x5 = cos(q(1,i))*(l3*cos(q(2,i) + q(3,i)) + l2*cos(q(2,i))) +
d5*sin(q(2,i) + q(3,i) + q(4,i))*cos(q(1,i));
y5 = sin(q(1,i))*(l3*cos(q(2,i) + q(3,i)) + l2*cos(q(2,i))) +
d5*sin(q(2,i) + q(3,i) + q(4,i))*sin(q(1,i));
z5 = d1 + l3*sin(q(2,i) + q(3,i)) + l2*sin(q(2,i)) - d5*cos(q(2,i) +
q(3,i) + q(4,i));

x_des(1,i) = cos(q_des(1,i))*(l3*cos(q_des(2,i) + q_des(3,i)) +
l2*cos(q_des(2,i))) + d5*sin(q_des(2,i) + q_des(3,i) +
q_des(4,i))*cos(q_des(1,i));
y_des(1,i) = sin(q_des(1,i))*(l3*cos(q_des(2,i) + q_des(3,i)) +
l2*cos(q_des(2,i))) + d5*sin(q_des(2,i) + q_des(3,i) +
q_des(4,i))*sin(q_des(1,i));
z_des(1,i) = d1 + l3*sin(q_des(2,i) + q_des(3,i)) + l2*sin(q_des(2,i)) -
d5*cos(q_des(2,i) + q_des(3,i) + q_des(4,i));


plot3(x_des, y_des, z_des, 'b-',linewidth=1)
hold on;
plot3([x0,x1,x2,x3,x4,x5],[y0,y1,y2,y3,y4,y5],[z0,z1,z2,z3,z4,z5],'r-o',
LineWidth=1)
hold on;
% view([94 22]);
grid on;
xlim([-0.600,0.600]);
ylim([-0.600,0.600]);
```

```
zlim([0,0.600]);
title("open loop dynamic control")
pause(0.001);
hold off;

end
% error plot
figure
plot(t, q_des(:,1:i)-q(:,1:i));
legend("err theta1", "err theta2", "err theta3", "err theta4")
xlabel("time[s]"); ylabel("error[rad]");
title("Joint Angle Errors (Open Loop)")
grid on;
```

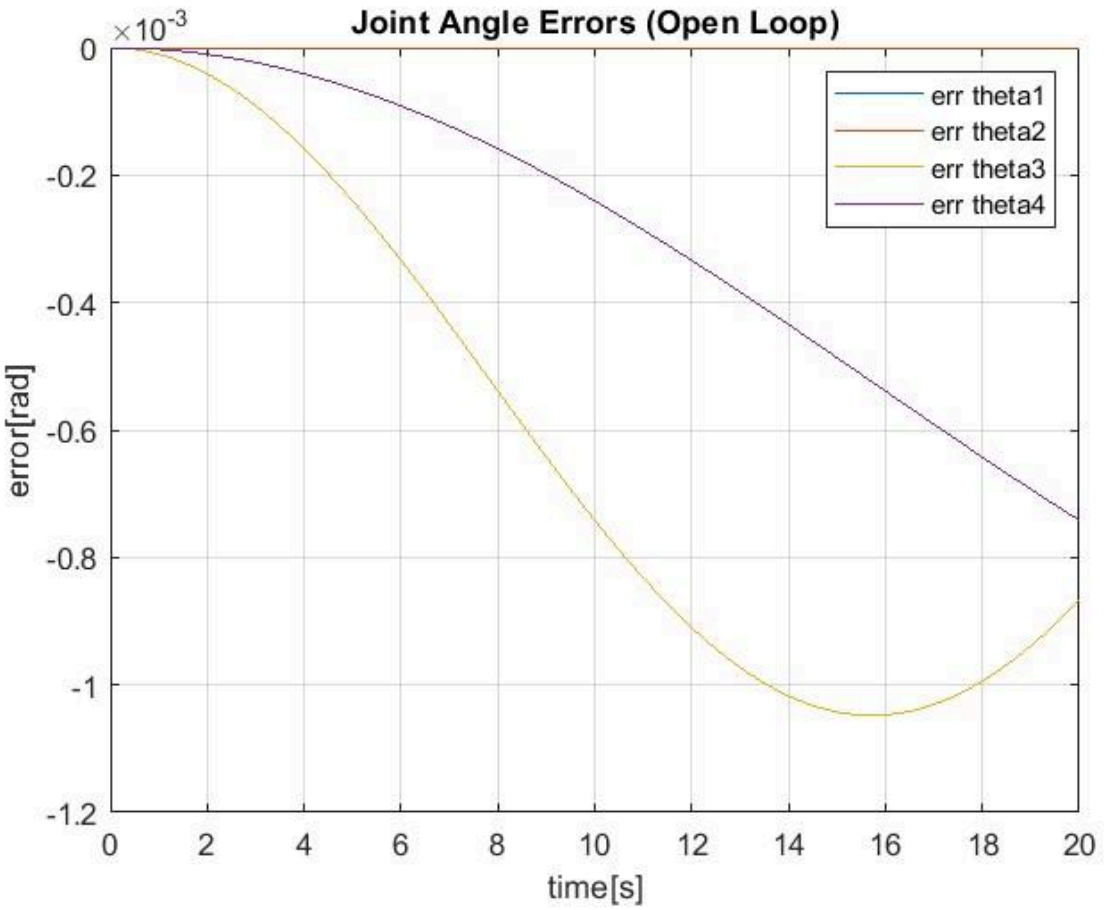

Fig.6 open loop joint angle error plot

Closed-loop Dynamics

A closed loop dynamics simulation was done which updates the control input(tau: Joint torques) based on the current location found in the simulation using euler integration.

```
clc; close all; clear all;
%% open loop dynamic control (configuration space control)
% simulation parameters
dt = 0.01;
ts = 20;
t = 0:dt:ts;

% initial conditions
q(:,1) = [0;0;0;0];
q_dot(:,1) = [0;0;0;0];

% physical parameters of manipulator
d1 = 0.250;
l2 = 0.200;
l3 = 0.150;
d5 = 0.080;

xc1 = -8.674e-3; yc1 = 0; zc1 = 94.921e-3;
Ix1 = 6.112e-9; Iy1 = 4.879e-9; Iz1 = 5.331e-9;
mc1 = 8.71;

xc2 = -1.667e-3; yc2 = -3.852e-3; zc2 = 3.638e-3;
Ix2 = 3e-9; Iy2 = 4.486e-9; Iz2 = 3.513e-9;
mc2 = 1.53;

xc3 = 124.38e-3; yc3 = -4.295e-3; zc3 = 0;
Ix3 = 4.64e-9; Iy3 = 6.338e-9; Iz3 = 2.143e-9;
mc3 = 6.8;

xc4 = 74.262e-3; yc4 = 4.295e-3; zc4 = -0.642e-3;
Ix4 = 2.41e-9; Iy4 = 8.08e-9; Iz4 = 5.99e-9;
mc4 = 1.07;

% gravity
g = 9.81;
% damping coeff
b1 = 0.5; b2 = 0.5; b3 = 0.5; b4 = 0.5;
c1 = 0.6; c2 = 0.6; c3 = 0.6; c4 = 0.6;

% controller gains
kp = 2;
% kd = 1.0;
kd = 2*sqrt(kp);

% inverse dynamics (configuration space)
```

```
for i = 1:length(t)
% desired joint positions, velocities, accelerations
q_des(:,i) = [-pi/6;0*pi/3;pi/6*sin(0.2*t(i));pi/3*sin(0.1*t(i))];
q_dot_des(:,i) = [0;0;0.2*pi/6*cos(0.2*t(i));0.1*pi/3*cos(0.01*t(i))];
q_ddot_des(:,i) =
[0;0;-0.2*0.2*pi/6*sin(0.2*t(i));-0.1*0.1*pi/3*sin(0.1*t(i))];

% set-point
% q_des(:,i) = [pi/2;0;0;0];
% q_dot_des(:,i) = [0;0;0;0];
% q_ddot_des(:,i) = [0;0;0;0];

m_11 = (mc4*l2^2 + 2*mc4*l2*l3 + mc3*l2*xc3 + 2*mc4*l2*xc4 + mc4*l3^2 +
2*mc4*l3*xc4 + mc1*xc1^2 + mc3*xc3^2 + mc4*xc4^2 + mc1*yc1^2 -
mc2*zc2*yc1 + mc3*zc3^2 + mc4*zc4^2 + Iy2 + Iy3 + Iy4 + Iz1);
m_12 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_13 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_14 =  -(mc4*yc4*zc4);

m_21 = (mc2*xc1*xc2 + mc2*yc1*yc2 - mc3*yc3*zc3 - mc4*yc4*zc4);
m_22 = (mc4*l2^2 + 2*mc4*l2*l3 + mc3*l2*xc3 + 2*mc4*l2*xc4 + mc4*l3^2 +
2*mc4*l3*xc4 + mc3*xc3^2 + mc4*xc4^2 + mc3*yc3^2 + mc4*yc4^2 + Iz2 + Iz3
+ Iz4);
m_23 = (mc4*l3^2 + 2*mc4*l3*xc4 + l2*mc4*l3 + mc3*xc3^2 + mc4*xc4^2 +
l2*mc4*xc4 + mc3*yc3^2 + mc4*yc4^2 + Iz3 + Iz4);
m_24 = (Iz4 + mc4*xc4^2 + mc4*yc4^2 + l2*mc4*xc4 + l3*mc4*xc4);

m_31 = -(mc3*yc3*zc3 + mc4*yc4*zc4);
m_32 = (mc4*l3^2 + 2*mc4*l3*xc4 + l2*mc4*l3 + mc3*xc3^2 + l2*mc3*xc3 +
mc4*xc4^2 + l2*mc4*xc4 + mc3*yc3^2 + mc4*yc4^2 + Iz3 + Iz4);
m_33 = (mc4*l3^2 + 2*mc4*l3*xc4 + mc3*xc3^2 + mc4*xc4^2 + mc3*yc3^2 +
mc4*yc4^2 + Iz3 + Iz4);
m_34 = (mc4*xc4^2 + l3*mc4*xc4 + mc4*yc4^2 + Iz4);

m_41 = -(mc4*yc4*zc4);
m_42 = (mc4*yc4^2+Iz4+mc4*xc4*(l2+l3+xc4));
m_43 = (mc4*xc4^2 + l3*mc4*xc4 + mc4*yc4^2 + Iz4);
m_44 = (mc4*xc4^2 + mc4*yc4^2 + Iz4);

M = [m_11, m_12, m_13, m_14;
    m_21, m_22, m_23, m_24;
    m_31, m_32, m_33, m_34;
    m_41, m_42, m_43, m_44];

% other effects
oe_v1 = (Ix3*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + Ix3*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) +
```

```
2*q_des(3,i)) - (Iy3*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 - Iy3*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) +
2*q_des(3,i)) + (Ix2*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i)))/2
- (Iy2*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i)))/2 +
(Ix4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 + Ix4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) +
Ix4*q_dot_des(1,i)*q_dot_des(4,i)*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)) - (Iy4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 -
Iy4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)) - Iy4*q_dot_des(1,i)*q_dot_des(4,i)*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) -
(mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 -
mc3*q_dot_des(1,i)*q_dot_des(3,i)*xc3^2*sin(2*q_des(2,i) + 2*q_des(3,i))
+ (mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 +
mc3*q_dot_des(1,i)*q_dot_des(3,i)*yc3^2*sin(2*q_des(2,i) + 2*q_des(3,i))
- l3*mc4*q_dot_des(2,i)^2*zc4*cos(q_des(2,i) + q_des(3,i)) -
l3*mc4*q_dot_des(3,i)^2*zc4*cos(q_des(2,i) + q_des(3,i)) -
mc3*q_dot_des(2,i)^2*xc3*zc3*cos(q_des(2,i) + q_des(3,i)) -
mc3*q_dot_des(3,i)^2*xc3*zc3*cos(q_des(2,i) + q_des(3,i)) +
mc3*q_dot_des(2,i)^2*yc3*zc3*sin(q_des(2,i) + q_des(3,i)) +
mc3*q_dot_des(3,i)^2*yc3*zc3*sin(q_des(2,i) + q_des(3,i)) -
l2*mc3*q_dot_des(1,i)^2*zc3*cos(q_des(2,i)) -
l2*mc3*q_dot_des(2,i)^2*zc3*cos(q_des(2,i)) -
l2*mc4*q_dot_des(2,i)^2*zc4*cos(q_des(2,i)) -
mc2*q_dot_des(1,i)^2*xc1*zc2*cos(q_des(2,i)) +
mc2*q_dot_des(1,i)^2*yc1*zc2*sin(q_des(2,i)) -
mc4*q_dot_des(2,i)^2*xc4*zc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*q_dot_des(3,i)^2*xc4*zc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*q_dot_des(4,i)^2*xc4*zc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) +
mc4*q_dot_des(2,i)^2*yc4*zc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)) +
mc4*q_dot_des(3,i)^2*yc4*zc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)) +
mc4*q_dot_des(4,i)^2*yc4*zc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
(l2^2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i)))/2 -
(mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 -
mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) -
mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) +
(mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 +
mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) +
```

```
mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)) -
(l3^2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 -
l3^2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) + 2*q_des(3,i))
- mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*yc3*cos(2*q_des(2,i) +
2*q_des(3,i)) -
2*mc3*q_dot_des(1,i)*q_dot_des(3,i)*xc3*yc3*cos(2*q_des(2,i) +
2*q_des(3,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*cos(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4*cos(2*q_des(2,i)
+ q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*sin(2*q_des(2,i)
+ q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(2*q_des(2,i)
+ q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4*cos(q_des(3,i) + q_des(4,i)) -
l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(q_des(3,i) + q_des(4,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*zc4*cos(q_des(2,i) + q_des(3,i))
- 2*mc3*q_dot_des(2,i)*q_dot_des(3,i)*xc3*zc3*cos(q_des(2,i) +
q_des(3,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*sin(q_des(3,i) +
q_des(4,i)) - l2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(q_des(3,i) +
q_des(4,i)) + 2*mc3*q_dot_des(2,i)*q_dot_des(3,i)*yc3*zc3*sin(q_des(2,i)
+ q_des(3,i)) +
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*cos(q_des(3,i)))/2 -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
l2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*sin(q_des(3,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*cos(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) -
2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*yc4*cos(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*sin(q_des(3,i)))/2 -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*sin(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) -
2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*sin(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) -
l3*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*sin(2*q_des(2,i) + 2*q_des(3,i)
+ q_des(4,i)) -
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*cos(2*q_des(2,i) +
q_des(3,i)))/2 -
l2*l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*sin(2*q_des(2,i) + q_des(3,i)) -
l2*l3*mc4*q_dot_des(1,i)*q_dot_des(3,i)*sin(2*q_des(2,i) + q_des(3,i)) -
```

```
(l2*mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*sin(2*q_des(2,i) +
q_des(3,i)))/2 -
2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) -
2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) -
2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) +
2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*yc4*zc4*sin(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) +
2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*zc4*sin(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) +
2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*zc4*sin(q_des(2,i) + q_des(3,i)
+ q_des(4,i)) -
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) -
2*mc4*q_dot_des(1,i)*q_dot_des(3,i)*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) -
2*mc4*q_dot_des(1,i)*q_dot_des(4,i)*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i));
oe_v2 = (Iy2*q_dot_des(1,i)^2*sin(2*q_des(2,i)))/2 -
(Ix2*q_dot_des(1,i)^2*sin(2*q_des(2,i)))/2 -
(Ix4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) + 2*q_des(4,i)))/2
+ (Iy4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 - (Ix3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (Iy3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (l2^2*mc4*q_dot_des(1,i)^2*sin(q_des(4,i)))/2 +
l2^2*mc4*q_dot_des(2,i)^2*sin(q_des(4,i)) +
(l2^2*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + q_des(4,i)))/2 +
mc2*q_dot_des(1,i)^2*xc1*yc2 - mc2*q_dot_des(1,i)^2*xc2*yc1 +
(mc4*q_dot_des(1,i)^2*xc4^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 - (mc4*q_dot_des(1,i)^2*yc4^2*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 +
(l3^2*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i)))/2 +
(mc3*q_dot_des(1,i)^2*xc3^2*sin(2*q_des(2,i) + 2*q_des(3,i)))/2 -
(mc3*q_dot_des(1,i)^2*yc3^2*sin(2*q_des(2,i) + 2*q_des(3,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*l3*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
q_des(3,i) + q_des(4,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)) + (l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*yc3*cos(q_des(3,i)))/2 +
l2*mc3*q_dot_des(2,i)^2*yc3*cos(q_des(3,i)) -
(l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i)))/2 -
```

```
l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i)) -
l2*mc4*q_dot_des(3,i)^2*yc4*cos(q_des(3,i)) -
l2*mc4*q_dot_des(4,i)^2*yc4*cos(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*yc4*cos(q_des(4,i)) +
(l2*l3*mc4*q_dot_des(1,i)^2*sin(q_des(3,i)))/2 +
l2*l3*mc4*q_dot_des(2,i)^2*sin(q_des(3,i)) +
(l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + q_des(3,i) +
2*q_des(4,i)))/2 + l3*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)^2*xc3*sin(q_des(3,i)))/2 +
l2*mc3*q_dot_des(2,i)^2*xc3*sin(q_des(3,i)) -
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i)))/2 -
l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i)) -
l2*mc4*q_dot_des(3,i)^2*xc4*sin(q_des(3,i)) -
l2*mc4*q_dot_des(4,i)^2*xc4*sin(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*xc4*sin(q_des(4,i)) +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
2*q_des(4,i)))/2 + l3*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) +
2*q_des(3,i) + q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)^2*yc3*cos(2*q_des(2,i) + q_des(3,i)))/2 +
(l2*l3*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + q_des(3,i)))/2 -
(l2*l3*mc4*q_dot_des(1,i)^2*sin(q_des(3,i) - q_des(4,i)))/2 -
l2*l3*mc4*q_dot_des(2,i)^2*sin(q_des(3,i) - q_des(4,i)) -
l2*l3*mc4*q_dot_des(3,i)^2*sin(q_des(3,i) - q_des(4,i)) +
(l2*mc3*q_dot_des(1,i)^2*xc3*sin(2*q_des(2,i) + q_des(3,i)))/2 +
mc4*q_dot_des(1,i)^2*xc4*yc4*cos(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)) + mc3*q_dot_des(1,i)^2*xc3*yc3*cos(2*q_des(2,i) +
2*q_des(3,i)) + l2*mc4*q_dot_des(1,i)*q_dot_des(2,i)*zc4*cos(q_des(2,i)
+ q_des(4,i)) + l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*zc4*cos(q_des(2,i)
+ q_des(3,i)) + mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*zc3*cos(q_des(2,i)
+ q_des(3,i)) - mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*zc3*sin(q_des(2,i)
+ q_des(3,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*yc4*cos(q_des(3,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*cos(q_des(3,i)) -
2*l2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*cos(q_des(3,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(3,i)*xc4*sin(q_des(3,i)) -
2*l2*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*sin(q_des(3,i)) -
2*l2*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*sin(q_des(3,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
2*l2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*sin(q_des(3,i) - q_des(4,i)) +
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i) +
q_des(4,i)) - mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*zc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i));
oe_v3 = (Iy4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
```

```
2*q_des(4,i)))/2 - (Ix4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 - (Ix3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (Iy3*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (mc4*q_dot_des(1,i)^2*xc4^2*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 -
(mc4*q_dot_des(1,i)^2*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 + (l3^2*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (mc3*q_dot_des(1,i)^2*xc3^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 - (mc3*q_dot_des(1,i)^2*yc3^2*sin(2*q_des(2,i) +
2*q_des(3,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) +
q_des(3,i) + q_des(4,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)) + (l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*yc3*cos(q_des(3,i)))/2 +
l2*mc3*q_dot_des(2,i)^2*yc3*cos(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*yc4*cos(q_des(4,i)) +
(l2*l3*mc4*q_dot_des(1,i)^2*sin(q_des(3,i)))/2 +
l2*l3*mc4*q_dot_des(2,i)^2*sin(q_des(3,i)) +
l3*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*xc3*sin(q_des(3,i)))/2 +
l2*mc3*q_dot_des(2,i)^2*xc3*sin(q_des(3,i)) -
l3*mc4*q_dot_des(4,i)^2*xc4*sin(q_des(4,i)) +
l3*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)) + (l2*mc3*q_dot_des(1,i)^2*yc3*cos(2*q_des(2,i) +
q_des(3,i)))/2 + (l2*l3*mc4*q_dot_des(1,i)^2*sin(2*q_des(2,i) +
q_des(3,i)))/2 + (l2*mc3*q_dot_des(1,i)^2*xc3*sin(2*q_des(2,i) +
q_des(3,i)))/2 + mc4*q_dot_des(1,i)^2*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) +
mc3*q_dot_des(1,i)^2*xc3*yc3*cos(2*q_des(2,i) + 2*q_des(3,i)) +
l3*mc4*q_dot_des(1,i)*q_dot_des(2,i)*zc4*cos(q_des(2,i) + q_des(3,i)) +
mc3*q_dot_des(1,i)*q_dot_des(2,i)*xc3*zc3*cos(q_des(2,i) + q_des(3,i)) -
mc3*q_dot_des(1,i)*q_dot_des(2,i)*yc3*zc3*sin(q_des(2,i) + q_des(3,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*yc4*cos(q_des(4,i)) -
2*l3*mc4*q_dot_des(2,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) -
2*l3*mc4*q_dot_des(3,i)*q_dot_des(4,i)*xc4*sin(q_des(4,i)) +
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i) +
q_des(4,i)) - mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*zc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i));
oe_v4 = (Iy4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
2*q_des(4,i)))/2 - (Ix4*q_dot_des(1,i)^2*sin(2*q_des(2,i) + 2*q_des(3,i)
+ 2*q_des(4,i)))/2 + (mc4*q_dot_des(1,i)^2*xc4^2*sin(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)))/2 -
(mc4*q_dot_des(1,i)^2*yc4^2*sin(2*q_des(2,i) + 2*q_des(3,i) +
```

```
2*q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) +
q_des(3,i) + q_des(4,i)))/2 +
(l2*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + q_des(3,i) +
q_des(4,i)))/2 + (l2*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(3,i) +
q_des(4,i)) + (l2*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)))/2 + l2*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(3,i) +
q_des(4,i)) + (l3*mc4*q_dot_des(1,i)^2*yc4*cos(q_des(4,i)))/2 +
l3*mc4*q_dot_des(2,i)^2*yc4*cos(q_des(4,i)) +
l3*mc4*q_dot_des(3,i)^2*yc4*cos(q_des(4,i)) +
(l3*mc4*q_dot_des(1,i)^2*yc4*cos(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)))/2 + (l3*mc4*q_dot_des(1,i)^2*xc4*sin(q_des(4,i)))/2 +
l3*mc4*q_dot_des(2,i)^2*xc4*sin(q_des(4,i)) +
l3*mc4*q_dot_des(3,i)^2*xc4*sin(q_des(4,i)) +
(l3*mc4*q_dot_des(1,i)^2*xc4*sin(2*q_des(2,i) + 2*q_des(3,i) +
q_des(4,i)))/2 + mc4*q_dot_des(1,i)^2*xc4*yc4*cos(2*q_des(2,i) +
2*q_des(3,i) + 2*q_des(4,i)) +
2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*yc4*cos(q_des(4,i)) +
2*l3*mc4*q_dot_des(2,i)*q_dot_des(3,i)*xc4*sin(q_des(4,i)) +
mc4*q_dot_des(1,i)*q_dot_des(2,i)*xc4*zc4*cos(q_des(2,i) + q_des(3,i) +
q_des(4,i)) - mc4*q_dot_des(1,i)*q_dot_des(2,i)*yc4*zc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i));

oe_v = [oe_v1;oe_v2;oe_v3;oe_v4];

% gravitational compensation
g_v1 = -g*mc2*(xc2*cos(q_des(2,i)) - yc2*sin(q_des(2,i)));
g_v2 = g*(mc4*xc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc3*yc3*sin(q_des(2,i) + q_des(3,i)) - mc4*yc4*sin(q_des(2,i) +
q_des(3,i) + q_des(4,i)) + l2*mc4*cos(q_des(2,i) + q_des(4,i)) +
l3*mc4*cos(q_des(2,i) + q_des(3,i)) + mc3*xc3*cos(q_des(2,i) +
q_des(3,i)));
g_v3 = g*(l3*mc4*cos(q_des(2,i) + q_des(3,i)) + mc3*xc3*cos(q_des(2,i) +
q_des(3,i)) - mc3*yc3*sin(q_des(2,i) + q_des(3,i)) +
mc4*xc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*yc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)));
g_v4 = g*(mc4*xc4*cos(q_des(2,i) + q_des(3,i) + q_des(4,i)) -
mc4*yc4*sin(q_des(2,i) + q_des(3,i) + q_des(4,i)));

g_v = [g_v1;g_v2;g_v3;g_v4];

% joint friction
fr =
[b1*q_dot(1,i)+c1*sign(q_dot(1,i));b2*q_dot(2,i)+c2*sign(q_dot(2,i));b3*
q_dot(3,i)+c3*sign(q_dot(3,i));b4*q_dot(4,i)+c4*sign(q_dot(4,i))];

% desired input torque
```

```
tau_des(:,i) = M*((q_ddot_des(:,i))+ kp*(q_des(:,i)-q(:,i))+
kd*(q_dot_des(:,i)-q_dot(:,i))) + g_v + oe_v;
% tau_des(:,i) = [0;0;0;0];

q_ddot(:,i) = inv(M)*(tau_des(:,i) - (g_v + oe_v + 0*fr));
q_dot(:,i+1) = q_dot(:,i) + dt*q_ddot(:,i);
q(:,i+1) = q(:,i) + dt*q_dot(:,i) + (1/2)*dt*dt*(q_ddot(:,i));

end


% inverse dynamics animation
figure
for i = 1:length(t)

x0 = zeros(1,i);
y0 = zeros(1,i);
z0 = zeros(1,i);

x1 = zeros(1,i);
y1 = zeros(1,i);
z1 = d1.*ones(1,i);

x2 = zeros(1,i);
y2 = zeros(1,i);
z2 = d1.*ones(1,i);

x3 = l2*cos(q(1,i))*cos(q(2,i));
y3 = l2*cos(q(2,i))*sin(q(1,i));
z3 = l2*sin(q(2,i)) + d1;

x4 = l2*cos(q(1,i))*cos(q(2,i)) - l3*cos(q(1,i))*sin(q(2,i))*sin(q(3,i))
+ l3*cos(q(1,i))*cos(q(2,i))*cos(q(3,i));
y4 = l2*cos(q(2,i))*sin(q(1,i)) - l3*sin(q(1,i))*sin(q(2,i))*sin(q(3,i))
+ l3*cos(q(2,i))*cos(q(3,i))*sin(q(1,i));
z4 = l2*sin(q(2,i)) + l3*cos(q(2,i))*sin(q(3,i)) +
l3*cos(q(3,i))*sin(q(2,i)) + d1;

x5 = cos(q(1,i))*(l3*cos(q(2,i) + q(3,i)) + l2*cos(q(2,i))) +
d5*sin(q(2,i) + q(3,i) + q(4,i))*cos(q(1,i));
y5 = sin(q(1,i))*(l3*cos(q(2,i) + q(3,i)) + l2*cos(q(2,i))) +
d5*sin(q(2,i) + q(3,i) + q(4,i))*sin(q(1,i));
z5 = d1 + l3*sin(q(2,i) + q(3,i)) + l2*sin(q(2,i)) - d5*cos(q(2,i) +
q(3,i) + q(4,i));

x_des(1,i) = cos(q_des(1,i))*(l3*cos(q_des(2,i) + q_des(3,i)) +
l2*cos(q_des(2,i))) + d5*sin(q_des(2,i) + q_des(3,i) +
```

```
q_des(4,i))*cos(q_des(1,i));
y_des(1,i) = sin(q_des(1,i))*(l3*cos(q_des(2,i) + q_des(3,i)) +
l2*cos(q_des(2,i))) + d5*sin(q_des(2,i) + q_des(3,i) +
q_des(4,i))*sin(q_des(1,i));
z_des(1,i) = d1 + l3*sin(q_des(2,i) + q_des(3,i)) + l2*sin(q_des(2,i)) -
d5*cos(q_des(2,i) + q_des(3,i) + q_des(4,i));


plot3(x_des, y_des, z_des, 'b-',linewidth=1)
% plot3(x_des, y_des, z_des, 'bx')
hold on;
plot3([x0,x1,x2,x3,x4,x5],[y0,y1,y2,y3,y4,y5],[z0,z1,z2,z3,z4,z5],'r-o',
LineWidth=1)
hold on;
grid on;
xlim([-0.600,0.600]);
ylim([-0.600,0.600]);
zlim([0,0.600]);
title("Closed Loop Dynamic Control")
pause(0.01);
hold off;

end
% error plot
figure
plot(t, q_des(:,1:i)-q(:,1:i));
legend("err theta1", "err theta2", "err theta3", "err theta4")
xlabel("time[s]"); ylabel("error[rad]");
title("Joint Angle Errors (Closed Loop)")
grid on;
```

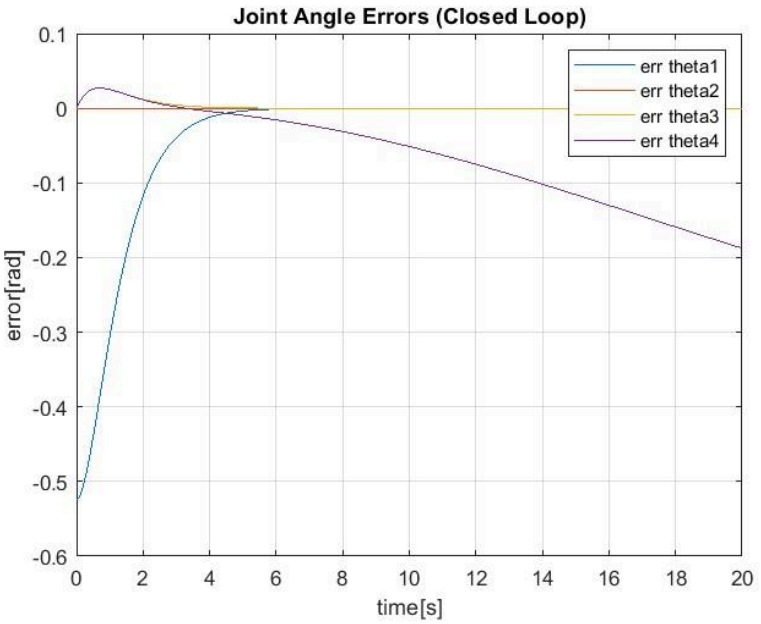

Fig.7 Closed loop joint angle error plot

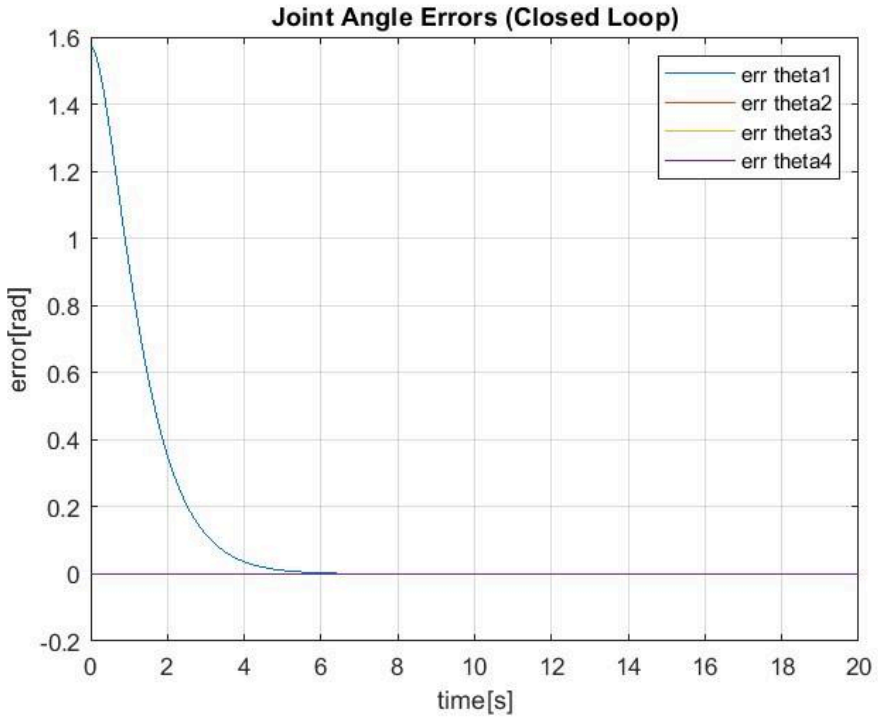

Fig 8. Joint angle error for set point control (closed loop)

---THE END---