PROBLEM 1

The equivalent homogenous transformation for these set of rotations is given by,

R = R1*R2*R3

where R1 = $\begin{bmatrix} \cos(45)^o & -\sin(45)^o & 0 & 0 \\ \sin(45)^o & \cos(45)^o & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

R2 = $\begin{bmatrix} \cos(30)^o & 0 & \sin(30)^o & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(30)^o & 0 & \cos(30)^o & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

R3 = $\begin{bmatrix} \cos(20)^o & -\sin(20)^o & 0 & 0 \\ \sin(20)^o & \cos(20)^o & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

```
clear all

R1 = [[cosd(45) -sind(45) 0 0];
      [sind(45) cosd(45) 0 0];
      [0 0 1 0];
      [0 0 0 1]];
R2 = [[cosd(30) 0 sind(30) 0];
      [0 1 0 0];
      [-sind(30) 0 cosd(30) 0];
      [0 0 0 1]];
R3 = [[cosd(20) -sind(20) 0 0];
      [sind(20) cosd(20) 0 0];
      [0 0 1 0];
      [0 0 0 1]];
R = R1*R2*R3;
```

R = $\begin{bmatrix} 0.3336 & -0.8379 & 0.3536 & 0 \\ 0.8173 & 0.4550 & 0.3536 & 0 \\ -0.4698 & 0.1710 & 0.8660 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

(A) Since this is a case of pure rotation, let's consider only the rotation matrix,

```
r10 = [[0.3336 -0.8379 0.3536];
     [0.8173 0.4550 0.3536];
     [-0.4698 0.1710 0.8660]];
r0 = [[1 0 0];
      [0 1 0];
      [0 0 1]];

% time steps are from t=0 to t=10, in steps of 2
```

```
t = (2:2:8)';
r_int = zeros(3,3,4);
% interpolating each element of the matrix linearly
for i = (1:1:length(t))
    T = t(i,1);
    for j = (1:1:size(r0,1))
        for k = (1:1:size(r0,2))
            r_int(j,k,i) = (((r10(j,k)-r0(j,k))/(10))*T)+r0(j,k);
        end
    end
end
r2 = r_int(:,:,1); % intermediate rotation matrix at t=2s
r4 = r_int(:,:,2); % intermediate rotation matrix at t=4s
r6 = r_int(:,:,3); % intermediate rotation matrix at t=6s
r8 = r_int(:,:,4); % intermediate rotation matrix at t=8s

figure;
view([170 35])
trplot(r0,'color','r');
hold on
trplot(r2,'color','g');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

```
hold on
trplot(r4,'color','b');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

```
hold on
trplot(r6,'color','m');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

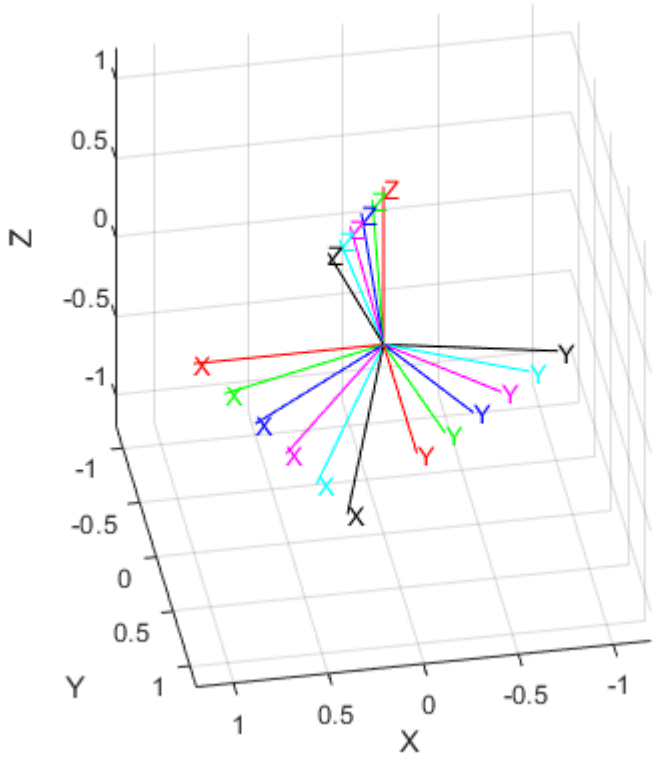
```
hold on
trplot(r8,'color','c');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

```
hold on
trplot(r10,'color','k');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

```
hold off
```



B) The absolute XYZ Euler angles can be obtained from the rotation matrix by considering the sequence of rotations as Z-Y-X,

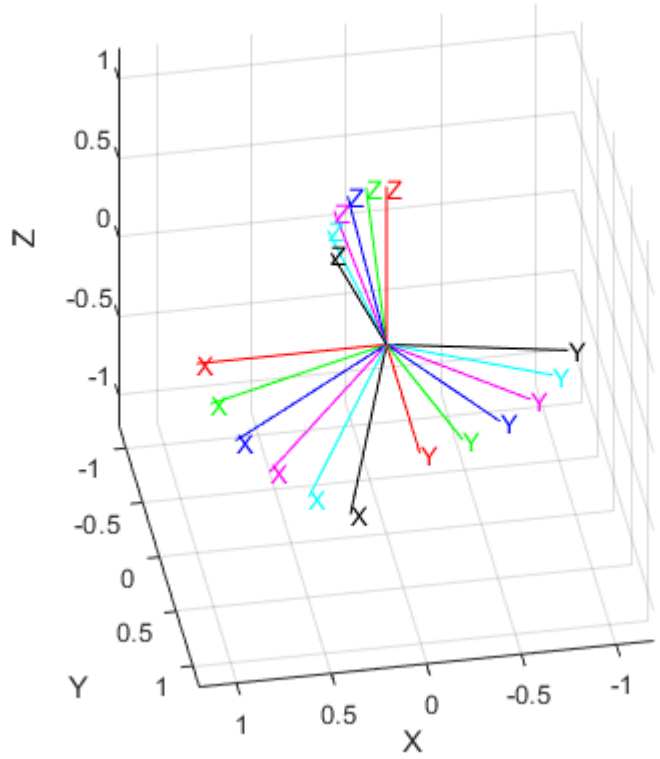
For the given rotation matrix, we get two sets of Euler angles,

$\theta_1 = 28.02\mathbf{deg}\ \psi_1 = 11.17\mathbf{deg}\ \phi_1 = 67.8\mathbf{deg}$

$\theta_2 = 151.98\mathbf{deg}\ \psi_2 = 191.17\mathbf{deg}\ \phi_2 = 247.8\mathbf{deg}$

```
%linear interpolation between the angles
% time steps are from t=0 to t=10, in steps of 2
r_eul_int = zeros(1,3,4);
r0_eul = [0 0 0];
r10_eul = [28.02 11.17 67.8];

% interpolating the angles linearly
for i = (1:1:length(t))
    T = t(i,1);
    for j = (1:1:size(r0_eul,1))
        for k = (1:1:size(r0_eul,2))
            r_eul_int(j,k,i) = (((r10_eul(j,k)-r0_eul(j,k))/(10))*T)+r0_eul(j,k);
        end
    end
end
r2_eul = r_eul_int(:,:,1);
r4_eul = r_eul_int(:,:,2);
r6_eul = r_eul_int(:,:,3);
r8_eul = r_eul_int(:,:,4);
```

```
% finding the corresponding rotation matrix using these angles

syms phi theta psi real

Rzphi   = [[cosd(phi) -sind(phi) 0];
           [sind(phi) cosd(phi) 0];
           [0 0 1]];
Rytheta = [[cosd(theta) 0 sind(theta)];
           [0 1 0];
           [-sind(theta) 0 cosd(theta)]];
Rxpsi   = [[1 0 0];
           [0 cosd(psi) -sind(psi)];
           [0 sind(psi) cosd(psi)]];
Rzyx    = Rzphi*Rytheta*Rxpsi;

r0_eul2rotm = double(subs(Rzyx,{theta,psi,phi},{r0_eul(1,1),r0_eul(1,2),r0_eul(1,3)}));
r2_eul2rotm = double(subs(Rzyx,{theta,psi,phi},{r2_eul(1,1),r2_eul(1,2),r2_eul(1,3)}));
r4_eul2rotm = double(subs(Rzyx,{theta,psi,phi},{r4_eul(1,1),r4_eul(1,2),r4_eul(1,3)}));
r6_eul2rotm = double(subs(Rzyx,{theta,psi,phi},{r6_eul(1,1),r6_eul(1,2),r6_eul(1,3)}));
r8_eul2rotm = double(subs(Rzyx,{theta,psi,phi},{r8_eul(1,1),r8_eul(1,2),r8_eul(1,3)}));
r10_eul2rotm= double(subs(Rzyx,{theta,psi,phi},{r10_eul(1,1),r10_eul(1,2),r10_eul(1,3)}));

figure;
view([170 35])
trplot(r0_eul2rotm,'color','r');
hold on
trplot(r2_eul2rotm,'color','g');
hold on
trplot(r4_eul2rotm,'color','b');
hold on
trplot(r6_eul2rotm,'color','m');
hold on
trplot(r8_eul2rotm,'color','c');
hold on
trplot(r10_eul2rotm,'color','k');
hold off
```



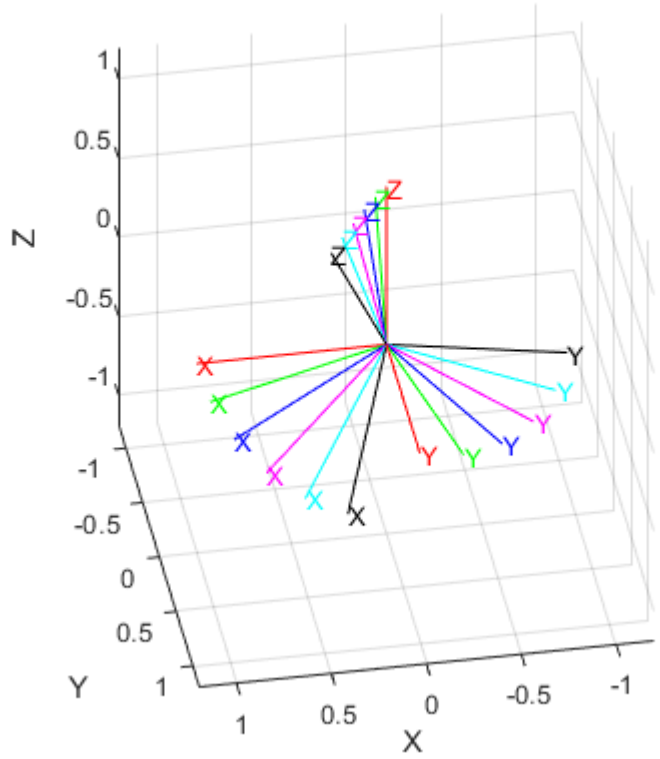
Since the other set of angles also yield the same rotation matrix, the intermediate euler angles would also render the same rotation matrices.

C) The relative XYZ Euler angles can be obtained from the rotation matrix by considering the sequence of rotations as X-Y-Z,

For the given rotation matrix, the Euler angles obtained are,

```
% finding the XYZ Euler angles corresponding to the two rotation matrices
r0_xyz = rotm2eul(r0,'XYZ');
r10_xyz = rotm2eul(r10,'XYZ');
r_xyz_int = zeros(1,3,4);

% interpolating the angles linearly
for i = (1:1:length(t))
    T = t(i,1);
    for j = (1:1:size(r0_xyz,1))
        for k = (1:1:size(r0_xyz,2))
            r_xyz_int(j,k,i) = (((r10_xyz(j,k)-r0_xyz(j,k))/(10))*T)+r0_xyz(j,k);
        end
    end
end

r2_xyz = r_xyz_int(:,:,1);
r4_xyz = r_xyz_int(:,:,2);
r6_xyz = r_xyz_int(:,:,3);
r8_xyz = r_xyz_int(:,:,4);
```

```
% finding the corresponding rotation matrix using these angles
syms phi theta psi real

Rxphi   = [[1 0 0];[0 cos(phi) -sin(phi)];[0 sin(phi) cos(phi)]];
Rytheta = [[cos(theta) 0 sin(theta)];[0 1 0];[-sin(theta) 0 cos(theta)]];
Rzpsi   = [[cos(psi) -sin(psi) 0];[sin(psi) cos(psi) 0];[0 0 1]];
Rxyz    = Rxphi*Rytheta*Rzpsi;

r0_xyz2rotm = double(subs(Rxyz,{phi,theta,psi},{r0_xyz(1,1),r0_xyz(1,2),r0_xyz(1,3)}));
r2_xyz2rotm = double(subs(Rxyz,{phi,theta,psi},{r2_xyz(1,1),r2_xyz(1,2),r2_xyz(1,3)}));
r4_xyz2rotm = double(subs(Rxyz,{phi,theta,psi},{r4_xyz(1,1),r4_xyz(1,2),r4_xyz(1,3)}));
r6_xyz2rotm = double(subs(Rxyz,{phi,theta,psi},{r6_xyz(1,1),r6_xyz(1,2),r6_xyz(1,3)}));
r8_xyz2rotm = double(subs(Rxyz,{phi,theta,psi},{r8_xyz(1,1),r8_xyz(1,2),r8_xyz(1,3)}));
r10_xyz2rotm= double(subs(Rxyz,{phi,theta,psi},{r10_xyz(1,1),r10_xyz(1,2),r10_xyz(1,3)}));

figure;
view([170 35])
trplot(r0_xyz2rotm,'color','r');
hold on
trplot(r2_xyz2rotm,'color','g');
hold on
trplot(r4_xyz2rotm,'color','b');
hold on
trplot(r6_xyz2rotm,'color','m');
hold on
trplot(r8_xyz2rotm,'color','c');
hold on
trplot(r10_xyz2rotm,'color','k');
hold off
```



D) Axis angle representation can be found out using the function 'rotm2axang'

```
r0_axang = rotm2axang(r0);
r10_axang = rotm2axang(r10);
r_axang_int = zeros(1,4,4);

% interpolating the angles linearly
for i = (1:1:length(t))
    T = t(i,1);
    for j = (1:1:size(r0_axang,1))
        for k = (1:1:size(r0_axang,2))
            r_axang_int(j,k,i) = (((r10_axang(j,k)-r0_axang(j,k))/(10))*T)+r0_axang(j,k);
        end
    end
end

r2_axang = r_axang_int(:,:,1);
r4_axang = r_axang_int(:,:,2);
r6_axang = r_axang_int(:,:,3);
r8_axang = r_axang_int(:,:,4);

%finding the corresponding rotation matrix using these axis and angles

syms kx ky kz theta real

Raxang = [[kx^2*(1-cos(theta))+cos(theta) kx*ky*(1-cos(theta))-kz*sin(theta) kx*kz*(1-cos(theta
          [kx*ky*(1-cos(theta))+kz*sin(theta) ky^2*(1-cos(theta))+cos(theta) ky*kz*(1-cos(theta
          [kx*kz*(1-cos(theta))-ky*sin(theta) ky*kz*(1-cos(theta))+kx*sin(theta) kz^2*(1-cos(th
```

```
r0_axang2rotm = double(subs(Raxang,{kx,ky,kz,theta},{r0_axang(1,1),r0_axang(1,2),r0_axang(1,3),
r2_axang2rotm = double(subs(Raxang,{kx,ky,kz,theta},{r2_axang(1,1),r2_axang(1,2),r2_axang(1,3),
r4_axang2rotm = double(subs(Raxang,{kx,ky,kz,theta},{r4_axang(1,1),r4_axang(1,2),r4_axang(1,3),
r6_axang2rotm = double(subs(Raxang,{kx,ky,kz,theta},{r6_axang(1,1),r6_axang(1,2),r6_axang(1,3),
r8_axang2rotm = double(subs(Raxang,{kx,ky,kz,theta},{r8_axang(1,1),r8_axang(1,2),r8_axang(1,3),
r10_axang2rotm= double(subs(Raxang,{kx,ky,kz,theta},{r10_axang(1,1),r10_axang(1,2),r10_axang(1,

figure;
view([170 35])
trplot(r0_axang2rotm,'color','r');
hold on
trplot(r2_axang2rotm,'color','g');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

```
hold on
trplot(r4_axang2rotm,'color','b');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

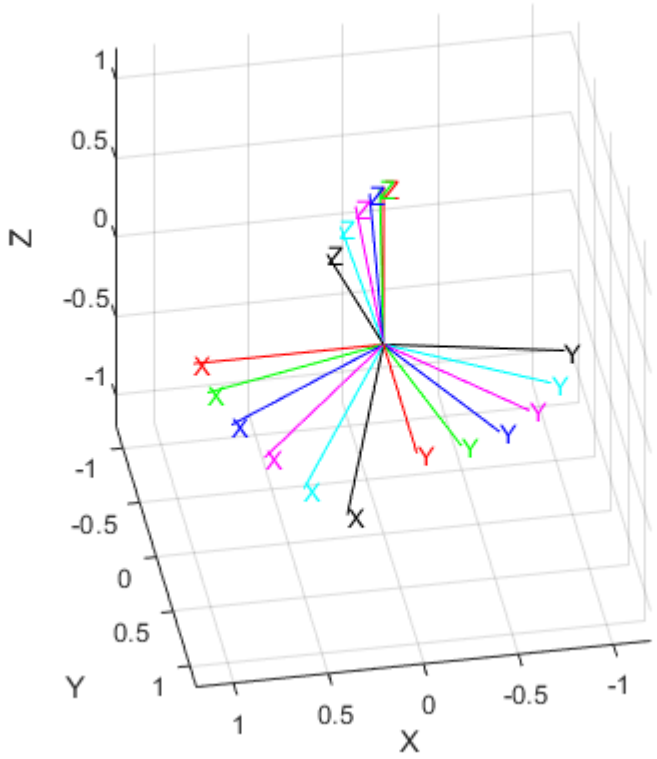
```
hold on
trplot(r6_axang2rotm,'color','m');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

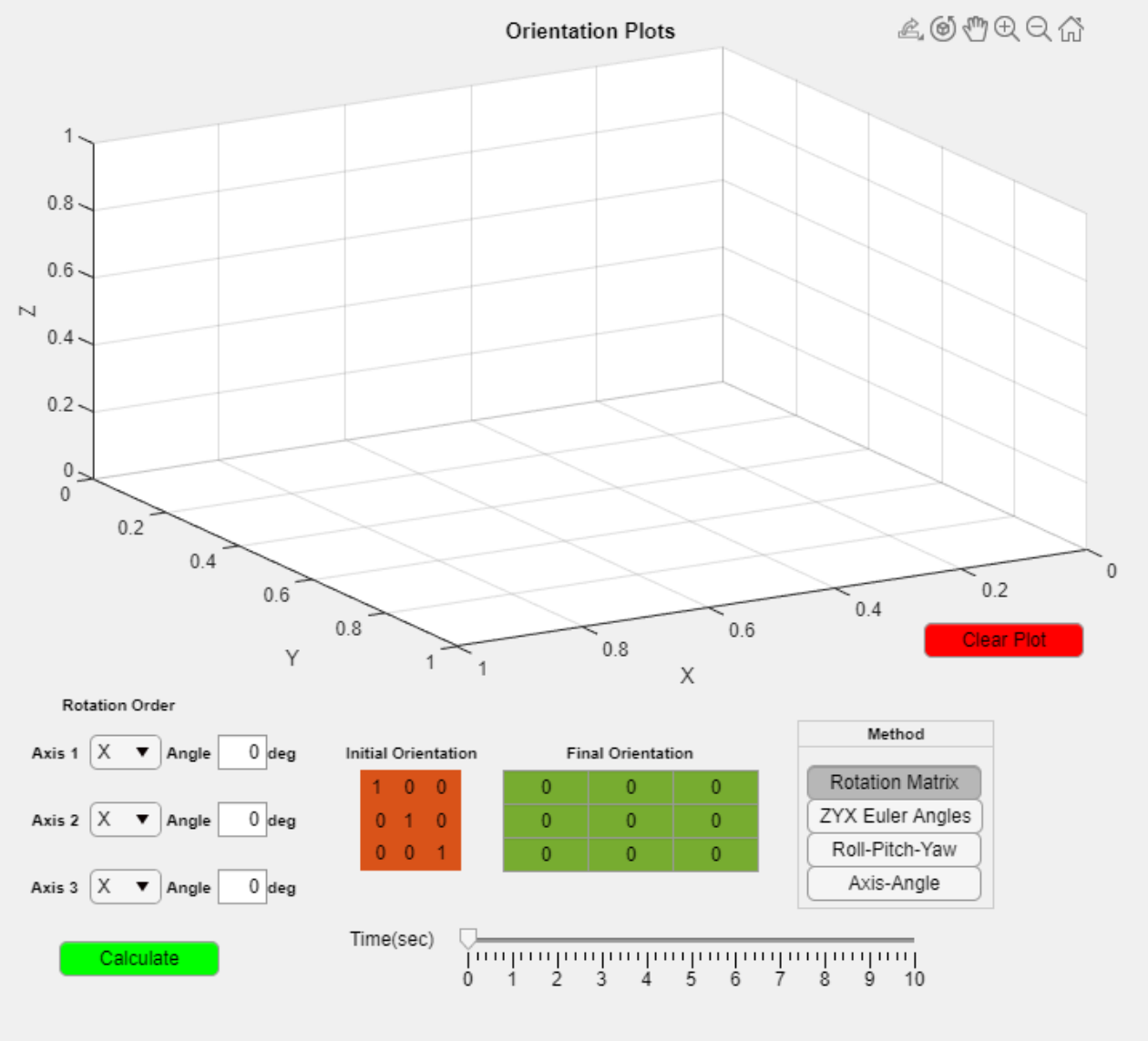
```
hold on
trplot(r8_axang2rotm,'color','c');
```

```
Warning: The new value for the Matrix property may cause rendering problems.
```

```
hold on
trplot(r10_axang2rotm,'color','k');
hold off
```




```
run('Q1_GUI.mlapp');
```



Inputs to the GUI are as follows,

(1) Axis of rotation for the frames

(2) Corresponding angles of rotation

(3) Method of interpolation

(4) Time in seconds, at which the position of the frame needs to be visualised

Directions to use the GUI,

(1) Enter the axis and angles of rotation, press calculate. This calculates the 'Final Orientation' matrix

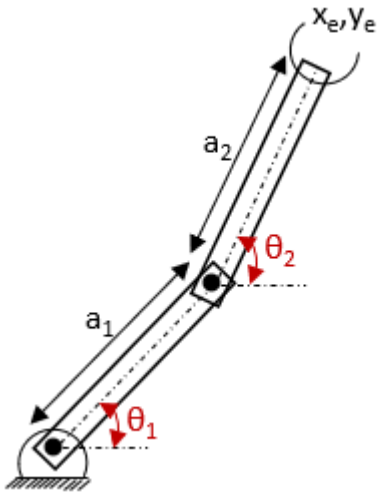
The initial orientation matrix is assumed to be aligned with the global frame of reference

(2) Select the method of interpolations, which essentially decides what will be interpolated

(3) Slide the 'time' slider to visualise the frames at different times

(4) Press clear plot to clear the figure, to be used when switching from one method to the other

PROBLEM 2

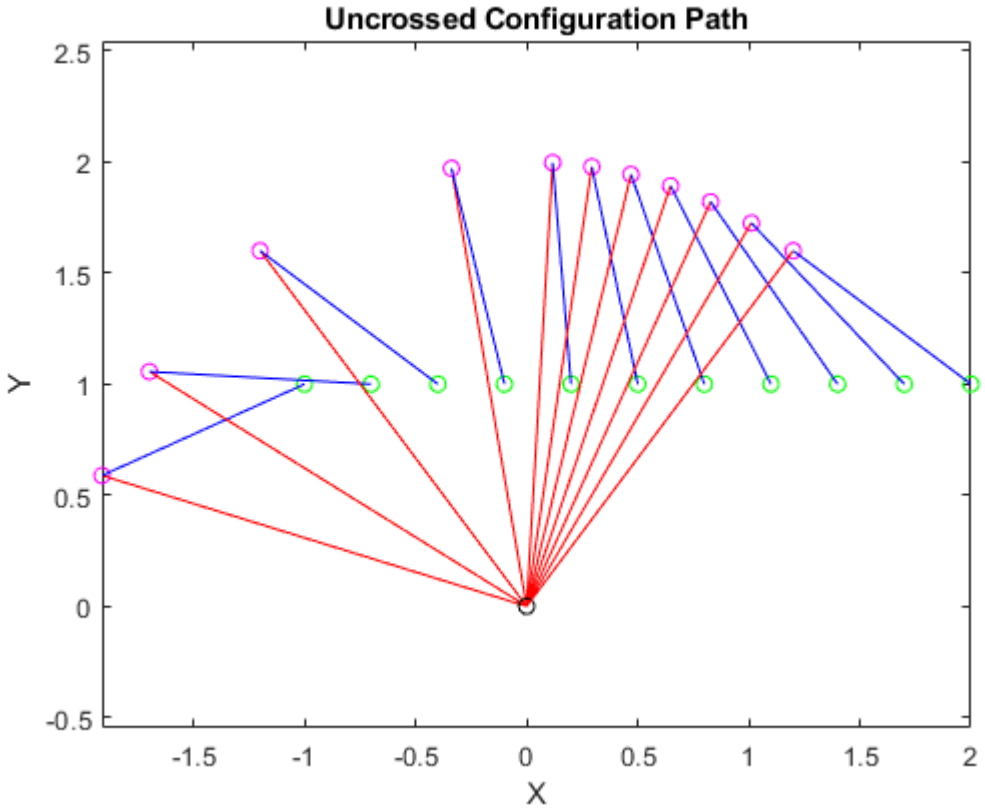


The end effector has to travel from P1 = (x_1, y_1) = (-1,1) to P2 = (x_2, y_2) = (2,1) in 10 seconds.

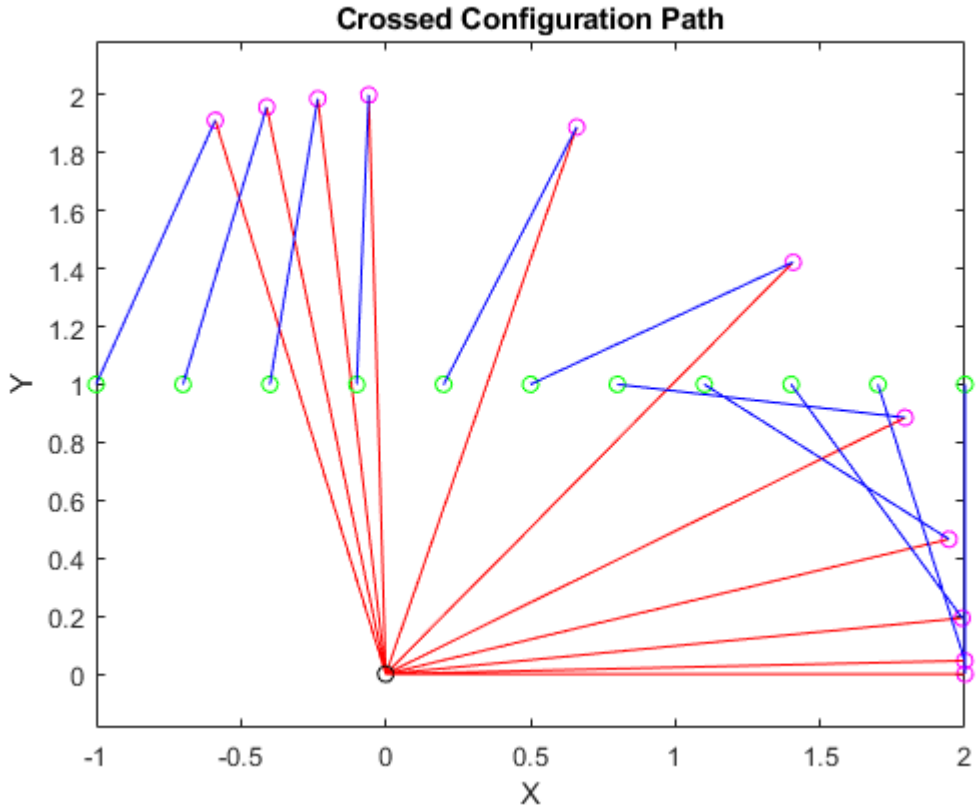
(A) **Linearly interpolated trajectory in Cartesian space:**

```
clear all;
cla reset;
p0  = [-1 1];   %position at t=0s
p10 = [2 1];    %position at t=10s
L   = [2 1];    %link-length vector
%creating the intermediate positions considering a linear trajectory
t = (1:1:9)'; %positions will be calculated every second
pe_int = zeros(1,2,length(t));  %intermediate coordinates for the end effector
for i = (1:1:length(t))
    for j = (1:1:size(p0,1))
        for k = (1:1:size(p0,2))
            pe_int(j,k,i) = (((p10(j,k) - p0(j,k))/10)*t(i,1))+(p0(j,k));
        end
    end
end
p1 = pe_int(:,:,1);
p2 = pe_int(:,:,2);
p3 = pe_int(:,:,3);
p4 = pe_int(:,:,4);
p5 = pe_int(:,:,5);
p6 = pe_int(:,:,6);
p7 = pe_int(:,:,7);
p8 = pe_int(:,:,8);
p9 = pe_int(:,:,9);
P  = [p0;p1;p2;p3;p4;p5;p6;p7;p8;p9;p10]; %vector for all the end effector positions

% calculating the theta values for these intermediate positions using the
% function 'invkinema.m'
theta1_u = zeros(length(P),1);
theta2_u = zeros(length(P),1);
theta1_c = zeros(length(P),1);
```

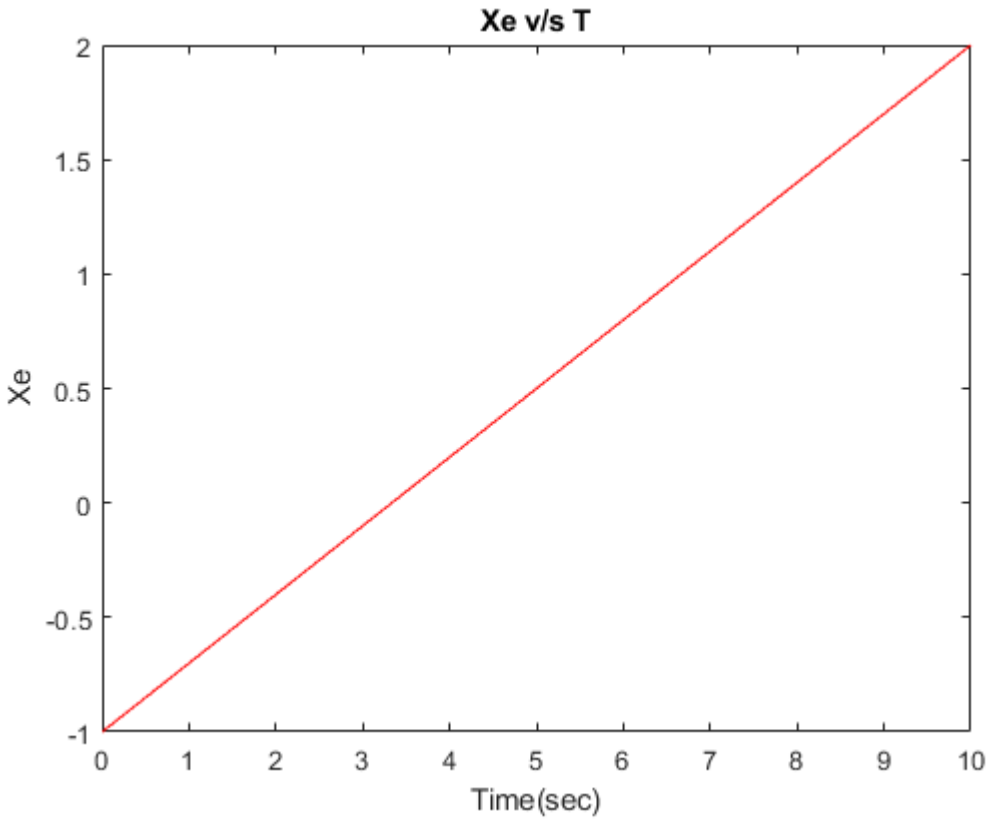
```
theta2_c = zeros(length(P),1);

% uncrossed position sigma = 1
for i = (1:1:length(P))
    [theta1_u(i,1),theta2_u(i,1)] = invkinema(L,P(i,1),P(i,2),1);
end

% crossed position sigma = -1
for i = (1:1:length(P))
    [theta1_c(i,1),theta2_c(i,1)] = invkinema(L,P(i,1),P(i,2),-1);
end

% vector for all positions of joint 2
Pu_j2 = [L(1).*cosd(theta1_u) L(1).*sind(theta1_u)]; %uncrossed configuration
Pc_j2 = [L(1).*cosd(theta1_c) L(1).*sind(theta1_c)]; %crossed configuration

%plotting the results
figure(1);  %uncrossed path
for i = (1:1:length(P))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(Pu_j2(i,1),Pu_j2(i,2),'mo');
    hold on
    plot(P(i,1),P(i,2),'go');
    hold on
    plot([0,Pu_j2(i,1)],[0,Pu_j2(i,2)],'r-');   %first link positions
    hold on
    plot([Pu_j2(i,1),P(i,1)],[Pu_j2(i,2),P(i,2)],'b-')  %second link positions
    title('Uncrossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```



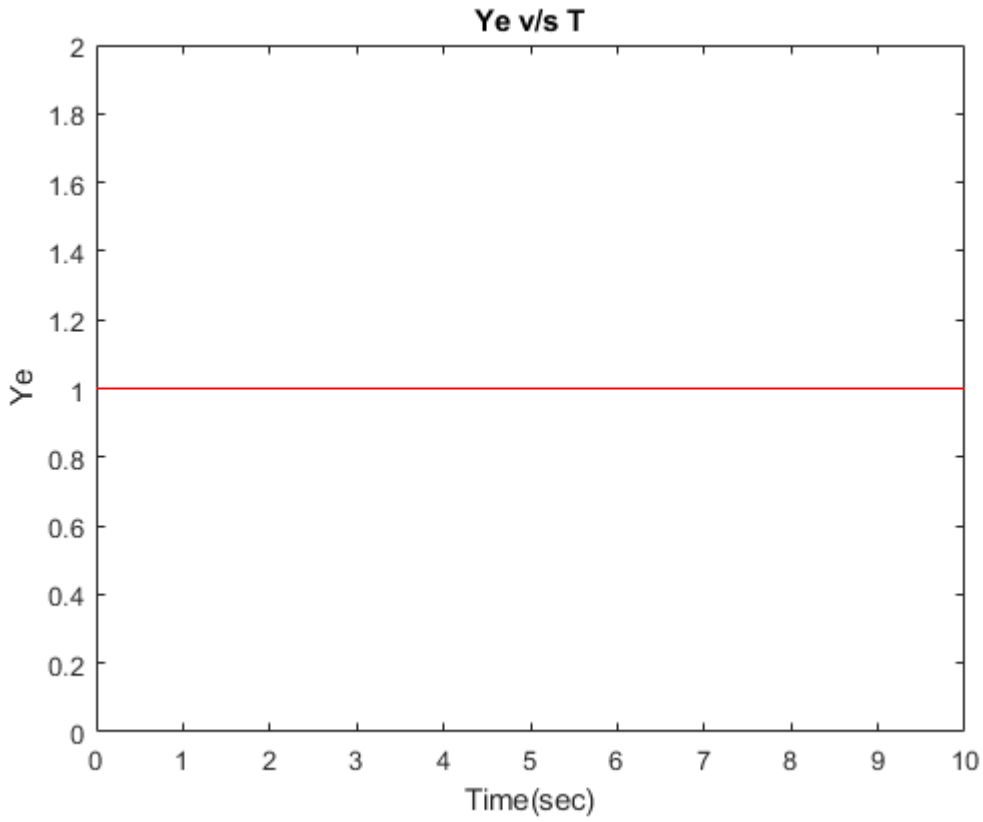

```
figure(2);  %crossed path
for i = (1:1:length(P))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(Pc_j2(i,1),Pc_j2(i,2),'mo');
    hold on
    plot(P(i,1),P(i,2),'go');
    hold on
    plot([0,Pc_j2(i,1)],[0,Pc_j2(i,2)],'r-');   %first link positions
    hold on
    plot([Pc_j2(i,1),P(i,1)],[Pc_j2(i,2),P(i,2)],'b-')  %second link positions
    title('Crossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```



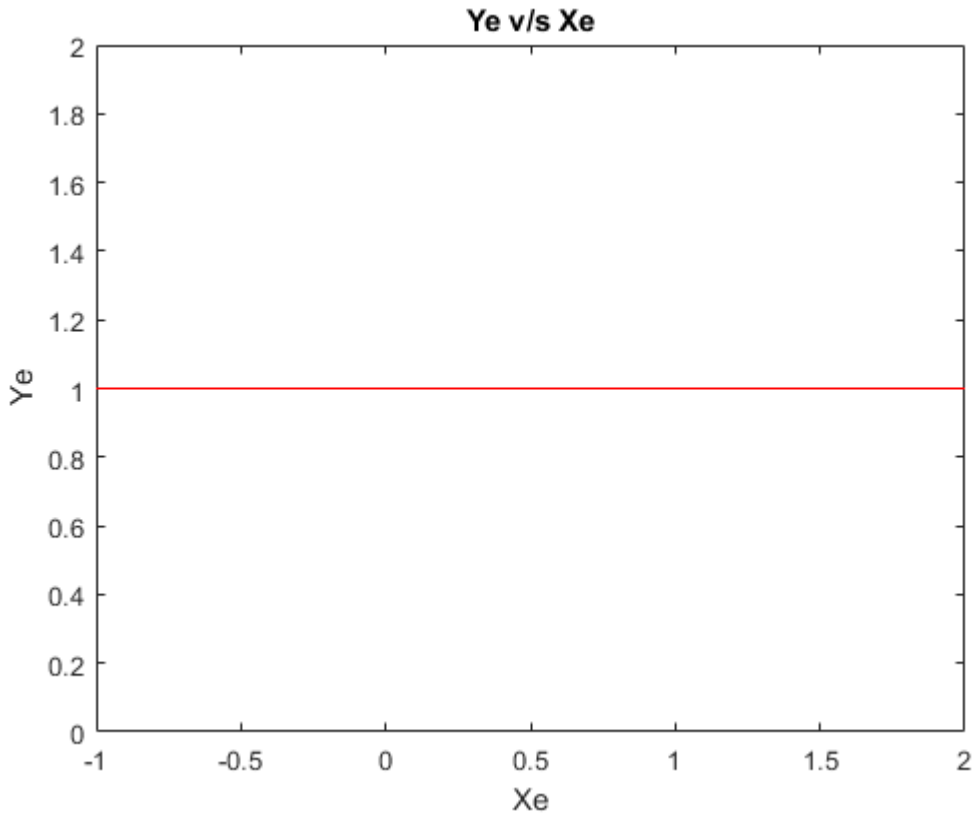

```
%(Ai) Xe v/s t
figure(3);
T = (0:1:10)';
plot(T,P(:,1),'r-')
title('Xe v/s T');
xlabel('Time(sec)');
ylabel('Xe');
```



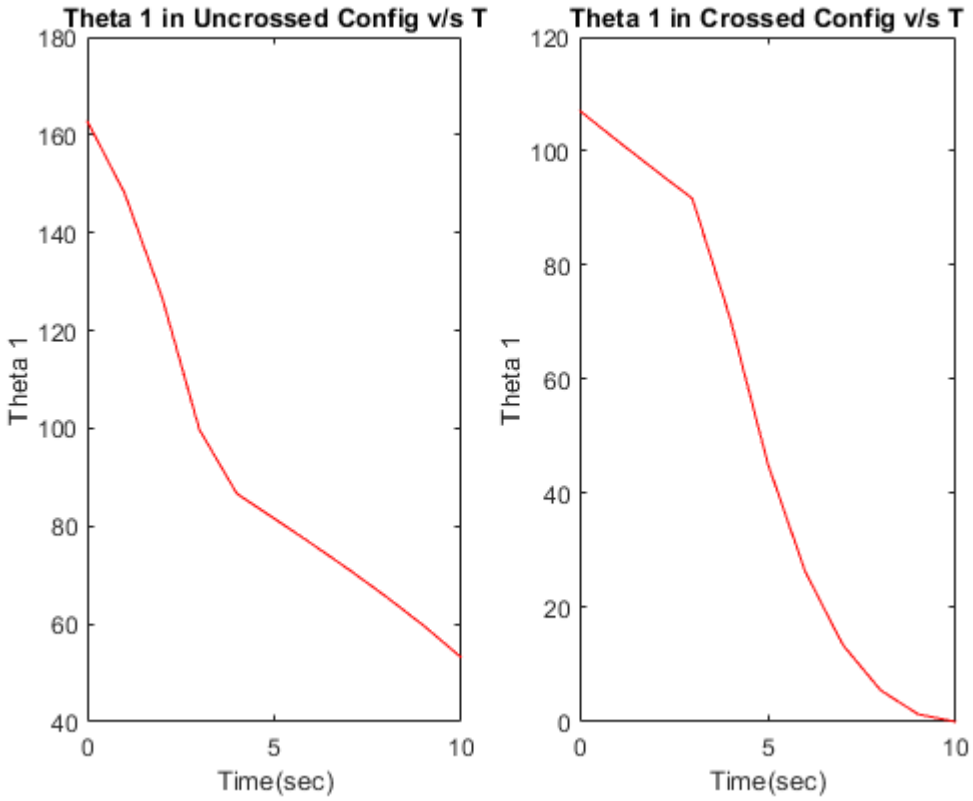

```
%(Aii) Ye v/s t
figure(4);
plot(T,P(:,2),'r-')
title('Ye v/s T');
xlabel('Time(sec)');
ylabel('Ye');
```



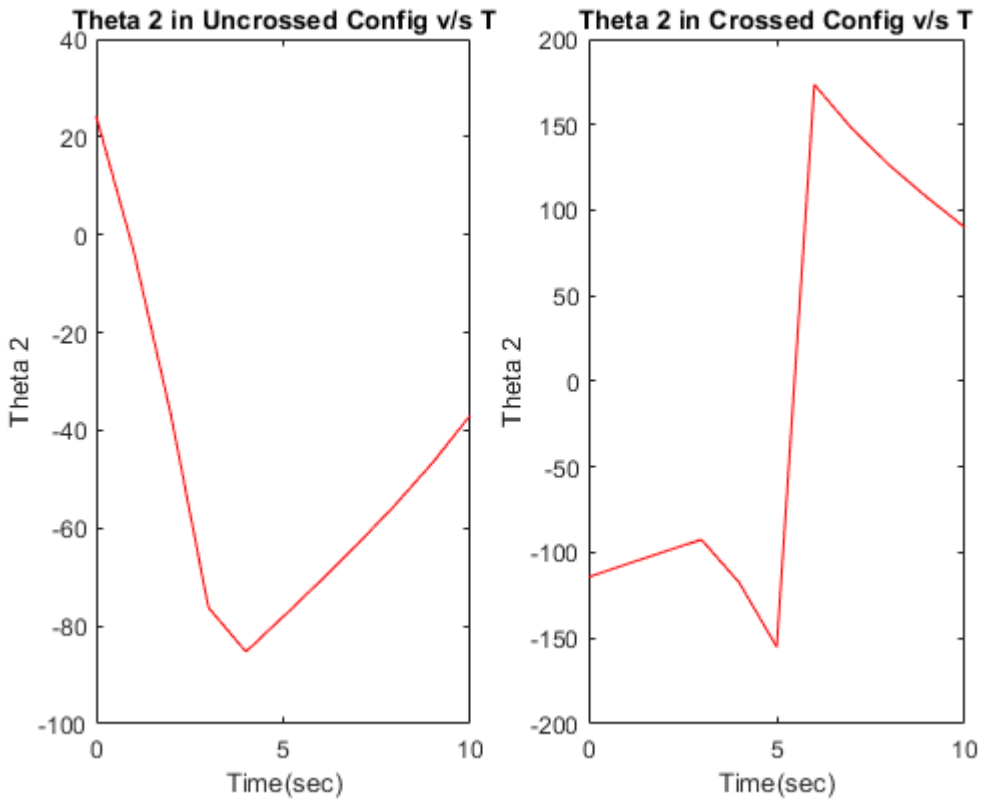

```
%(Aiii) Ye v/s Xe
figure(5);
plot(P(:,1),P(:,2),'r-')
title('Ye v/s Xe');
xlabel('Xe');
ylabel('Ye');
```




```
%(Aiv) theta 1 v/s t in crossed and uncrossed condition
figure(6);
subplot(1,2,1)
plot(T,theta1_u,'r-')
title('Theta 1 in Uncrossed Config v/s T');
xlabel('Time(sec)');
ylabel('Theta 1');
subplot(1,2,2)
plot(T,theta1_c,'r-')
title('Theta 1 in Crossed Config v/s T');
xlabel('Time(sec)');
ylabel('Theta 1');
```




```
%(Av) theta 2 v/s t in crossed and uncrossed condition
figure(7);
subplot(1,2,1)
plot(T,theta2_u,'r-')
title('Theta 2 in Uncrossed Config v/s T');
xlabel('Time(sec)');
ylabel('Theta 2');
subplot(1,2,2)
plot(T,theta2_c,'r-')
title('Theta 2 in Crossed Config v/s T');
xlabel('Time(sec)');
ylabel('Theta 2');
```

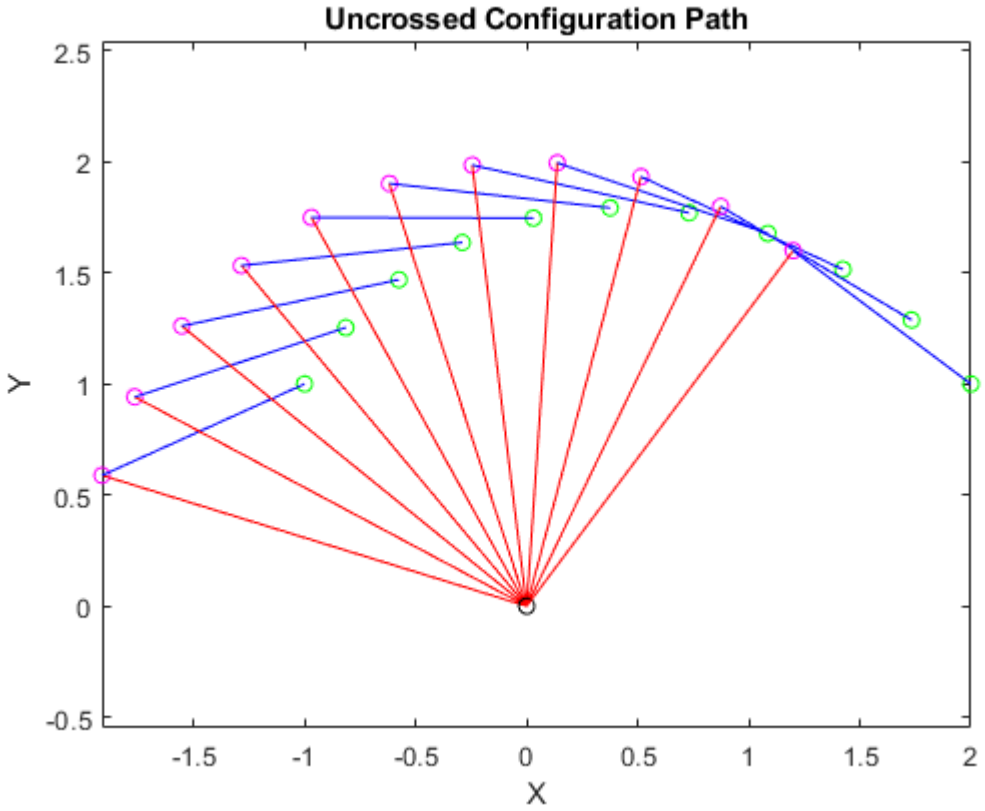


(B) **Linearly interpolated trajectory in joint space:**

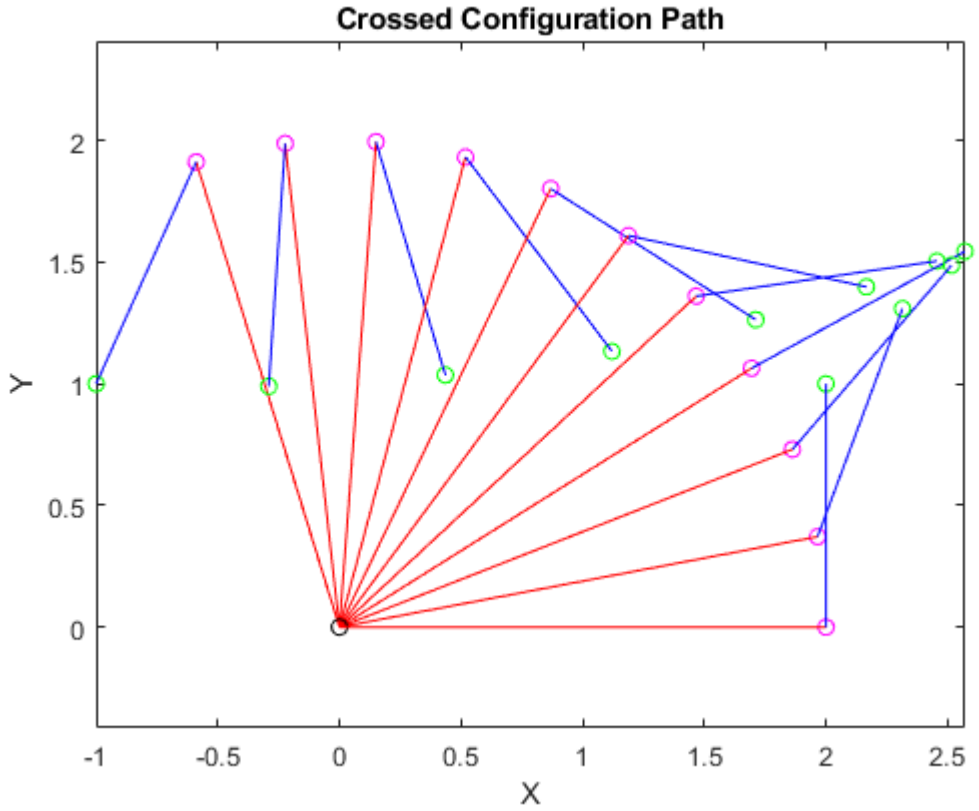
```
clear all;
p0  = [-1 1];   %position at t=0s
p10 = [2 1];    %position at t=10s
L   = [2 1];    %link-length vector

%calculating the joint 2 coordinates using the function 'RR_InvPosKin.m'
%corresponding to p0 and p10

[theta1_u0,theta2_u0] = invkinema(L,p0(1,1),p0(1,2),1);      %theta values at t=0,  uncrossed
[theta1_u10,theta2_u10] = invkinema(L,p10(1,1),p10(1,2),1);  %theta values at t=10, uncrossed
[theta1_c0,theta2_c0] = invkinema(L,p0(1,1),p0(1,2),-1);     %theta values at t=0, crossed
[theta1_c10,theta2_c10] = invkinema(L,p10(1,1),p10(1,2),-1); %theta values at t=10, crossed

%% uncrossed configuration
t = (1:1:9)';
theta1_uint = zeros(1,1,length(t));  %3D matrix to store the interim theta1(uncrossed)
theta2_uint = zeros(1,1,length(t));  %3D matrix to store the interim theta2(uncrossed)

for i = (1:1:length(t))
    theta1_uint(1,1,i) = (((theta1_u10-theta1_u0)/10)*t(i,1))+theta1_u0;
    theta2_uint(1,1,i) = (((theta2_u10-theta2_u0)/10)*t(i,1))+theta2_u0;
end
```

```
theta1_u1 = theta1_uint(1,1,1);
theta1_u2 = theta1_uint(1,1,2);
theta1_u3 = theta1_uint(1,1,3);
theta1_u4 = theta1_uint(1,1,4);
theta1_u5 = theta1_uint(1,1,5);
theta1_u6 = theta1_uint(1,1,6);
theta1_u7 = theta1_uint(1,1,7);
theta1_u8 = theta1_uint(1,1,8);
theta1_u9 = theta1_uint(1,1,9);
theta1_u = [theta1_u0;theta1_u1;theta1_u2;theta1_u3;theta1_u4;theta1_u5;theta1_u6;theta1_u7;the

theta2_u1 = theta2_uint(1,1,1);
theta2_u2 = theta2_uint(1,1,2);
theta2_u3 = theta2_uint(1,1,3);
theta2_u4 = theta2_uint(1,1,4);
theta2_u5 = theta2_uint(1,1,5);
theta2_u6 = theta2_uint(1,1,6);
theta2_u7 = theta2_uint(1,1,7);
theta2_u8 = theta2_uint(1,1,8);
theta2_u9 = theta2_uint(1,1,9);
theta2_u = [theta2_u0;theta2_u1;theta2_u2;theta2_u3;theta2_u4;theta2_u5;theta2_u6;theta2_u7;the

%coordinates of joint 2 and end effector (uncrossed)
j2_u = zeros(length(theta1_u),2); %initialize a 2D matrix to store all the joint 2 coordinates
ee_u = zeros(length(theta1_u),2); %initialize a 2D matrix to store all the end effector coordi
for i = (1:1:length(theta1_u))
    j2_u(i,1) = L(1)*cosd(theta1_u(i,1));
    j2_u(i,2) = L(1)*sind(theta1_u(i,1));
    ee_u(i,1) = L(1)*cosd(theta1_u(i,1)) + L(2)*cosd(theta2_u(i,1));
    ee_u(i,2) = L(1)*sind(theta1_u(i,1)) + L(2)*sind(theta2_u(i,1));
end

%% crossed configuration
theta1_cint = zeros(1,1,length(t));  %3D matrix to store the interim theta1(crossed)
theta2_cint = zeros(1,1,length(t));  %3D matrix to store the interim theta2(crossed)

for i = (1:1:length(t))
    theta1_cint(1,1,i) = (((theta1_c10-theta1_c0)/10)*t(i,1))+theta1_c0;
    theta2_cint(1,1,i) = (((theta2_c10-theta2_c0)/10)*t(i,1))+theta2_c0;
end

theta1_c1 = theta1_cint(1,1,1);
theta1_c2 = theta1_cint(1,1,2);
theta1_c3 = theta1_cint(1,1,3);
theta1_c4 = theta1_cint(1,1,4);
theta1_c5 = theta1_cint(1,1,5);
theta1_c6 = theta1_cint(1,1,6);
theta1_c7 = theta1_cint(1,1,7);
theta1_c8 = theta1_cint(1,1,8);
theta1_c9 = theta1_cint(1,1,9);
theta1_c = [theta1_c0;theta1_c1;theta1_c2;theta1_c3;theta1_c4;theta1_c5;theta1_c6;theta1_c7;the

theta2_c1 = theta2_cint(1,1,1);
theta2_c2 = theta2_cint(1,1,2);
```

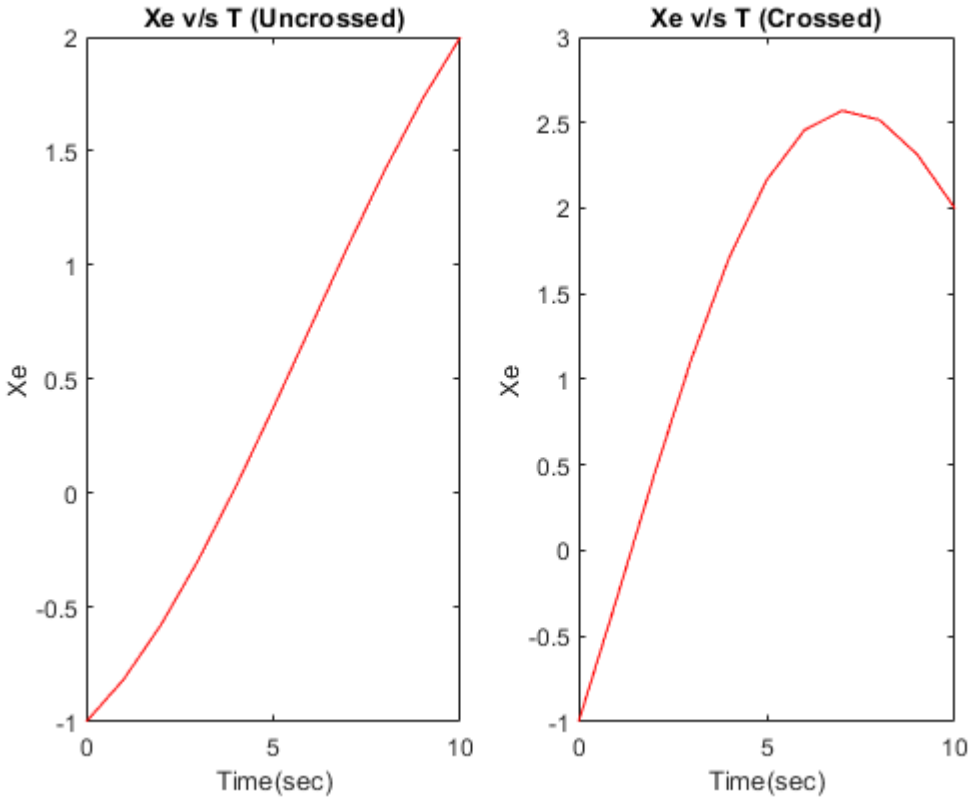
```
theta2_c3 = theta2_cint(1,1,3);
theta2_c4 = theta2_cint(1,1,4);
theta2_c5 = theta2_cint(1,1,5);
theta2_c6 = theta2_cint(1,1,6);
theta2_c7 = theta2_cint(1,1,7);
theta2_c8 = theta2_cint(1,1,8);
theta2_c9 = theta2_cint(1,1,9);
theta2_c = [theta2_c0;theta2_c1;theta2_c2;theta2_c3;theta2_c4;theta2_c5;theta2_c6;theta2_c7;the

%coordinates of joint 2 and end effector (uncrossed)
j2_c = zeros(length(theta1_c),2); %initialize a 2D matrix to store all the joint 2 coordinates
ee_c = zeros(length(theta1_c),2); %initialize a 2D matrix to store all the end effector coordi
for i = (1:1:length(theta1_c))
    j2_c(i,1) = L(1)*cosd(theta1_c(i,1));
    j2_c(i,2) = L(1)*sind(theta1_c(i,1));
    ee_c(i,1) = L(1)*cosd(theta1_c(i,1)) + L(2)*cosd(theta2_c(i,1));
    ee_c(i,2) = L(1)*sind(theta1_c(i,1)) + L(2)*sind(theta2_c(i,1));
end

%plotting the results
figure(8);  %uncrossed path
for i = (1:1:length(ee_u))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(j2_u(i,1),j2_u(i,2),'mo');
    hold on
    plot(ee_u(i,1),ee_u(i,2),'go');
    hold on
    plot([0,j2_u(i,1)],[0,j2_u(i,2)],'r-');   %first link positions
    hold on
    plot([j2_u(i,1),ee_u(i,1)],[j2_u(i,2),ee_u(i,2)],'b-')  %second link positions
    title('Uncrossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```



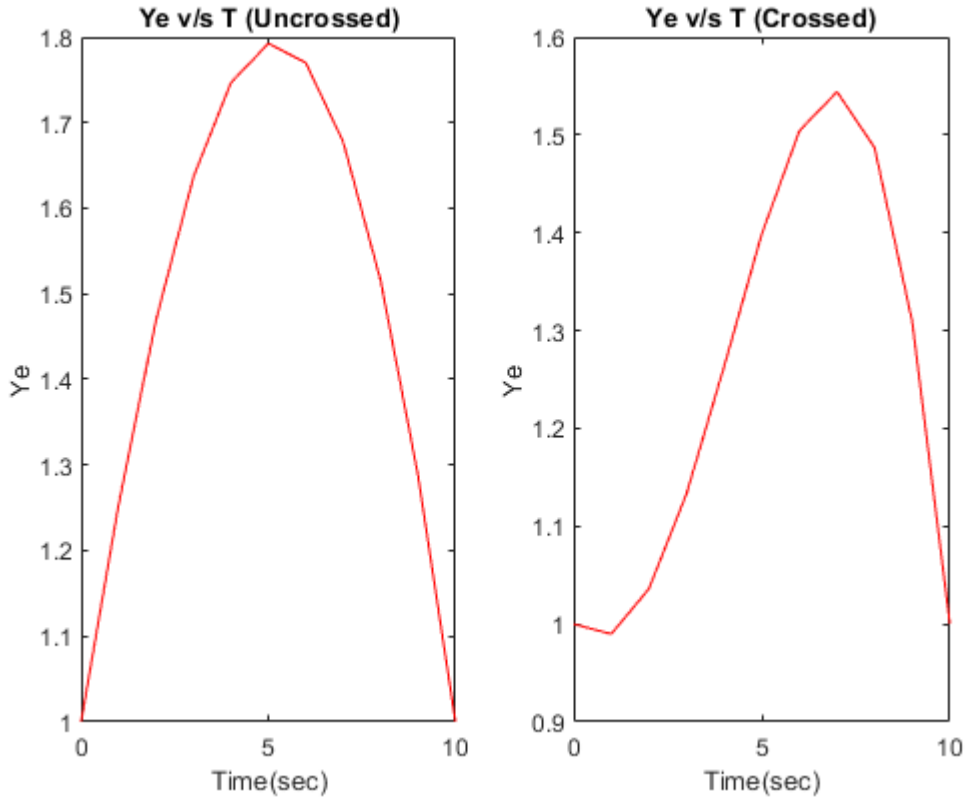

```
figure(9);  %crossed path
for i = (1:1:length(ee_c))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(j2_c(i,1),j2_c(i,2),'mo');
    hold on
    plot(ee_c(i,1),ee_c(i,2),'go');
    hold on
    plot([0,j2_c(i,1)],[0,j2_c(i,2)],'r-');   %first link positions
    hold on
    plot([j2_c(i,1),ee_c(i,1)],[j2_c(i,2),ee_c(i,2)],'b-')  %second link positions
    title('Crossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```



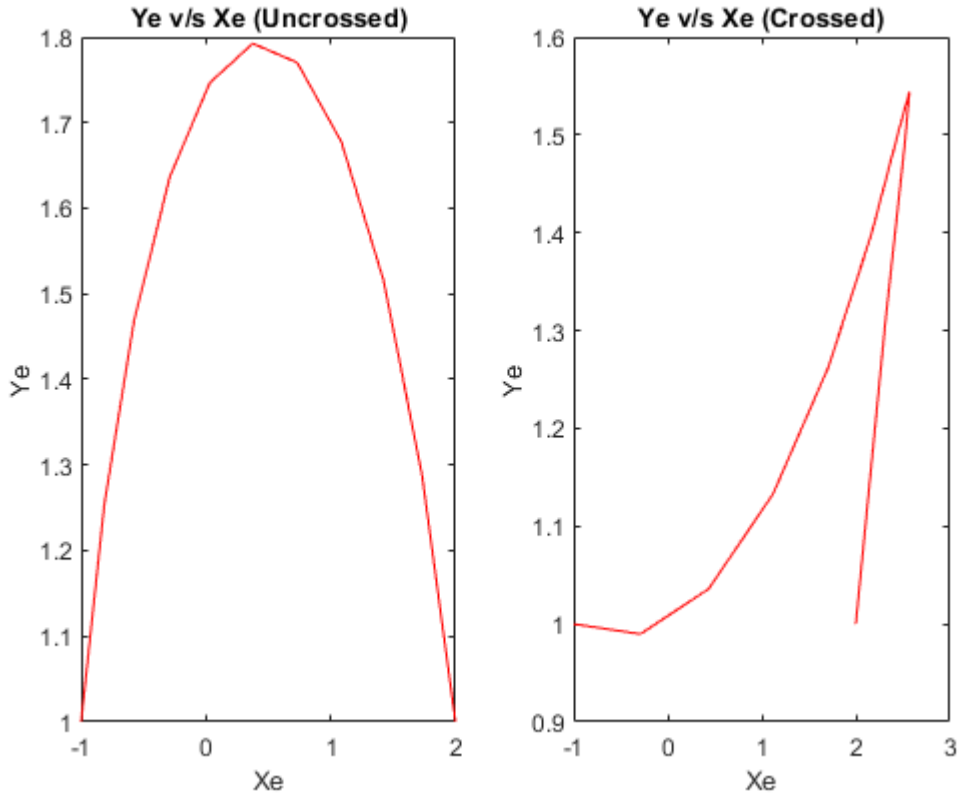

```
%(Bi) Xe v/s t
figure(10);
T = (0:1:10)';
subplot(1,2,1)
plot(T,ee_u(:,1),'r-')
title('Xe v/s T (Uncrossed)');
xlabel('Time(sec)');
ylabel('Xe');
subplot(1,2,2)
plot(T,ee_c(:,1),'r-')
title('Xe v/s T (Crossed)');
xlabel('Time(sec)');
ylabel('Xe');
```



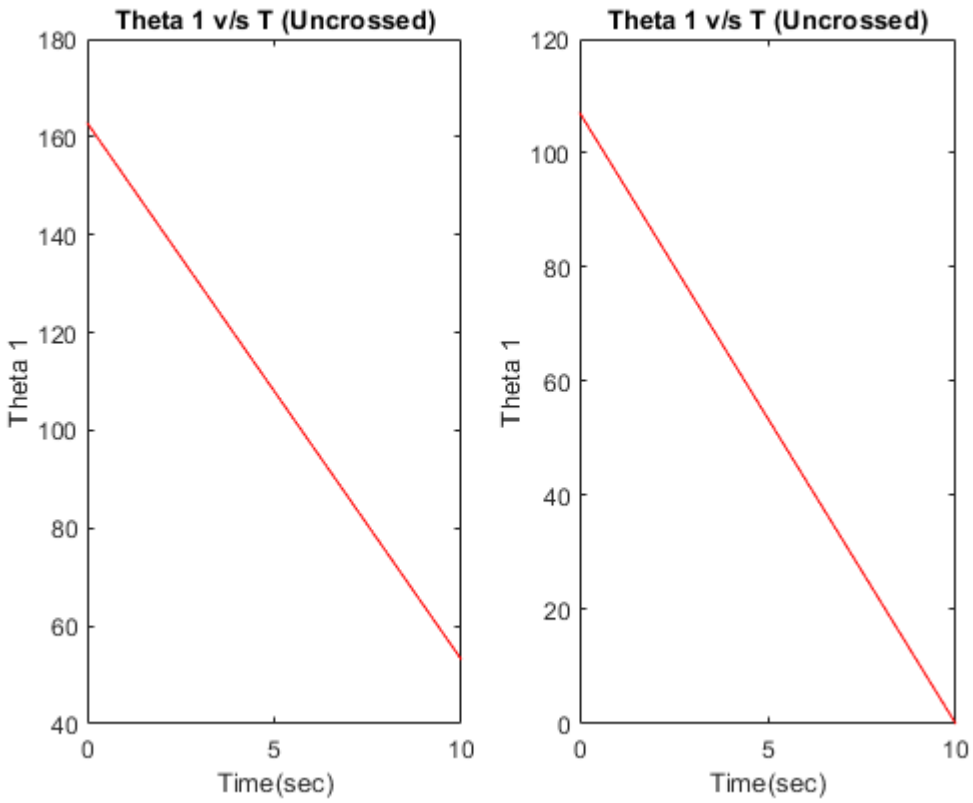

```
%(Bii) Ye v/s t
figure(11);
T = (0:1:10)';
subplot(1,2,1)
plot(T,ee_u(:,2),'r-')
title('Ye v/s T (Uncrossed)');
xlabel('Time(sec)');
ylabel('Ye');
subplot(1,2,2)
plot(T,ee_c(:,2),'r-')
title('Ye v/s T (Crossed)');
xlabel('Time(sec)');
ylabel('Ye');
```



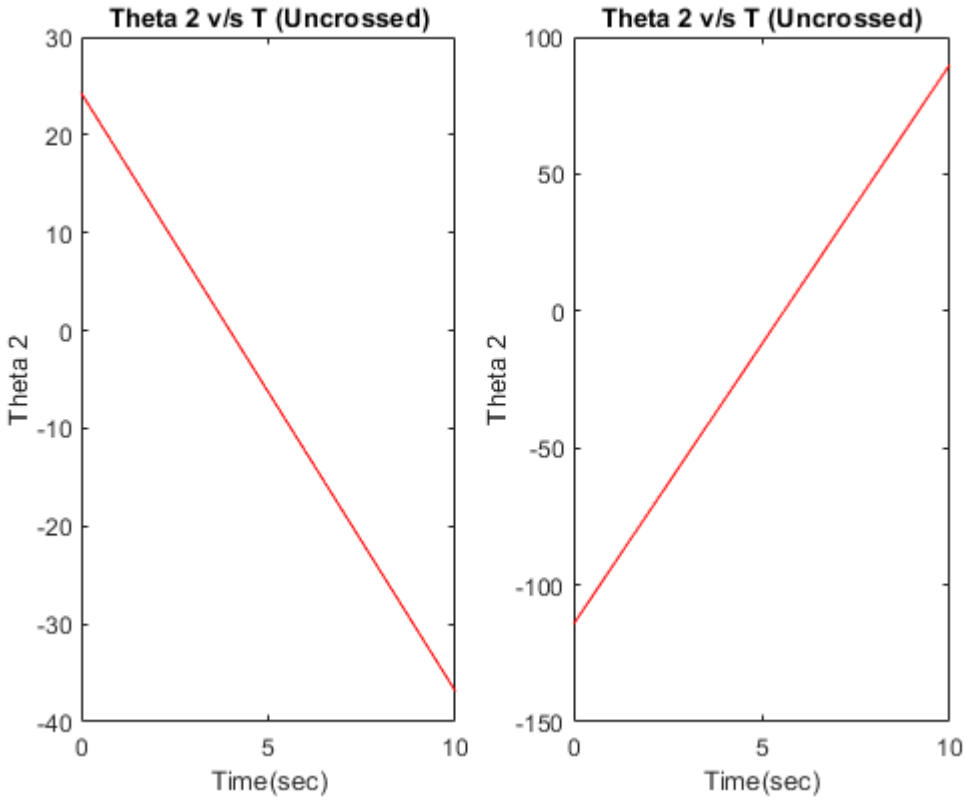

```
%(Biii) Ye v/s Xe
figure(12);
T = (0:1:10)';
subplot(1,2,1)
plot(ee_u(:,1),ee_u(:,2),'r-')
title('Ye v/s Xe (Uncrossed)');
xlabel('Xe');
ylabel('Ye');
subplot(1,2,2)
plot(ee_c(:,1),ee_c(:,2),'r-')
title('Ye v/s Xe (Crossed)');
xlabel('Xe');
ylabel('Ye');
```




```
%(Biv) theta 1 v/s t in crossed and uncrossed condition
figure(13);
subplot(1,2,1)
plot(T,theta1_u,'r-')
title('Theta 1 v/s T (Uncrossed)');
xlabel('Time(sec)');
ylabel('Theta 1');
subplot(1,2,2)
plot(T,theta1_c,'r-')
title('Theta 1 v/s T (Crossed)');
xlabel('Time(sec)');
ylabel('Theta 1');
```




```
%(Biv) theta 2 v/s t in crossed and uncrossed condition
figure(14);
subplot(1,2,1)
plot(T,theta2_u,'r-')
title('Theta 2 v/s T (Uncrossed)');
xlabel('Time(sec)');
ylabel('Theta 2');
subplot(1,2,2)
plot(T,theta2_c,'r-')
title('Theta 2 v/s T (Crossed)');
xlabel('Time(sec)');
ylabel('Theta 2');
```



(C) **Manipulator starts from rest and stops at rest, trajectory described by a quintic polynomial in cartesian coordinates**

The equations of motion of this system can be given as,

$$Y_e(t) = \mathrm{a}_y t^5 + \mathrm{b}_y t^4 + \mathrm{c}_y t^3 + \mathrm{d}_y t^2 + \mathrm{e}_y t + f_y$$
$$X_e(t) = \mathrm{a}_x t^5 + \mathrm{b}_x t^4 + \mathrm{c}_x t^3 + \mathrm{d}_x t^2 + \mathrm{e}_x t + f_x$$

First considering the equation of $Y_e(t)$, we have 6 equations and following are the 6 equations needed to find the coeffecients,

(i) $Y_e(t=0) = f_y = 1$

(ii) $Y_e(t=10) = \mathrm{a}_y 10^5 + \mathrm{b}_y 10^4 + \mathrm{c}_y 10^3 + \mathrm{d}_y 10^2 + \mathrm{e}_y 10 + f_y = 1$

(iii) $\dot{Y}_e(t=0) = 5a_y t^4 + 4b_y t^3 + 3c_y t^2 + 2d_y t \;\; + e_y = e_y = 0$

(iv) $\dot{Y}_e(t=10) = 5a_y 10^4 + 4b_y 10^3 + 3c_y 10^2 + 20d_y + e_y = 0$

(v) $\ddot{Y}_e(t=0) = 20a_y t^3 + 12b_y t^2 + 6c_y t \;\; + 2d_y = d_y = 0$

(vi) $\ddot{Y}_e(t=10) = 20a_y 10^3 + 12b_y 10^2 + 60c_y + 2d_y = 0$

Constructing these set of equations in matrix form we get,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 10^5 & 10^4 & 10^3 & 10^2 & 10 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5*10^4 & 4*10^3 & 3*10^2 & 20 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20*10^3 & 12*10^2 & 60 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_y \\ b_y \\ c_y \\ d_y \\ e_y \\ f_y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

```
clear all
%%solving for these system of equations
AY = [[0 0 0 0 0 1];
     [10^5 10^4 10^3 10^2 10 1];
     [0 0 0 0 1 0];
     [5*10^4 4*10^3 3*10^2 20 1 0];
     [0 0 0 1 0 0];
     [20*10^3 12*10^2 60 2 0 0]];
BY = [1;1;0;0;0;0];
Y = AY\BY;
```

Similarly considering the equation of $X_e(t)$, we have 6 equations and following are the 6 equations needed to find the coeffecients,

(i) $X_e(t=0) = f_x = -1$

(ii) $X_e(t=10) = \mathrm{a}_x 10^5 + \mathrm{b}_x 10^4 + \mathrm{c}_x 10^3 + \mathrm{d}_x 10^2 + \mathrm{e}_x 10 + f_x = 2$

(iii) $\dot{X}_e(t=0) = 5a_x t^4 + 4b_x t^3 + 3c_x t^2 + 2d_x t \ \ + e_x = e_x = 0$

(iv) $\dot{X}_e(t=10) = 5a_x 10^4 + 4b_x 10^3 + 3c_x 10^2 + 20d_x + e_x = 0$

(v) $\ddot{X}_e(t=0) = 20a_x t^3 + 12b_x t^2 + 6c_x t \ \ + 2d_x = d_x = 0$

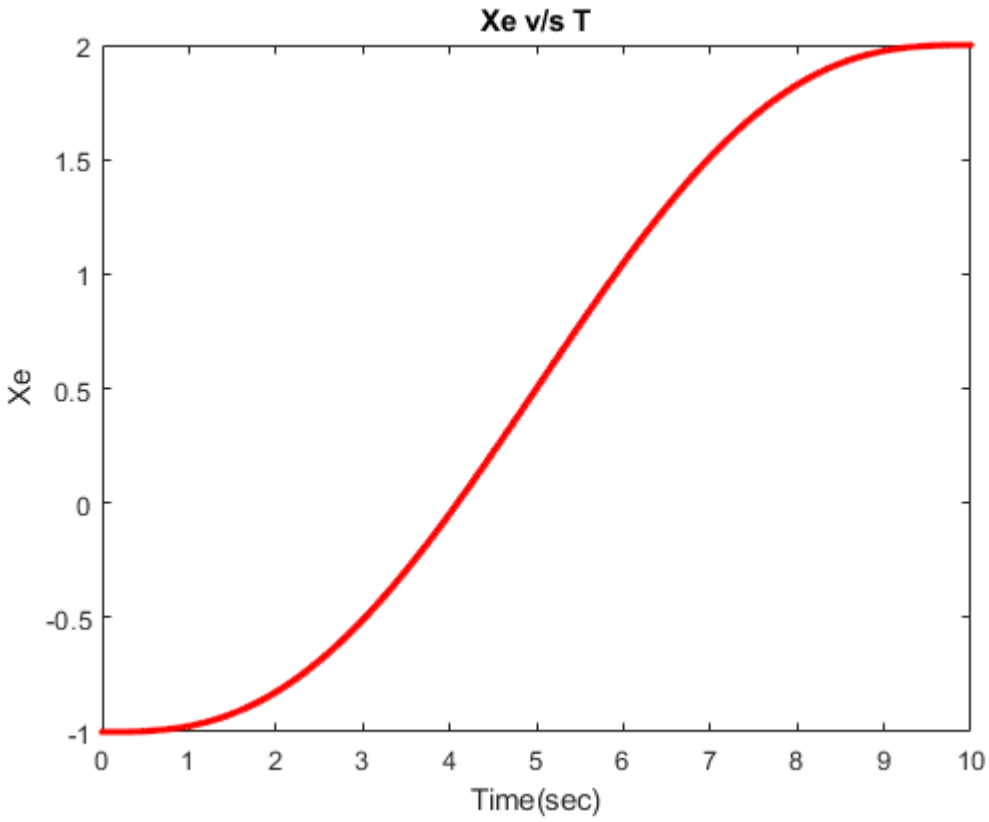
(vi) $\ddot{X}_e(t=10) = 20a_x 10^3 + 12b_x 10^2 + 60c_x + 2d_x = 0$

Constructing these set of equations in matrix form we get,

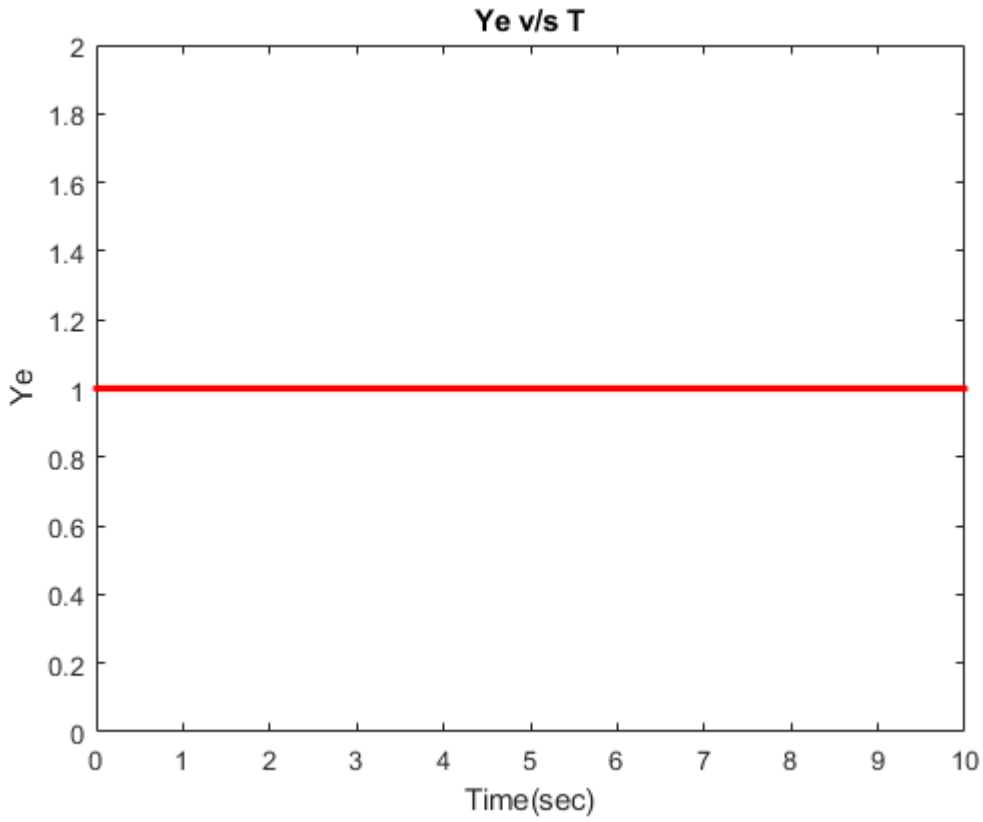
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 10^5 & 10^4 & 10^3 & 10^2 & 10 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5*10^4 & 4*10^3 & 3*10^2 & 20 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20*10^3 & 12*10^2 & 60 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \\ e_x \\ f_x \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

```
%%solving for these system of equations
AX = [[0 0 0 0 0 1];
     [10^5 10^4 10^3 10^2 10 1];
     [0 0 0 0 1 0];
     [5*10^4 4*10^3 3*10^2 20 1 0];
     [0 0 0 1 0 0];
```

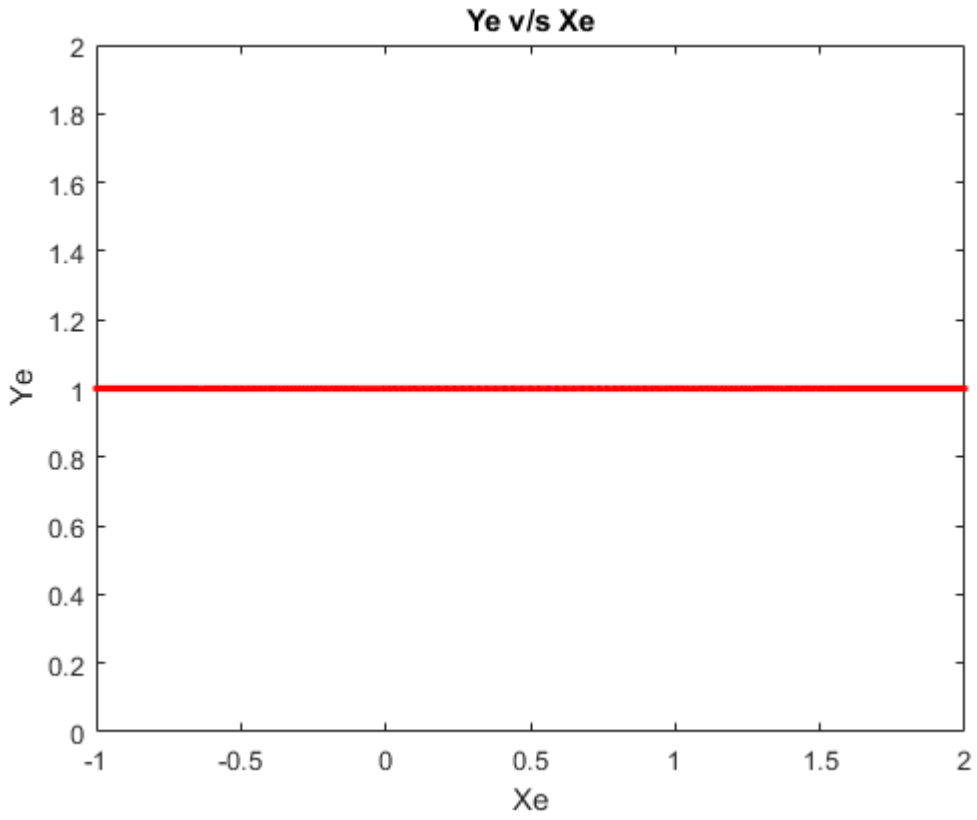
```
     [20*10^3 12*10^2 60 2 0 0]];
BX = [-1;2;0;0;0;0];
X = AX\BX;

%using the quintic equations to estimate the position
t = (0:0.01:10)';
L = [2 1];
Xe = zeros(length(t),1);
Ye = zeros(length(t),1);
for i = (1:1:length(t))
    Xe(i,1) = X(1,1)*(t(i,1))^5+X(2,1)*(t(i,1))^4+X(3,1)*(t(i,1))^3+X(4,1)*(t(i,1))^2+X(5,1)*(t
    Ye(i,1) = Y(1,1)*(t(i,1))^5+Y(2,1)*(t(i,1))^4+Y(3,1)*(t(i,1))^3+Y(4,1)*(t(i,1))^2+Y(5,1)*(t
end

%finding the theta1 and theta2 values at these end effector positions
theta1_u = zeros(length(t),1);
theta1_c = zeros(length(t),1);
theta2_u = zeros(length(t),1);
theta2_c = zeros(length(t),1);

[theta1_u,theta2_u] = invkinema(L,Xe,Ye,1);
[theta1_c,theta2_c] = invkinema(L,Xe,Ye,-1);

%plotting the results

%(Ci) Xe v/s t
figure(15);
plot(t,Xe,'r.')
title('Xe v/s T');
xlabel('Time(sec)');
ylabel('Xe');
```



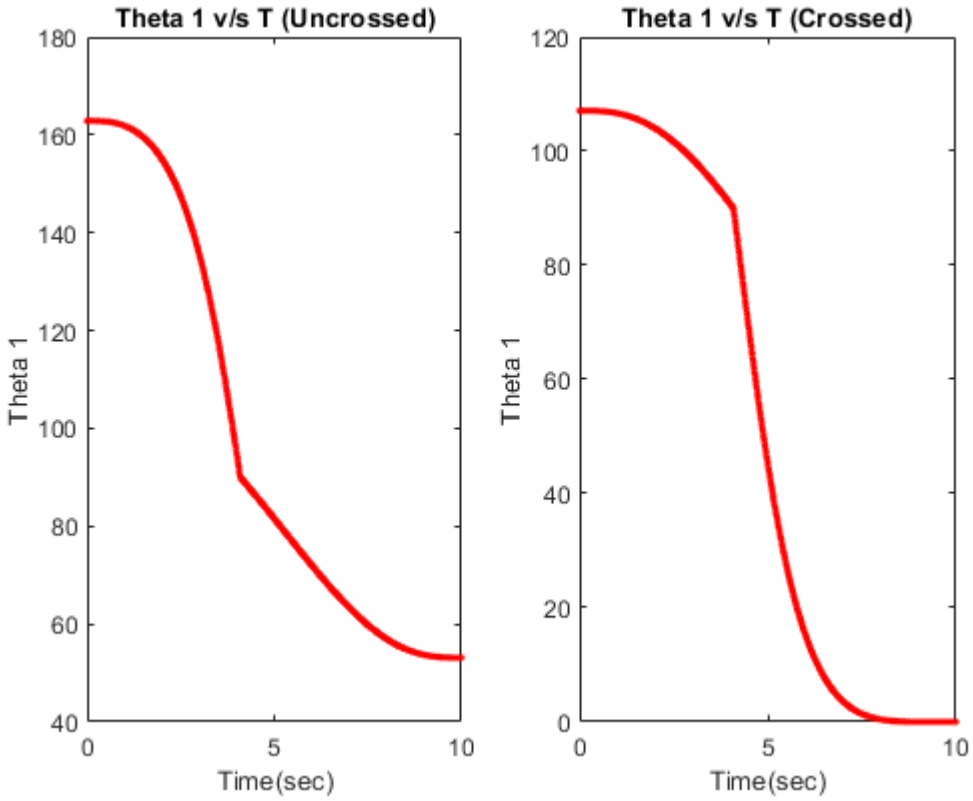

```
%(Cii) Ye v/s t
figure(16);
plot(t,Ye,'r.')
title('Ye v/s T');
xlabel('Time(sec)');
ylabel('Ye');
```



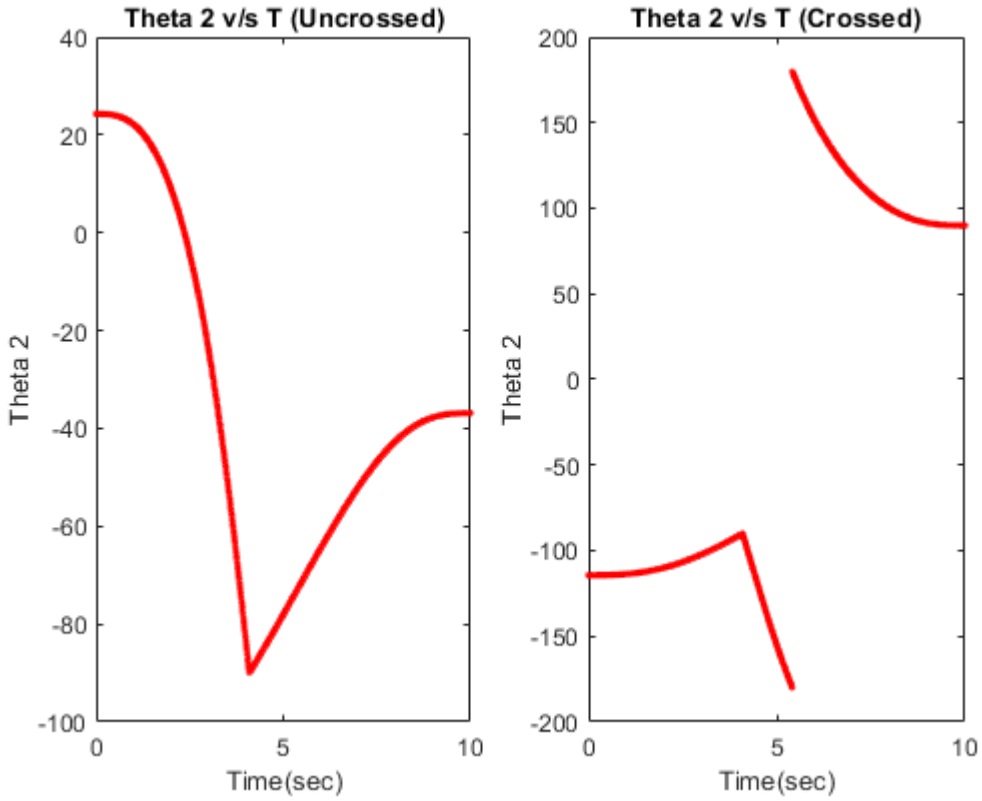

```
%(Ciii) Ye v/s Xe
figure(17);
plot(Xe,Ye,'r.')
title('Ye v/s Xe');
xlabel('Xe');
ylabel('Ye');
```



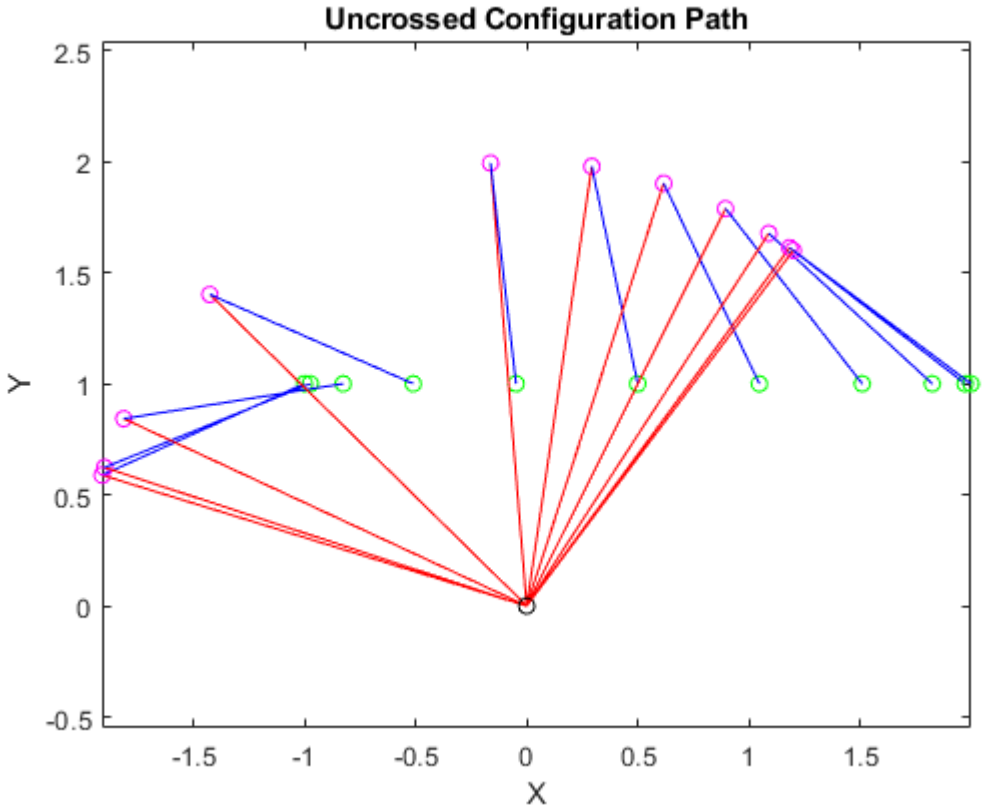

```
%(Civ) theta 1 v/s t in crossed and uncrossed condition
figure(18);
subplot(1,2,1)
plot(t,theta1_u,'r.')
title('Theta 1 v/s T (Uncrossed)');
xlabel('Time(sec)');
ylabel('Theta 1');
subplot(1,2,2)
plot(t,theta1_c,'r.')
title('Theta 1 v/s T (Crossed)');
xlabel('Time(sec)');
ylabel('Theta 1');
```



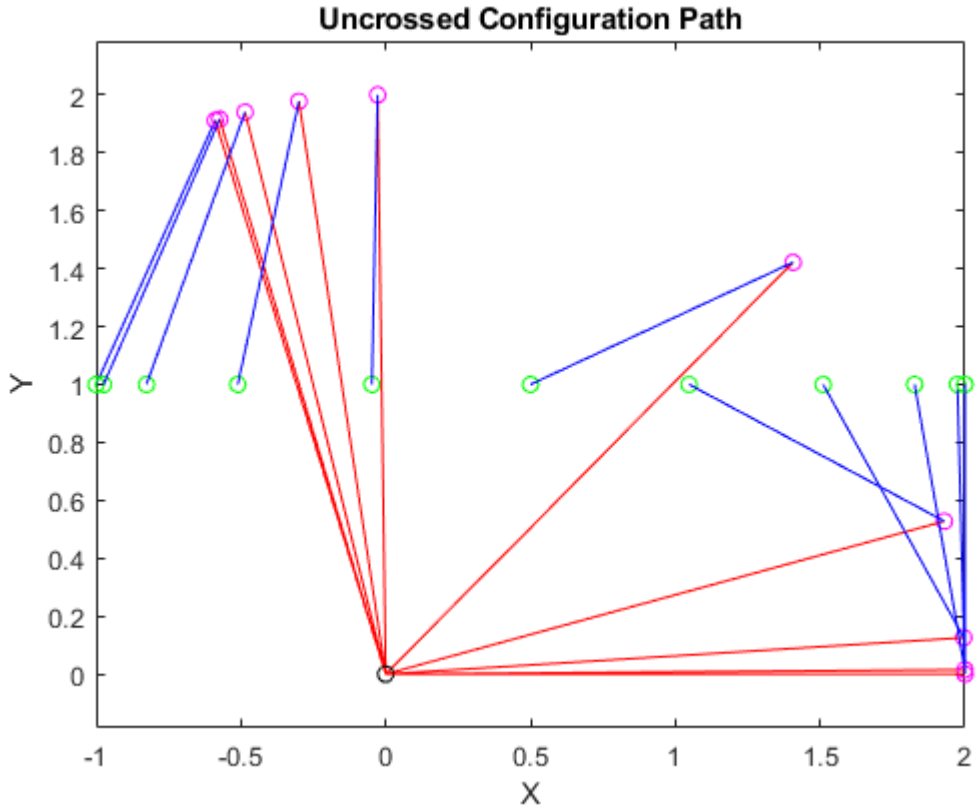

```
%(Biv) theta 2 v/s t in crossed and uncrossed condition
figure(19);
subplot(1,2,1)
plot(t,theta2_u,'r.')
title('Theta 2 v/s T (Uncrossed)');
xlabel('Time(sec)');
ylabel('Theta 2');
subplot(1,2,2)
plot(t,theta2_c,'r.')
title('Theta 2 v/s T (Crossed)');
xlabel('Time(sec)');
ylabel('Theta 2');
```




```
%% tracing the movement of the linkages

%uncrossed configuration
J2_ux = L(1)*cosd(theta1_u);
J2_uy = L(1)*sind(theta1_u);
j2_u = [J2_ux J2_uy];
figure(20);  %uncrossed path
for i = (1:100:length(J2_ux))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(j2_u(i,1),j2_u(i,2),'mo');
    hold on
    plot(Xe(i,1),Ye(i,1),'go');
    hold on
    plot([0,j2_u(i,1)],[0,j2_u(i,2)],'r-');   %first link positions
    hold on
    plot([j2_u(i,1),Xe(i,1)],[j2_u(i,2),Ye(i,1)],'b-')  %second link positions
    title('Uncrossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```




```
%crossed configuration
J2_cx = L(1)*cosd(theta1_c);
J2_cy = L(1)*sind(theta1_c);
j2_c = [J2_cx J2_cy];
figure(21);  %crossed path
for i = (1:100:length(J2_cx))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(j2_c(i,1),j2_c(i,2),'mo');
    hold on
    plot(Xe(i,1),Ye(i,1),'go');
    hold on
    plot([0,j2_c(i,1)],[0,j2_c(i,2)],'r-');   %first link positions
    hold on
    plot([j2_c(i,1),Xe(i,1)],[j2_c(i,2),Ye(i,1)],'b-')  %second link positions
    title('Crossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```



(D) **Manipulator starts from rest and stops at rest, trajectory described by a quintic polynomial in joint coordinates**

The equations of motion of this system can be given as,

$$\theta_1(t) = a_1t^5 + b_1t^4 + c_1t^3 + d_1t^2 + e_1t + f_1$$
$$\theta_2(t) = a_2t^5 + b_2t^4 + c_2t^3 + d_2t^2 + e_2t + f_2$$

First considering the equation of $\theta_1(t)$, we have 6 equations and following are the 6 equations needed to find the coeffecients,

(i) $\theta_1(t=0) = f_1 = \text{theta1 @ } t = 0 \text{ (obtained using the invkinema function)}$

(ii) $\theta_1(t=10) = a_1 10^5 + b_1 10^4 + c_1 10^3 + d_1 10^2 + e_1 10 + f_1 = \text{theta1 @ } t = 10 \text{ (obtained using invkinema function)}$

(iii) $\dot{\theta}_1(t=0) = 5a_1t^4 + 4b_1t^3 + 3c_1t^2 + 2d_1t \ + e_1 = e_1 = 0$

(iv) $\dot{\theta}_1(t=10) = 5a_1 10^4 + 4b_1 10^3 + 3c_1 10^2 + 20d_1 + e_1 = 0$

(v) $\ddot{\theta}_1(t=0) = 20a_1t^3 + 12b_1t^2 + 6c_1t \ + 2d_1 = d_1 = 0$

(vi) $\ddot{\theta}_1(t=10) = 20a_1 10^3 + 12b_1 10^2 + 60c_1 + 2d_1 = 0$

```
clear all
%obtaining the values of theta1 & theta2 @ crossed and uncrossed conditions
```

```
p0  = [-1 1];
p10 = [2 1];
L = [2 1];
[theta1_u0,theta2_u0] = invkinema(L,p0(1),p0(2),1);
[theta1_c0,theta2_c0] = invkinema(L,p0(1),p0(2),-1);
[theta1_u10,theta2_u10] = invkinema(L,p10(1),p10(2),1);
[theta1_c10,theta2_c10] = invkinema(L,p10(1),p10(2),-1);
```

Constructing these set of equations for the uncrossed configuration in matrix form we get,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 10^5 & 10^4 & 10^3 & 10^2 & 10 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5*10^4 & 4*10^3 & 3*10^2 & 20 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20*10^3 & 12*10^2 & 60 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{1u} \\ b_{1u} \\ c_{1u} \\ d_{1u} \\ e_{1u} \\ f_{1u} \end{bmatrix} = \begin{bmatrix} \text{theta1\_u0} \\ \text{theta1\_u10} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Similarly for the crossed configuration,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 10^5 & 10^4 & 10^3 & 10^2 & 10 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5*10^4 & 4*10^3 & 3*10^2 & 20 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20*10^3 & 12*10^2 & 60 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{1c} \\ b_{1c} \\ c_{1c} \\ d_{1c} \\ e_{1c} \\ f_{1c} \end{bmatrix} = \begin{bmatrix} \text{theta1\_c0} \\ \text{theta1\_c10} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

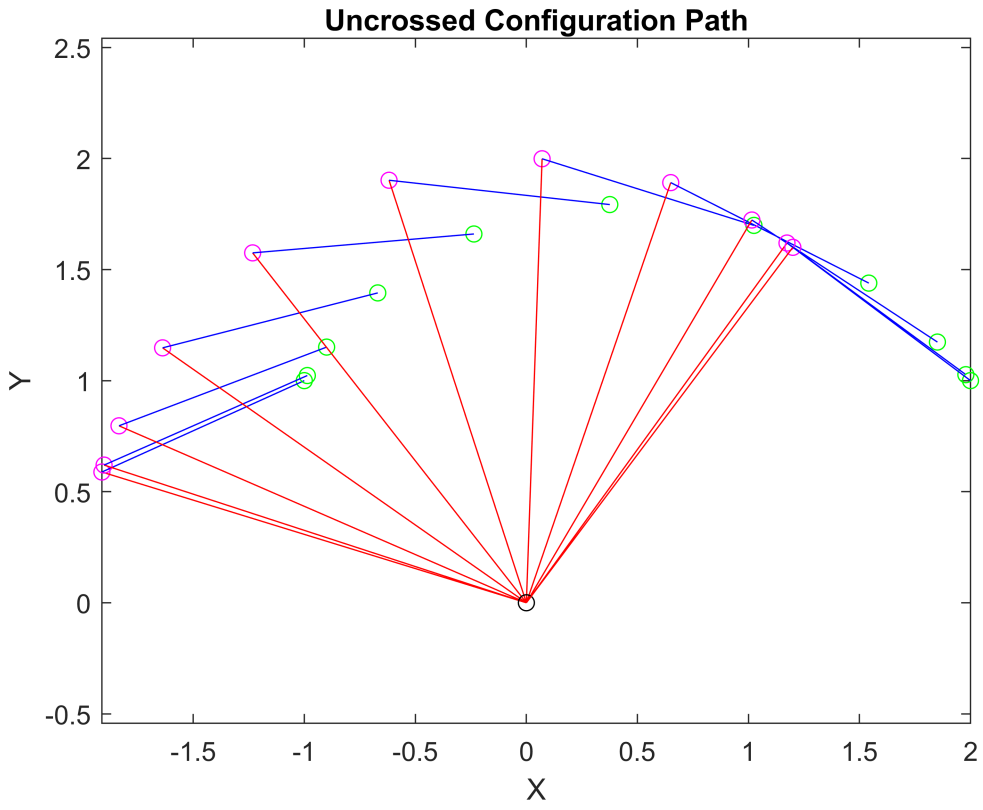
The theta 2 equations for the two configurations would be,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 10^5 & 10^4 & 10^3 & 10^2 & 10 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5*10^4 & 4*10^3 & 3*10^2 & 20 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20*10^3 & 12*10^2 & 60 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{2u} \\ b_{2u} \\ c_{2u} \\ d_{2u} \\ e_{2u} \\ f_{2u} \end{bmatrix} = \begin{bmatrix} \text{theta2\_u0} \\ \text{theta2\_u10} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

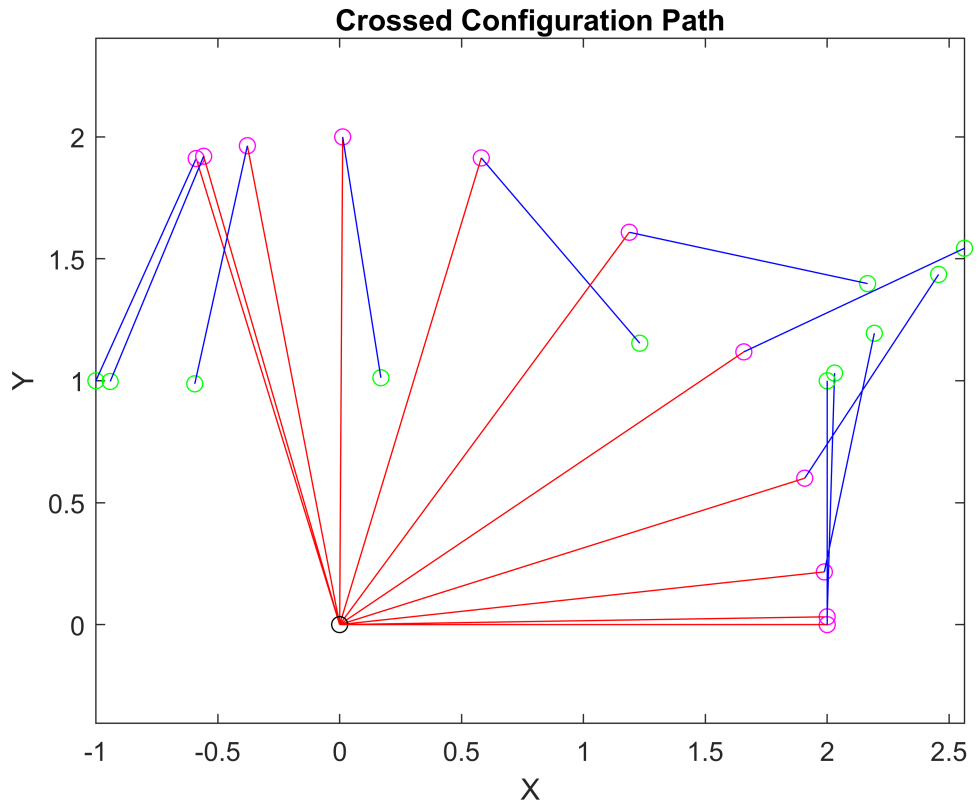
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 10^5 & 10^4 & 10^3 & 10^2 & 10 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5*10^4 & 4*10^3 & 3*10^2 & 20 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20*10^3 & 12*10^2 & 60 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{2c} \\ b_{2c} \\ c_{2c} \\ d_{2c} \\ e_{2c} \\ f_{2c} \end{bmatrix} = \begin{bmatrix} \text{theta2\_c0} \\ \text{theta2\_c10} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

```
% solving the system of equations for all the 4 conditions,
%uncrossed theta1
A = [[0 0 0 0 0 1];
     [10^5 10^4 10^3 10^2 10 1];
```

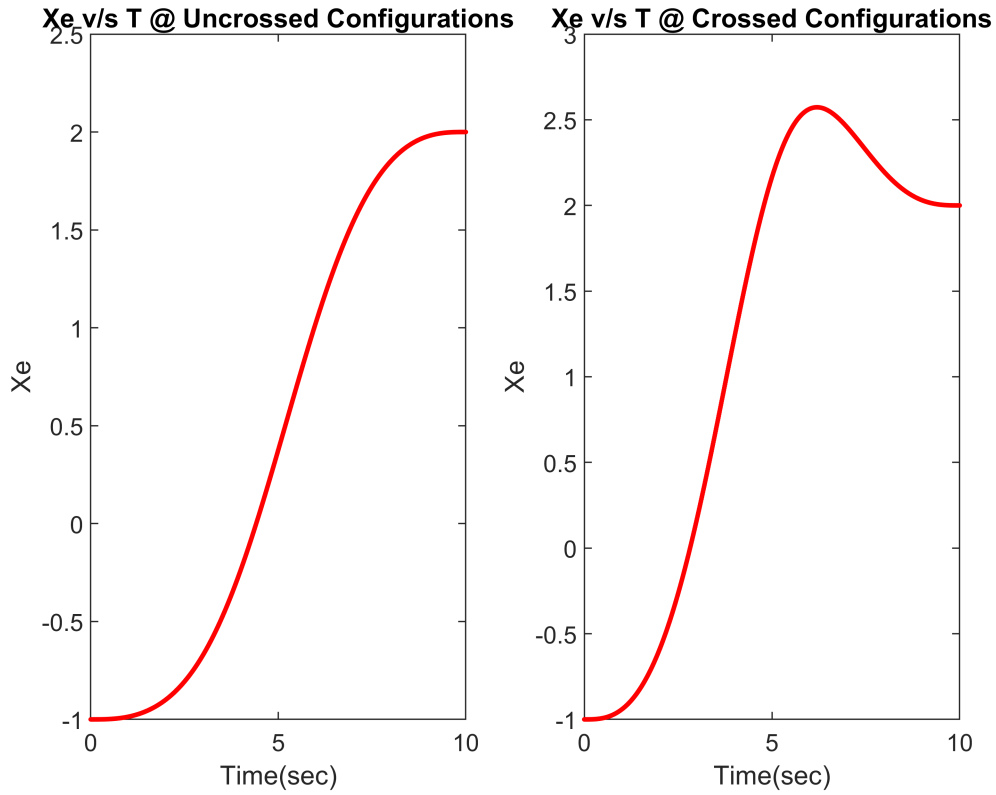
```
     [0 0 0 0 1 0];
     [5*10^4 4*10^3 3*10^2 20 1 0];
     [0 0 0 1 0 0];
     [20*10^3 12*10^2 60 2 0 0]];
B_theta1_u = [theta1_u0;theta1_u10;0;0;0;0];
B_theta1_c = [theta1_c0;theta1_c10;0;0;0;0];
B_theta2_u = [theta2_u0;theta2_u10;0;0;0;0];
B_theta2_c = [theta2_c0;theta2_c10;0;0;0;0];

%coeffecients
C_theta1_u = A\B_theta1_u;
C_theta1_c = A\B_theta1_c;
C_theta2_u = A\B_theta2_u;
C_theta2_c = A\B_theta2_c;

%theta values at different instants
t = (0:0.01:10)';

for i = (1:1:length(t))
    theta1_u(i,1) = C_theta1_u(1,1)*(t(i,1))^5+C_theta1_u(2,1)*(t(i,1))^4+C_theta1_u(3,1)*(t(i,
    theta1_c(i,1) = C_theta1_c(1,1)*(t(i,1))^5+C_theta1_c(2,1)*(t(i,1))^4+C_theta1_c(3,1)*(t(i,
    theta2_u(i,1) = C_theta2_u(1,1)*(t(i,1))^5+C_theta2_u(2,1)*(t(i,1))^4+C_theta2_u(3,1)*(t(i,
    theta2_c(i,1) = C_theta2_c(1,1)*(t(i,1))^5+C_theta2_c(2,1)*(t(i,1))^4+C_theta2_c(3,1)*(t(i,
end

%uncrossed joint and end effector coordinates
J2_xu = L(1)*cosd(theta1_u);
J2_yu = L(1)*sind(theta1_u);
ee_xu = L(1)*cosd(theta1_u) + L(2)*cosd(theta2_u);
ee_yu = L(1)*sind(theta1_u) + L(2)*sind(theta2_u);

%crossed joint and end effector coordinates
J2_xc = L(1)*cosd(theta1_c);
J2_yc = L(1)*sind(theta1_c);
ee_xc = L(1)*cosd(theta1_c) + L(2)*cosd(theta2_c);
ee_yc = L(1)*sind(theta1_c) + L(2)*sind(theta2_c);

%%plotting the results
figure(22);  %uncrossed path
for i = (1:100:length(J2_xu))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(J2_xu(i,1),J2_yu(i,1),'mo');
    hold on
    plot(ee_xu(i,1),ee_yu(i,1),'go');
    hold on
    plot([0,J2_xu(i,1)],[0,J2_yu(i,1)],'r-');   %first link positions
    hold on
    plot([J2_xu(i,1),ee_xu(i,1)],[J2_yu(i,1),ee_yu(i,1)],'b-')  %second link positions
    title('Uncrossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```



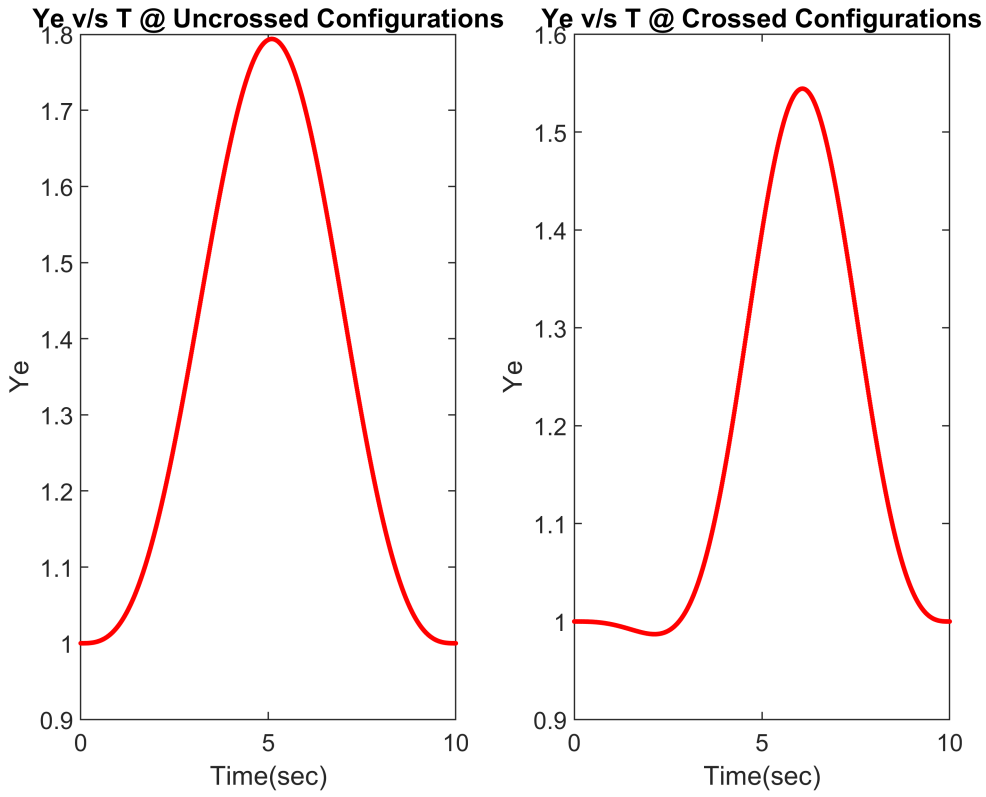

```
figure(23);  %crossed path
for i = (1:100:length(J2_xc))
    plot(0,0,'ko');
    axis equal
    hold on
    plot(J2_xc(i,1),J2_yc(i,1),'mo');
    hold on
    plot(ee_xc(i,1),ee_yc(i,1),'go');
    hold on
    plot([0,J2_xc(i,1)],[0,J2_yc(i,1)],'r-');   %first link positions
    hold on
    plot([J2_xc(i,1),ee_xc(i,1)],[J2_yc(i,1),ee_yc(i,1)],'b-')  %second link positions
    title('Crossed Configuration Path');
    xlabel('X');
    ylabel('Y');
end
```



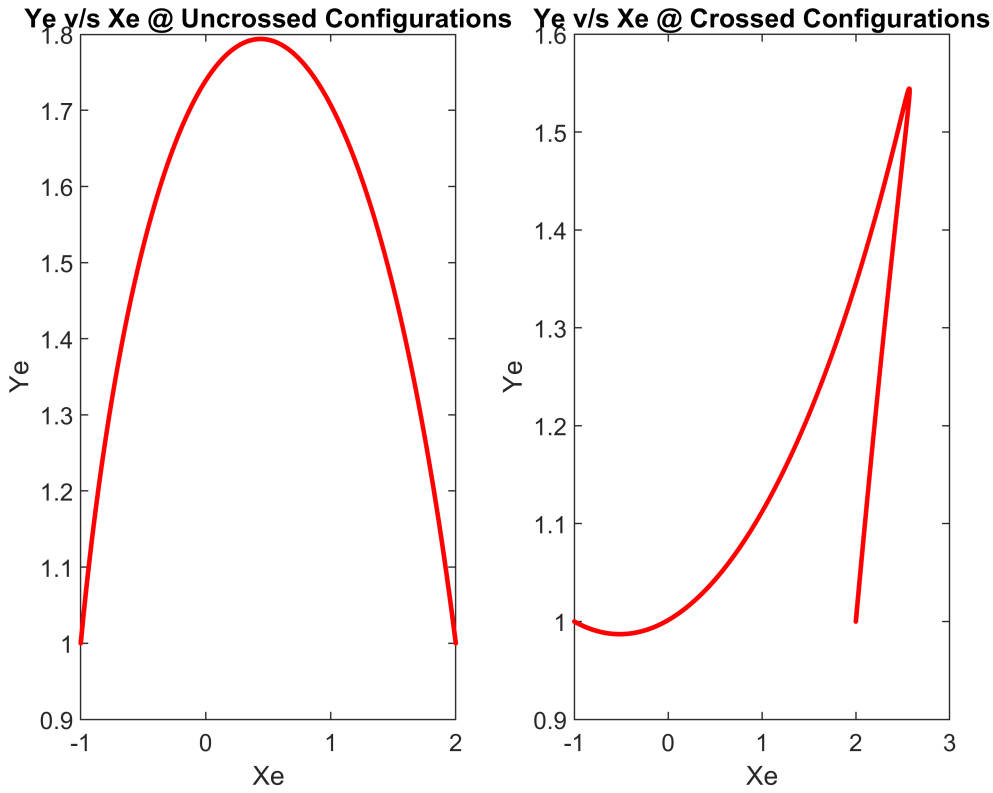

```
%(Di) Xe v/s t
figure(24);
subplot(1,2,1)
plot(t,ee_xu,'r.')
title('Xe v/s T @ Uncrossed Configurations');
xlabel('Time(sec)');
ylabel('Xe');
subplot(1,2,2)
plot(t,ee_xc,'r.')
title('Xe v/s T @ Crossed Configurations');
xlabel('Time(sec)');
ylabel('Xe');
```



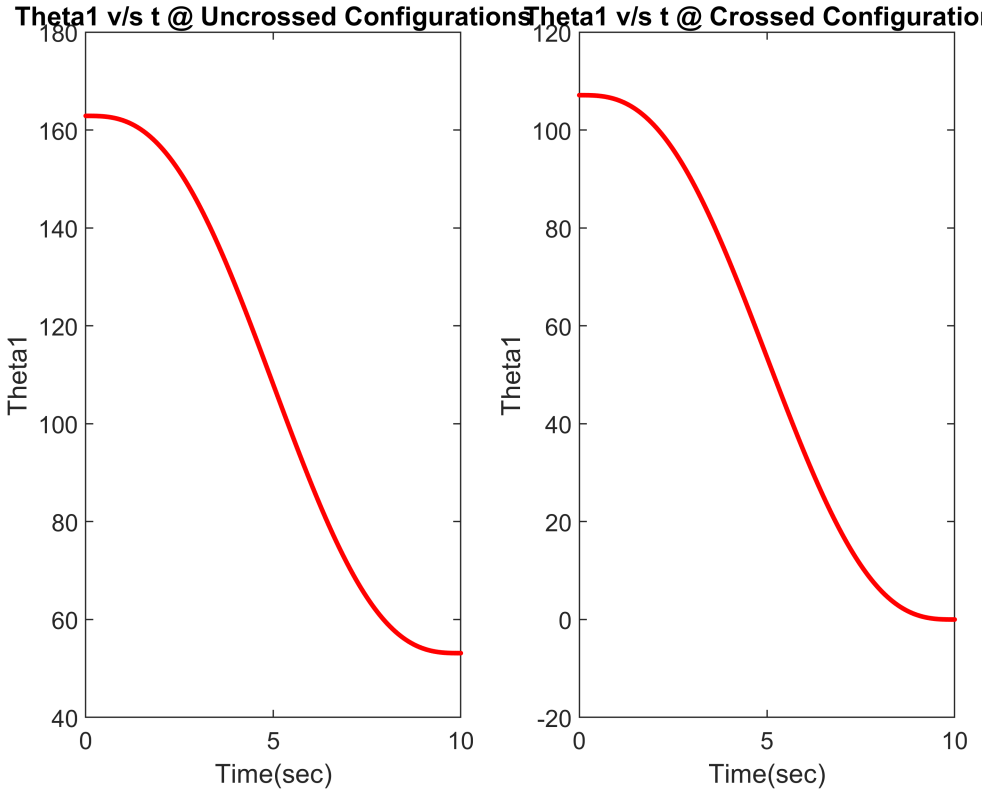

```
%(Dii) Ye v/s t
figure(25);
subplot(1,2,1)
plot(t,ee_yu,'r.')
title('Ye v/s T @ Uncrossed Configurations');
xlabel('Time(sec)');
ylabel('Ye');
subplot(1,2,2)
plot(t,ee_yc,'r.')
title('Ye v/s T @ Crossed Configurations');
xlabel('Time(sec)');
ylabel('Ye');
```



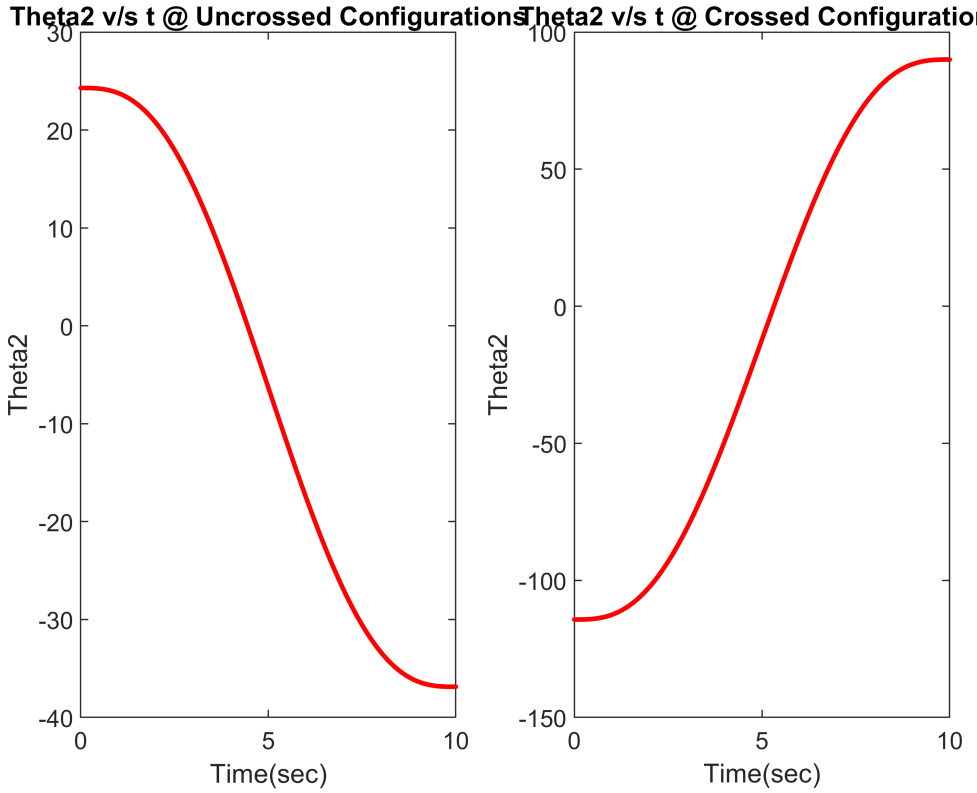

```
%(Diii) Ye v/s Xe
figure(26);
subplot(1,2,1)
plot(ee_xu,ee_yu,'r.')
title('Ye v/s Xe @ Uncrossed Configurations');
xlabel('Xe');
ylabel('Ye');
subplot(1,2,2)
plot(ee_xc,ee_yc,'r.')
title('Ye v/s Xe @ Crossed Configurations');
xlabel('Xe');
ylabel('Ye');
```



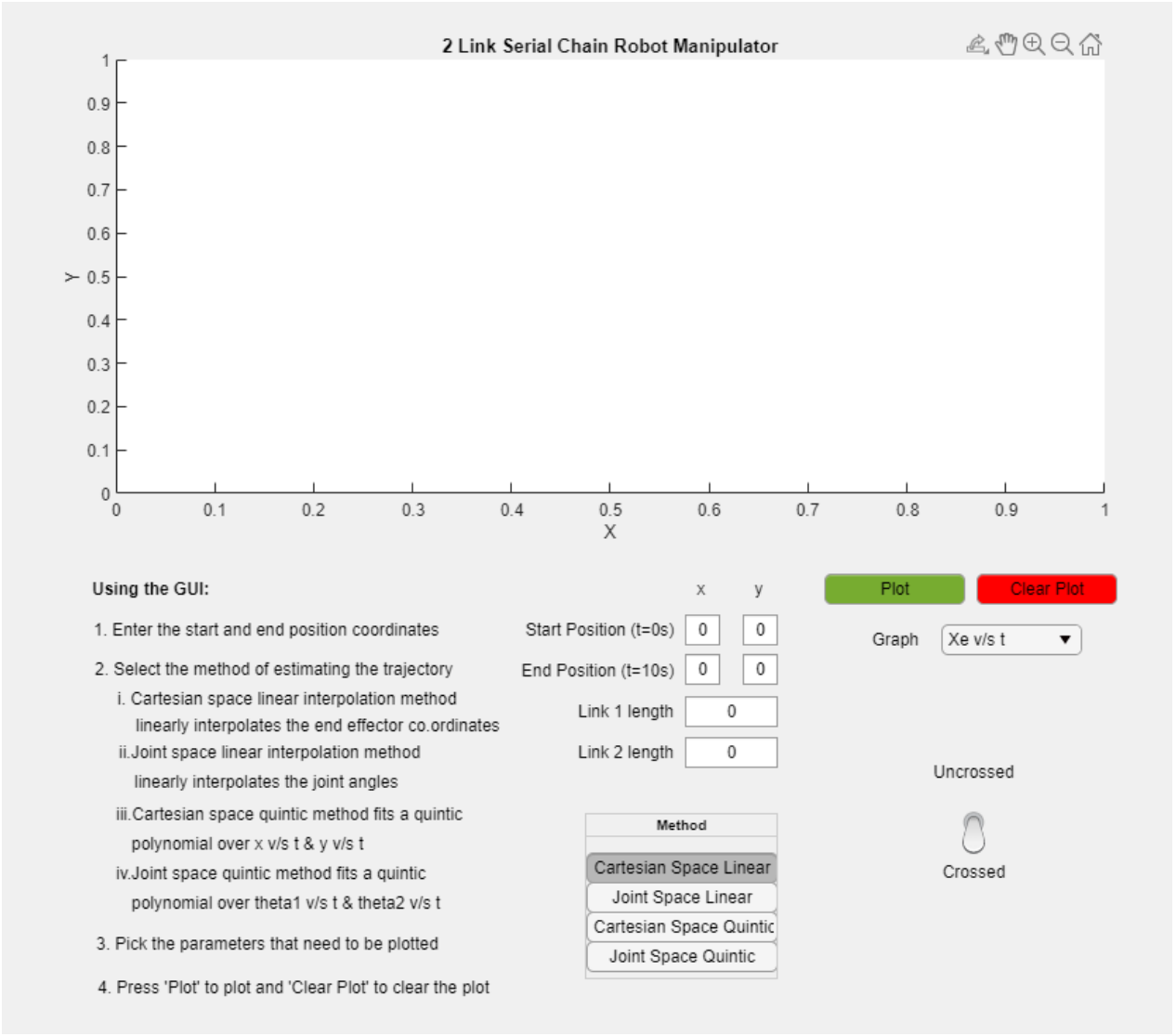

```
%(Div) theta1 v/s t
figure(27);
subplot(1,2,1)
plot(t,theta1_u,'r.')
title('Theta1 v/s t @ Uncrossed Configurations');
xlabel('Time(sec)');
ylabel('Theta1');
subplot(1,2,2)
plot(t,theta1_c,'r.')
title('Theta1 v/s t @ Crossed Configurations');
xlabel('Time(sec)');
ylabel('Theta1');
```




```
%(Dv) theta2 v/s t
figure(28);
subplot(1,2,1)
plot(t,theta2_u,'r.')
title('Theta2 v/s t @ Uncrossed Configurations');
xlabel('Time(sec)');
ylabel('Theta2');
subplot(1,2,2)
plot(t,theta2_c,'r.')
title('Theta2 v/s t @ Crossed Configurations');
xlabel('Time(sec)');
ylabel('Theta2');
```




```
run('Q2_GUI.mlapp');
```



NOTE: While using the GUI, kindly change the values for 'Method','Graph', and the configuration switch even if you want to choose the default settings. For e.g., if you want to use 'Cartesian Space Linear' method, plot 'Xe v/s t' for the uncrossed configuration, change these parameters and select these parameters once again. The default values are not fed to the program.

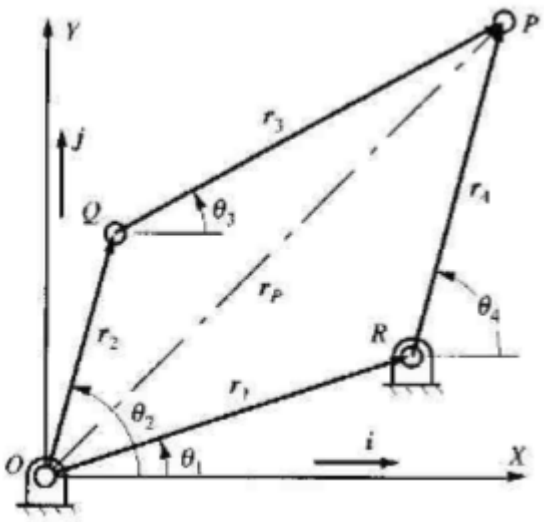
PROBLEM 3

Problem 3.1:

Assume a four bar mechanism with the notation as:

$r_1 = \text{Ground Link}\ ;\ r_2 = \text{Input Link}\ ;\ r_3 = \text{Coupler}\ ;\ r_4 = \text{Follower Link}\ ;\ r_p = \text{Vector from point "O" to "P"}$

with their corresponding angles as $\theta_1\ ;\theta_2\ ;\theta_3\ ;\theta_4$ as shown below:



.

Assuming loop closure equation on the four bar mechanism:

$$r_p = r_1 + r_4 = r_2 + r_3$$

This can be further split into the components of each vector as:

$$r_1 * \left(\cos\theta_1 \hat{i} + \sin\theta_1 \hat{j}\right) + r_4 * \left(\cos\theta_4 \hat{i} + \sin\theta_4 \hat{j}\right) = r_2 * \left(\cos\theta_2 \hat{i} + \sin\theta_2 \hat{j}\right) + r_3 * \left(\cos\theta_3 \hat{i} + \sin\theta_3 \hat{j}\right)$$

On equating $\hat{i}$ & $\hat{j}$ components of LHS & RHS, we get:

$$r_1 * \cos\theta_1 + r_4 * \cos\theta_4 = r_2 * \cos\theta_2 + r_3 * \cos\theta_3$$

$$r_1 * \sin\theta_1 + r_4 * \sin\theta_4 = r_2 * \sin\theta_2 + r_3 * \sin\theta_3$$

In the above equations, the known variables are $r_1, r_2, r_3, r_4, \theta_1, \theta_2$, and unknown variables are θ_3, θ_4 . Therefore we will isolate 1 unknown variable to eliminate it as shown

$$r_3 * \cos\theta_3 = r_1 * \cos\theta_1 + r_4 * \cos\theta_4 - r_2 * \cos\theta_2$$
$$r_3 * \sin\theta_3 = r_1 * \sin\theta_1 + r_4 * \sin\theta_4 - r_2 * \sin\theta_2$$

Squaring and adding both the above equations we get:

$$)^2 + (r_2)^2 + (r_4)^2 + 2 * r_1 * r_4 * (\cos\theta_1 * \cos\theta_4 + \sin\theta_1 * \sin\theta_4) - 2 * r_1 * r_2 * (\cos\theta_1 * \cos\theta_2 + \sin\theta_1 * \sin\theta_2) - 2 * r_2 * r_4 * (\cos\theta_2 * \cos\theta_4 + \sin\theta$$

On re-arranging and writing the above equation in terms of the unknown (θ_4) of the format:
$A * \cos\theta_4 + B * \sin\theta_4 + C = 0$, where the co-efficients can be defined as:

$$A = 2 * r_1 * r_4 * \cos\theta_1 - 2 * r_2 * r_4 * \cos\theta_2$$
$$B = 2 * r_1 * r_4 * \sin\theta_1 - 2 * r_2 * r_4 * \sin\theta_2$$
$$C = (r_1)^2 + (r_2)^2 + (r_4)^2 - (r_3)^2 - 2 * r_1 * r_2 * (\cos\theta_1 * \cos\theta_2 + \sin\theta_1 * \sin\theta_2)$$

To solve the equation $A * \cos\theta_4 + B * \sin\theta_4 + C = 0$, we use trignometric substitution of half angle formula given as:

$$\sin\theta_4 = \frac{2 * \tan\frac{\theta_4}{2}}{1 + \left(\tan\frac{\theta_4}{2}\right)^2}$$

$$\cos\theta_4 = \frac{1 - \left(\tan\frac{\theta_4}{2}\right)^2}{1 + \left(\tan\frac{\theta_4}{2}\right)^2}$$

Using the above equations in $A * \cos\theta_4 + B * \sin\theta_4 + C = 0$; we tranform the equation into :

$$(C - A) * t^2 + 2 * B * t + (A + C) = 0$$

where $t = \tan\frac{\theta_4}{2}$; Solving for $t$gives us the following solution:

$$t = \frac{-B + \sigma\sqrt{B^2 - C^2 + A^2}}{C - A}$$

where $\sigma = +1$ for crossed configuration & $\sigma = -1$ for uncrossed configuration.

From the above solution we know the value of t and can therefore find out θ_4 as:

$$\theta_4 = 2 * \text{atan2}\left(-B + \sigma\sqrt{B^2 - C^2 + A^2}, (C - A)\right)$$

Once θ_4 is known, we can compute θ_3 as follows:

$$\theta_3 = \text{atan2}((r_1 * \sin\theta_1 + r_4 * \sin\theta_4 - r_2 * \sin\theta_2), (r_1 * \cos\theta_1 + r_4 * \cos\theta_4 - r_2 * \cos\theta_2))$$

Know all the variables are known, we can write and define the position of each point on the four bar mechanism shown in the figure above as:

$$O_x = 0$$
$$O_y = 0$$

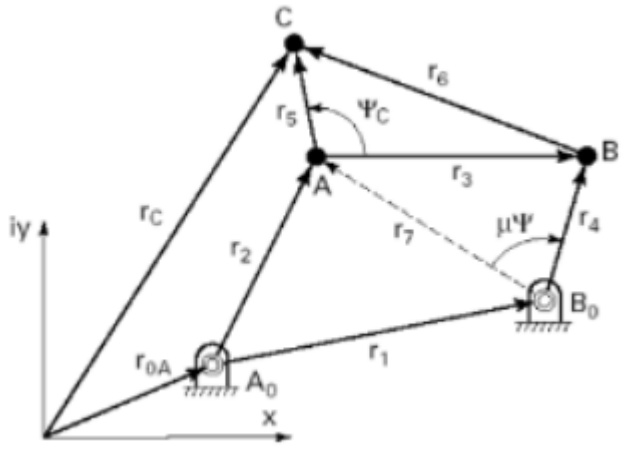
$$R_x = O_x + r_1 * \cos\theta_1$$
$$R_y = O_y + r_1 * \sin\theta_1$$

$$Q_x = O_x + r_2 * \cos\theta_2$$
$$Q_y = O_y + r_2 * \sin\theta_2$$

$$P_x = Q_x + r_3 * \cos\theta_3$$
$$P_y = Q_y + r_3 * \sin\theta_3$$

We can then plot the points O, P, Q, R & S for any given link lenghts and angle.

For the particular four bar mechanism in question 3 as shown:



Given information of link:

		Ground Link	Input Link	Coupler	Follower	Coupler Point of Inte
ths	$r_{0A} = 0$	$r_1 = 4.0$	$r_2 = 2.0$	$r_3 = 3.0$	$r_4 = 6.0$	$r_5 = 4$
es	$\theta_0 = 0°$	$\theta_1 = 30°$	θ_2	θ_3	θ_4	Included Angle) ψ_c =

Part 3.A.i) Develop a MATLAB program for the forward kinematic equations developed above given that θ_2 is the input.

Putting the value of all known variable into the program

```
%Lenght of Ground Link:%
r1=4;
%Length of Input Link:%
r2=2;
% Lenght of Coupler: %
r3=3;
%Length of Follower:%
r4=6;
%Rigid Body Side Length: %
r5=4;
%Ground Link Angle: %
theta_1=30;
%Included angle%
rigid_ang=30;
%Desired Input Link angle For example here we put theta_2 = 145%
theta_2=input('Enter desired Input link angle :')
```

```
theta_2 = 2
```

Calculating value of A, B & C for the given inputs

```
A= (2*r1*r4*cosd(theta_1))-(2*r2*r4*cosd(theta_2));
B= (2*r1*r4*sind(theta_1))-(2*r2*r4*sind(theta_2));
```

```
C= r1^2 + r2^2 + r4^2 - r3^2 - 2*r1*r2*(cosd(theta_1)*cosd(theta_2)+sind(theta_1)*sind(theta_2
```

For $t = \frac{-B + \sigma\sqrt{B^2 - C^2 + A^2}}{C - A}$ to yeild an answer, $B^2 - C^2 + A^2$ should be greater than equal to 0

```
cond=B^2-C^2+A^2;
  if cond>=0
      %Calculating values of theta_3 and theta_4 for the crossed configuration
        theta_4= 2*atan2d((-B+sqrt(B^2-C^2+A^2)),(C-A));
        theta_3= atan2d((r1*sind(theta_1)+r4*sind(theta_4)-r2*sind(theta_2)),(r1*cosd(theta_1)-

      % Calculating the position of all points
        Ao_x = 0;
        Ao_y = 0;

        Ax = Ao_x+r2*cosd(theta_2);
        Ay = Ao_y+r2*sind(theta_2);

        Bo_x = r1*cosd(theta_1);
        Bo_y = r1*sind(theta_1);

        Bx = Ao_x+Ax + r3*cosd(theta_3);
        By = Ao_y+Ay + r3*sind(theta_3);

        % The co-ordinates of the point "C" on the rigid body are defined
        % with respect to the point "A"
        Cx= Ax+r5*cosd(theta_3+30);
        Cy= Ay+r5*sind(theta_3+30);

        %Plotting configuration of 4 bar mechanism
        figure
        plot([Ao_y Ax], [Ao_y Ay], [Ax Bx], [Ay By],[Bx Bo_x], [By Bo_y], [Cx Ax], [Cy Ay ], [C
        %Link Ao-A = blue; Link A-B = Orange; Link B-Bo = Yellow; Link
        %A-C=purple, Link C-B=Green
        title('crossed configuration')
        grid on;
        L=[-7,7,-7,7];
        axis(L);
```

Calculating value for the uncrossed configuration

```
        %CALCULATING VALUE OF theta_3 AND theta_4 FOR UNCROSSED CONFIGURATION
        theta_4= 2*atan2d((-B-sqrt(B^2-C^2+A^2)),(C-A));
        theta_3= atan2d((r1*sind(theta_1)+r4*sind(theta_4)-r2*sind(theta_2)),(r1*cosd(theta_1)-

      % Calculating the position of all points
        UAo_x = 0;
        UAo_y = 0;

        UAx = UAo_x+r2*cosd(theta_2);
        UAy = UAo_y+r2*sind(theta_2);
```

```
        UBo_x = r1*cosd(theta_1);
        UBo_y = r1*sind(theta_1);

        UBx = UAo_x+UAx + r3*cosd(theta_3);
        UBy = UAo_y+UAy + r3*sind(theta_3);

        % The co-ordinates of the point "C" on the rigid body are defined
        % with respect to the point "A"
        UCx= UAx+r5*cosd(theta_3+30);
        UCy= UAy+r5*sind(theta_3+30);

        %Plotting configuration of 4 bar  mechanism
        figure
        plot([UAo_y UAx], [UAo_y UAy], [UAx UBx], [UAy UBy],[UBx UBo_x], [UBy UBo_y], [UCx UAx]
        %Link Ao-A = blue; Link A-B = Orange; Link B-Bo = Yellow; Link
        %A-C=purple, Link C-B=Green
        title('uncrossed configuration')
        grid on;
        L=[-7,7,-7,7];
        axis(L);
```

Assigning null values if no values of θ_3 & θ_4 exist for the given θ_2

```
    else
        %Assigning null value to all variables if solution is not possible
        theta_4= NaN;
        theta_3= NaN;

        Cx=NaN;
        Cy=NaN;

        Ao_x = NaN;
        Ao_y = NaN;

        Ax = NaN;
        Ay = NaN;

        Bx = NaN;
        By = NaN;

        Bo_x = NaN;
        Bo_y = NaN;
    end
```

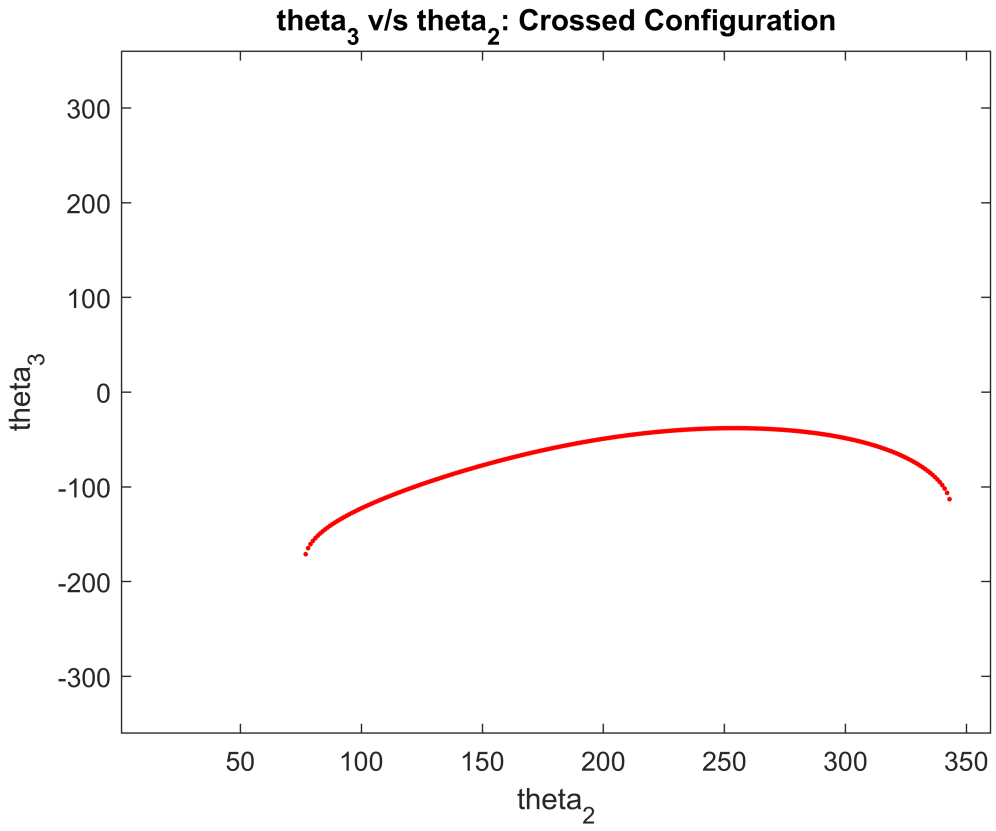
Part 3.A.ii) Then for a range of angles $0^o < \theta_2 < 360^o$ (in degree increments):

```
r_oa=0;
%Lenght of Ground Link:%
r1=4;
```

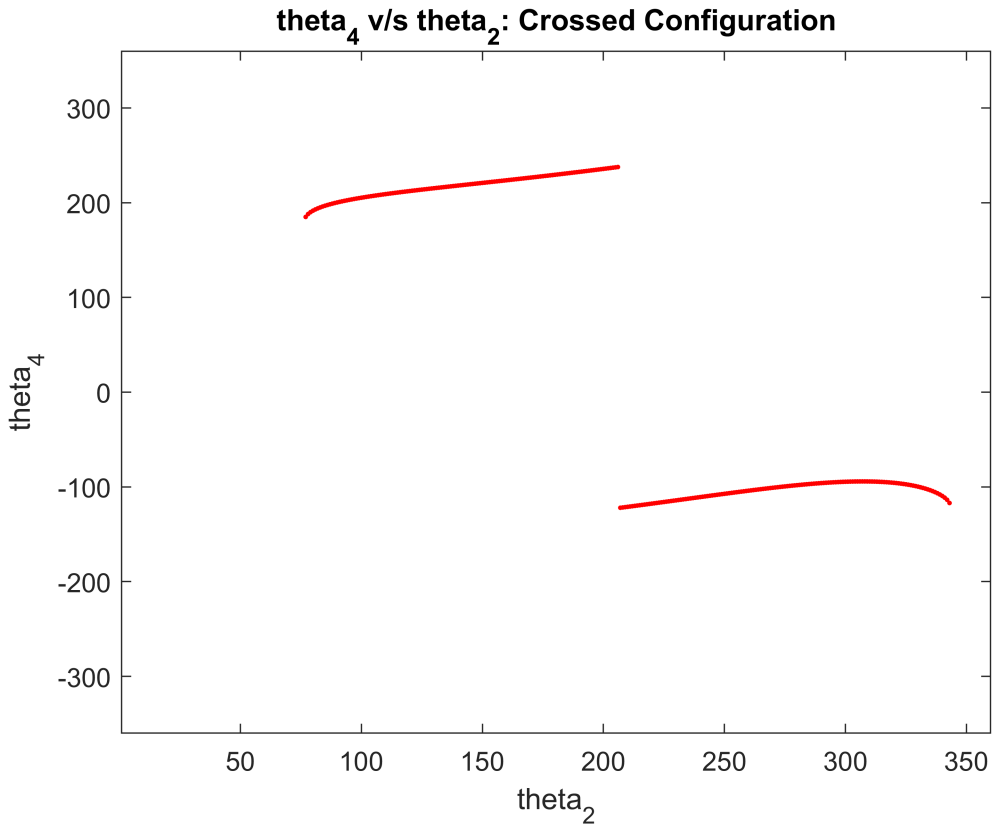
```
%Length of Input Link:%
r2=2;
% Lenght of Coupler: %
r3=3;
%Length of Follower:%
r4=6;
%Rigid Body Side Length: %
r5=4;
%Ground Link Angle: %
theta_1=30;
%Included angle%
rigid_ang=30;
```

Calculating A, B and C parameters fore analytical linkage analysi

```
%Defining theta_2 in increments of 1 degree
for i=1:1:360
    %finding value of A, B & C for each valule as theta_2 where variable
    %theta_2 has been replaced by index i
    A= (2*r1*r4*cosd(theta_1))-(2*r2*r4*cosd(i));
    B= (2*r1*r4*sind(theta_1))-(2*r2*r4*sind(i));
    C= r1^2 + r2^2 + r4^2 - r3^2 - 2*r1*r2*(cosd(theta_1)*cosd(i)+sind(theta_1)*sind(i));
    cond=B^2-C^2+A^2;
%condition for real solution of a theta_4 value existing for a given value
%of theta_1 & theta_2
    if cond>=0
        %calculating value of theta_3 & theta_4 in crossed configuration
        theta_4(i)= 2*atan2d((-B+sqrt(B^2-C^2+A^2)),(C-A));
        theta_3(i)= atan2d((r1*sind(theta_1)+r4*sind(theta_4(i))-r2*sind(i)),(r1*cosd(theta_1)-
    else
        theta_4(i)= NaN;
        theta_3(i)=NaN;
    end
end

        %plotting theta_3 v/s theta_2 for crossed configuration
        figure
        plot(theta_3,'r.','LineWidth',2)
        xlabel('theta_2')
        ylabel('theta_3')
        title('theta_3 v/s theta_2: Crossed Configuration')
        axis([1,360,-360,360])
```

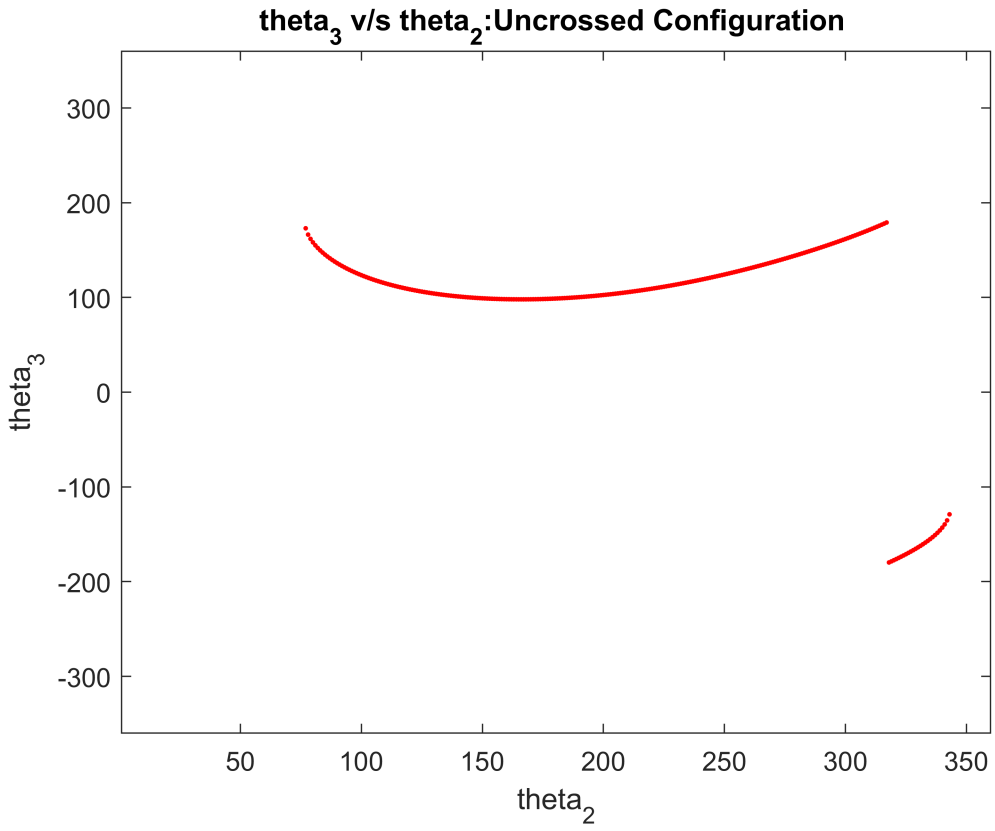



```
%plotting theta_4 v/s theta_2 for Crossed configuration
figure
plot(theta_4,'r.','LineWidth',2)
xlabel('theta_2')
ylabel('theta_4')
title('theta_4 v/s theta_2: Crossed Configuration')
axis([1,360,-360,360])
```

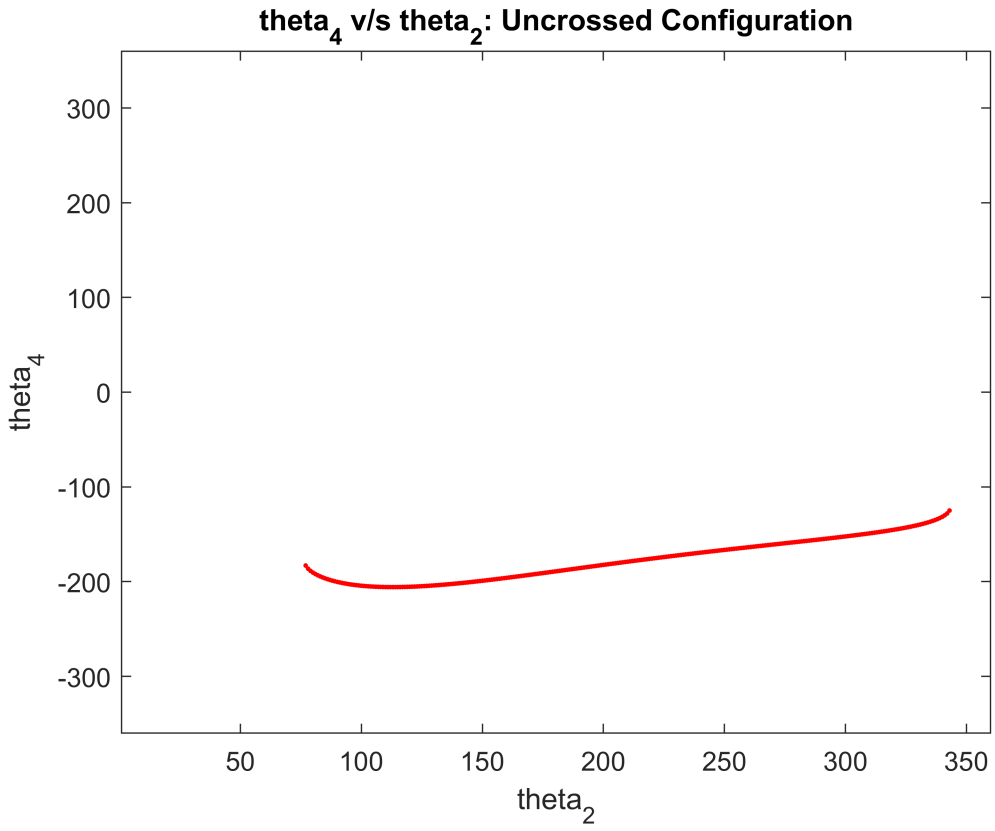


Plotting for crossed configuration

```
   %Defining theta_2 in increments of 1 degree
for i=1:1:360
    %finding value of A, B & C for each valule as theta_2 where variable
    %theta_2 has been replaced by index i
    A= (2*r1*r4*cosd(theta_1))-(2*r2*r4*cosd(i));
    B= (2*r1*r4*sind(theta_1))-(2*r2*r4*sind(i));
    C= r1^2 + r2^2 + r4^2 - r3^2 - 2*r1*r2*(cosd(theta_1)*cosd(i)+sind(theta_1)*sind(i));
    cond=B^2-C^2+A^2;
    %condition for real solution of a theta_4 value existing for a given value
    %of theta_1 & theta_2
    if cond>=0
        %calculating value of theta_3 & theta_4 in uncrossed configuration
        theta_4(i)= 2*atan2d((-B-sqrt(B^2-C^2+A^2)),(C-A));
        theta_3(i)= atan2d((r1*sind(theta_1)+r4*sind(theta_4(i))-r2*sind(i)),(r1*cosd(theta_1)-
    else
        theta_4(i)= NaN;
        theta_3(i)= NaN;
    end
end
        figure
        plot(theta_3,'r.','LineWidth',2)
        xlabel('theta_2')
        ylabel('theta_3')
        title('theta_3 v/s theta_2:Uncrossed Configuration ')
        axis([1,360,-360,360])
```



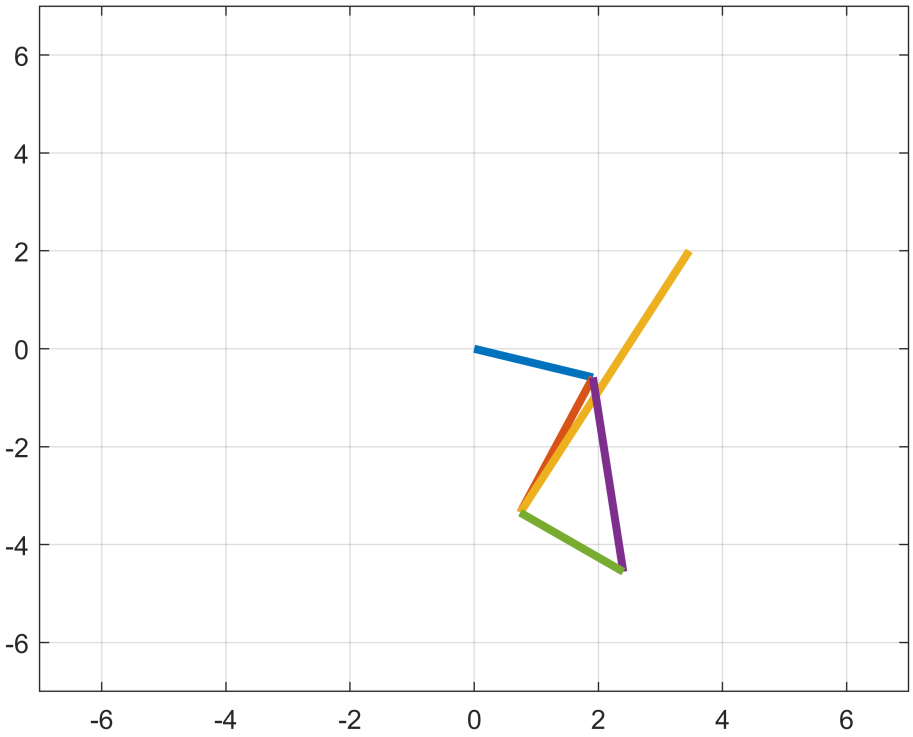

```
figure
plot(theta_4,'r.','LineWidth',2)
xlabel('theta_2')
ylabel('theta_4')
title('theta_4 v/s theta_2: Uncrossed Configuration')
axis([1,360,-360,360])
```



Part 3.A.iii) Plot the Cartesian trajectory (Y_c v/s X_c) of the coupler point C

```
 r_oa=0;
%Lenght of Ground Link:%
r1=4;
%Length of Input Link:%
r2=2;
% Lenght of Coupler: %
r3=3;
%Length of Follower:%
r4=6;
%Rigid Body Side Length: %
r5=4;
%Ground Link Angle: %
theta_1=30;
%Included angle%
rigid_ang=30;

%Defining theta_2 in increments of 1 degree
for i=1:1:360
    %finding value of A, B & C for each valule as theta_2 where variable
    %theta_2 has been replaced by index i
    A= (2*r1*r4*cosd(theta_1))-(2*r2*r4*cosd(i));
    B= (2*r1*r4*sind(theta_1))-(2*r2*r4*sind(i));
    C= r1^2 + r2^2 + r4^2 - r3^2 - 2*r1*r2*(cosd(theta_1)*cosd(i)+sind(theta_1)*sind(i));
    cond=B^2-C^2+A^2;
    %condition for real solution of a theta_4 value existing for a given value
```

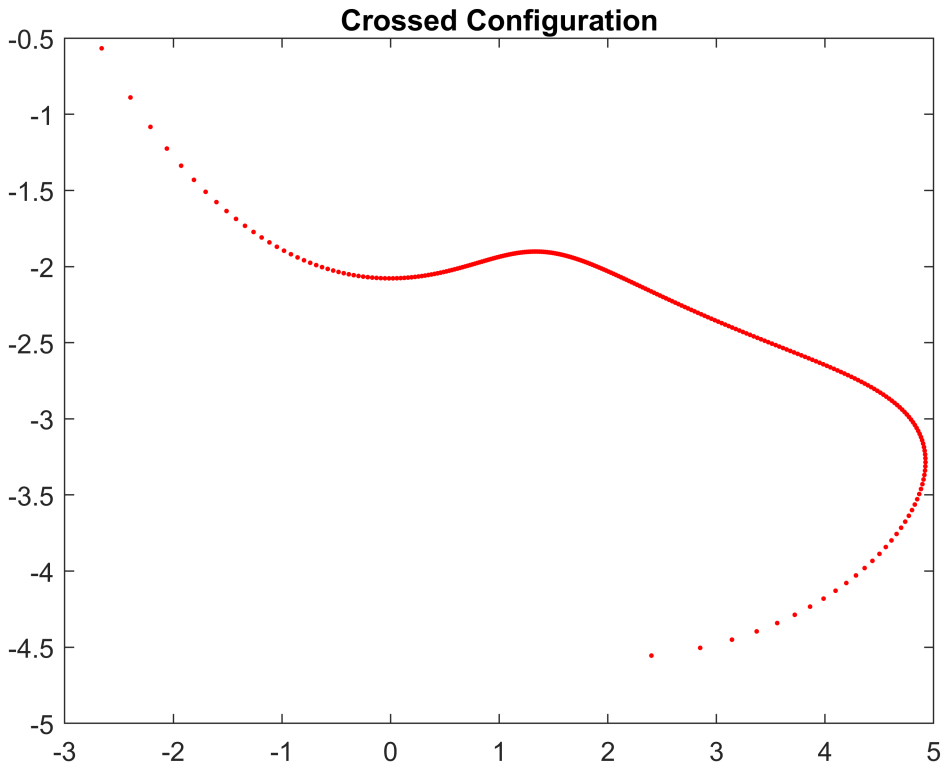
```
    %of theta_1 & theta_2
    if cond>=0
        %calculating value of theta_3 & theta_4 in crossed configuration
        theta_4(i)= 2*atan2d((-B+sqrt(B^2-C^2+A^2)),(C-A));
        theta_3(i)= atan2d((r1*sind(theta_1)+r4*sind(theta_4(i))-r2*sind(i)),(r1*cosd(theta_1)-

        %defining co-ordinates of each point
        Cx(i)=r2*cosd(i)+r5*cosd(theta_3(i)+30);
        Cy(i)=r2*sind(i)+r5*sind(theta_3(i)+30);

        Ao_x(i) = 0;
        Ao_y(i) = 0;

        Ax(i) = Ao_x(i)+r2*cosd(i);
        Ay(i) = Ao_y(i)+r2*sind(i);

        Bo_x(i) = r1*cosd(theta_1);
        Bo_y(i) = r1*sind(theta_1);

        Bx(i) = Ao_x(i)+Ax(i) + r3*cosd(theta_3(i));
        By(i) = Ao_y(i)+Ay(i) + r3*sind(theta_3(i));

        plot([Ao_x(i) Ax(i)], [Ao_y(i) Ay(i)], [Ax(i) Bx(i)], [Ay(i) By(i)],[Bx(i) Bo_x(i)], [B
        %Link Ao-A = blue; Link A-B = Orange; Link B-Bo = Yellow; Link
        %A-C=purple, Link C-B=Green
        grid on;
        L=[-7,7,-7,7];
        drawnow;
        axis(L);
        hold off;

    else
       theta_4(i)= NaN;
       theta_3(i)= NaN;

       xc(i)=NaN;
       yc(i)=NaN;

       Ox(i) = NaN;
       Oy(i) = NaN;

       Ax(i) = NaN;
       Ay(i) = NaN;

       Bx(i) = NaN;
       By(i) = NaN;

       Cx(i) = NaN;
       Cy(i) = NaN;
    end

end
```




```
%Link Ao-A = blue; Link A-B = Orange; Link B-Bo = Yellow; Link
%A-C=purple, Link C-B=Green
```

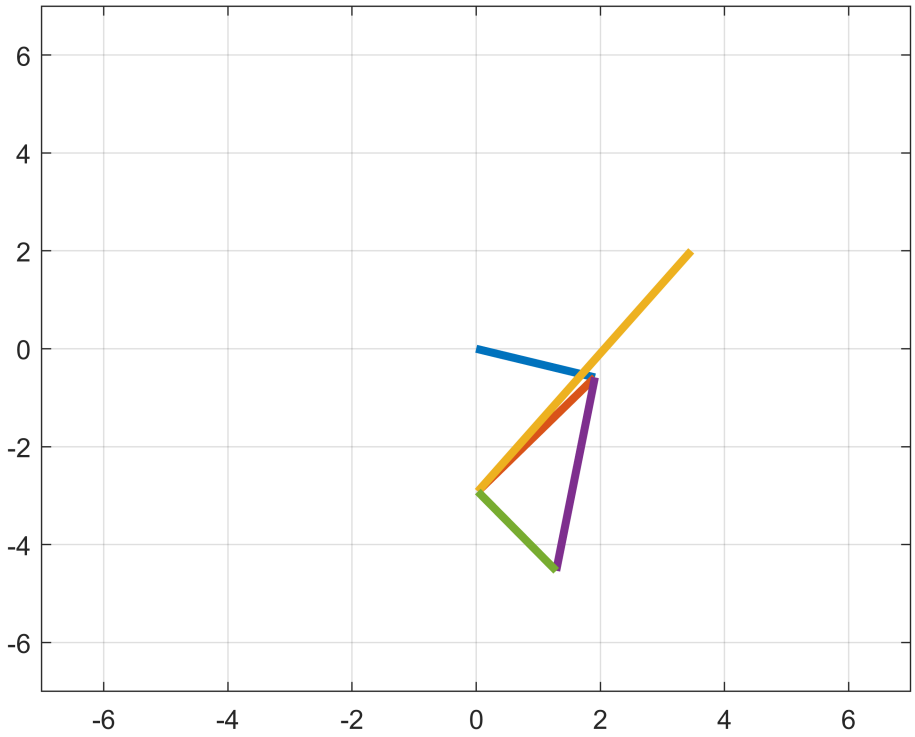
In the mehcanism animation above, the links look like they are changing their length, but that illusion is created as the "hold off" function re-assign limits for the axis in each iteration hence it is the dynamic nature of the axis limit which gives the effect that the link length is changing

```
figure
plot(Cx,Cy,'r.')
title('Crossed Configuration')
```




```
%Defining theta_2 in increments of 1 degree
for i=1:1:360
    %finding value of A, B & C for each valule as theta_2 where variable
    %theta_2 has been replaced by index i
    A= (2*r1*r4*cosd(theta_1))-(2*r2*r4*cosd(i));
    B= (2*r1*r4*sind(theta_1))-(2*r2*r4*sind(i));
    C= r1^2 + r2^2 + r4^2 - r3^2 - 2*r1*r2*(cosd(theta_1)*cosd(i)+sind(theta_1)*sind(i));
    cond=B^2-C^2+A^2;
    %condition for real solution of a theta_4 value existing for a given value
    %of theta_1 & theta_2
    if cond>=0
        %calculating value of theta_3 & theta_4 in Uncrossed configuration
        theta_4(i)= 2*atan2d((-B-sqrt(B^2-C^2+A^2)),(C-A));
        theta_3(i)= atan2d((r1*sind(theta_1)+r4*sind(theta_4(i))-r2*sind(i)),(r1*cosd(theta_1)-

        %defining co-ordinates of each point
        Cx(i)=r2*cosd(i)+r5*cosd(theta_3(i)+30);
        Cy(i)=r2*sind(i)+r5*sind(theta_3(i)+30);

        Ao_x(i) = 0;
        Ao_y(i) = 0;

        Ax(i) = Ao_x(i)+r2*cosd(i);
        Ay(i) = Ao_y(i)+r2*sind(i);

        Bo_x(i) = r1*cosd(theta_1);
        Bo_y(i) = r1*sind(theta_1);
```

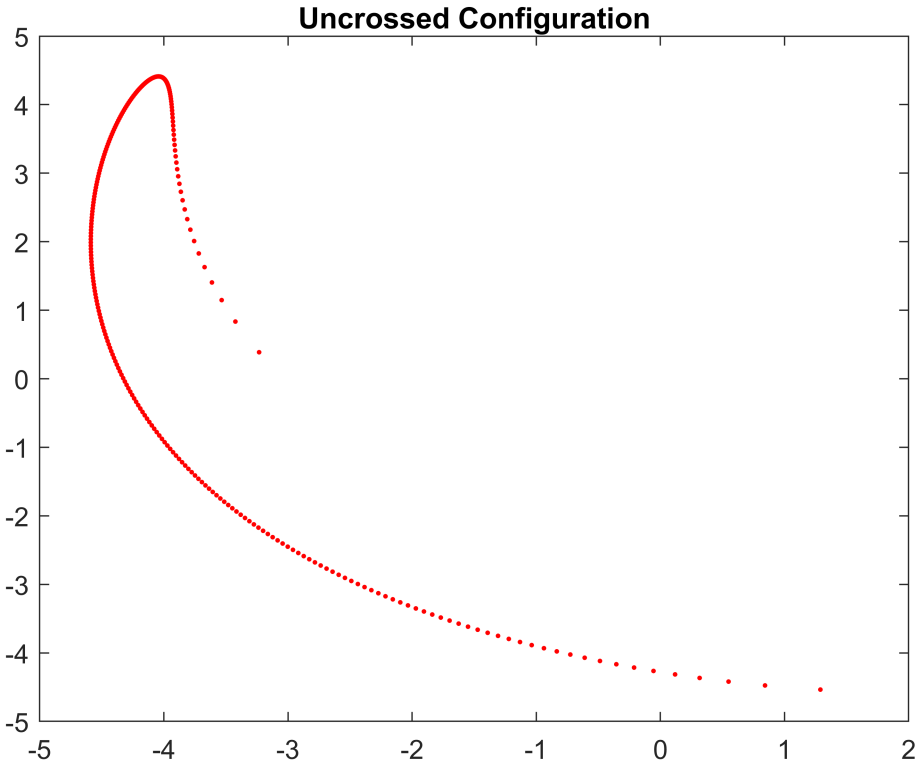
```
        Bx(i) = Ao_x(i)+Ax(i) + r3*cosd(theta_3(i));
        By(i) = Ao_y(i)+Ay(i) + r3*sind(theta_3(i));

        plot([Ao_x(i) Ax(i)], [Ao_y(i) Ay(i)], [Ax(i) Bx(i)], [Ay(i) By(i)],[Bx(i) Bo_x(i)], [B

        grid on;
        L=[-7,7,-7,7];
        drawnow;
        axis(L);
        hold off;

    else
       theta_4(i)= NaN;
       theta_3(i)= NaN;

       xc(i)=NaN;
       yc(i)=NaN;

       Ox(i) = NaN;
       Oy(i) = NaN;

       Ax(i) = NaN;
       Ay(i) = NaN;

       Bx(i) = NaN;
       By(i) = NaN;

       Cx(i) = NaN;
       Cy(i) = NaN;
    end

end
```




```
%Link Ao-A = blue; Link A-B = Orange; Link B-Bo = Yellow; Link
%A-C=purple, Link C-B=Green
```

In the mehcanism animation above, the links look like they are changing their length, but that illusion is created as the "hold off" function re-assign limits for the axis in each iteration hence it is the dynamic nature of the axis limit which gives the effect that the link length is changing

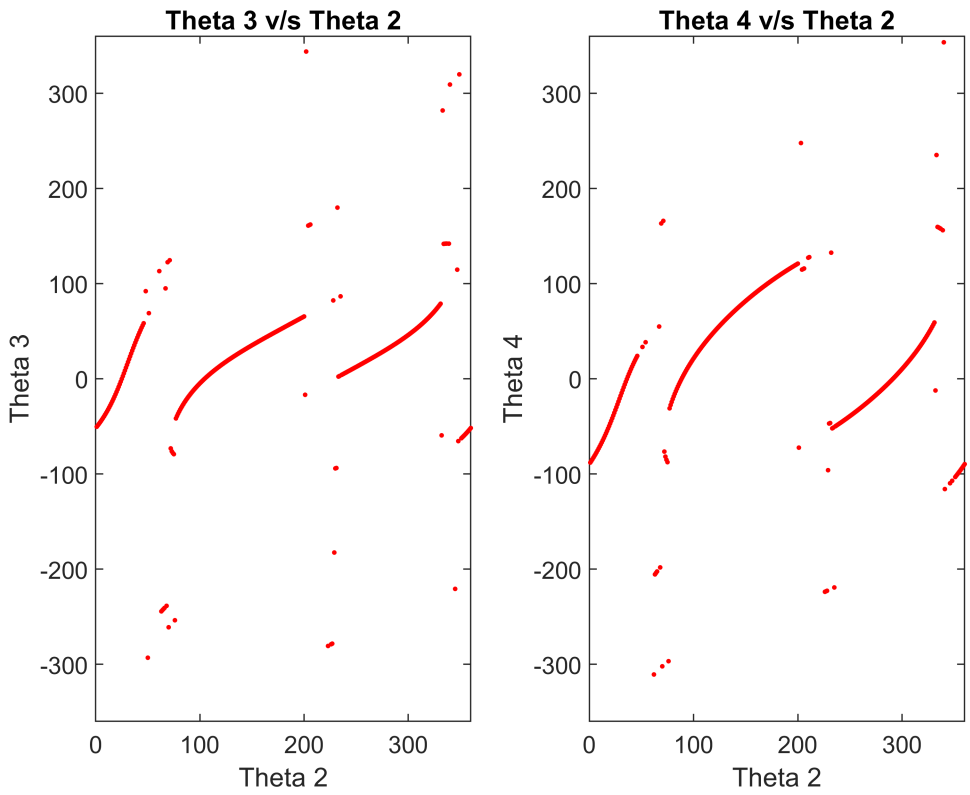
```
figure
plot(Cx,Cy,'r.')
title('Uncrossed Configuration')
```



B(i) is the function 'Fourbar_Pos_NR_GivenT2.m'

```
%(ii)
clear all;
L = [2 3 6 4];
for i = (1:1:360)
    theta = [30 i];
    [Theta] = Fourbar_Pos_NR_GivenT2(L,theta);
    C1(i,1) = L(1)*cos(theta(1)*(pi/180)) + L(4)*cos(Theta(2,1)) - L(2)*cos(i*(pi/180)) - L(3)*
    C2(i,1) = L(1)*sin(theta(1)*(pi/180)) + L(4)*sin(Theta(2,1)) - L(2)*sin(i*(pi/180)) - L(3)*
    angles(:,:,i) = Theta;
end

theta3 = angles(1,:,:);
theta3 = reshape(theta3,[360,1]);
theta3 = theta3*(180/pi);
theta4 = angles(2,:,:);
theta4 = reshape(theta4,[360,1]);
theta4 = theta4*(180/pi);
theta2 = (1:1:360)';
figure(29);
subplot(1,2,1)
plot(theta2,theta3,'r.');
axis([0,360,-360,360]);
xlabel('Theta 2');
```

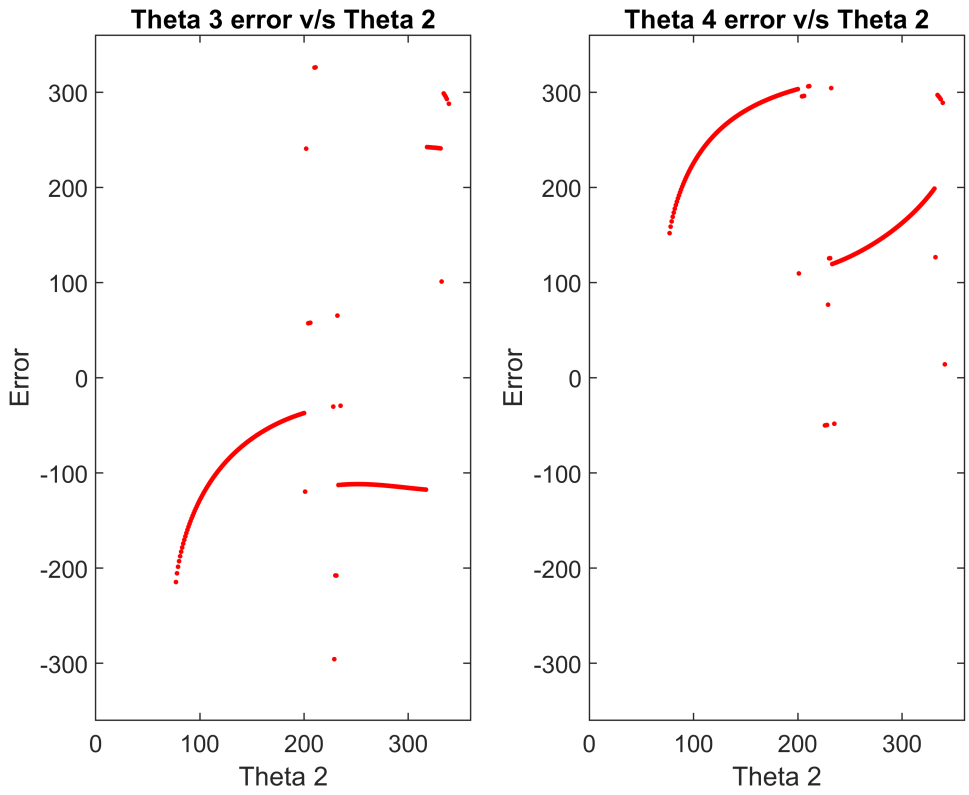
```
ylabel('Theta 3');
title('Theta 3 v/s Theta 2');
subplot(1,2,2)
plot(theta2,theta4,'r.');
axis([0,360,-360,360]);
xlabel('Theta 2');
ylabel('Theta 4');
title('Theta 4 v/s Theta 2');
```



(iia) The 'analyticalsol.xls' contains the theta 3(col2) & theta 4(col3) values for the uncrossed config obtained in Q3 part A(ii)

```
anly_sol = table2array(readtable("analycticalsol.xlsx"));
theta3_anly = anly_sol(:,2);
theta4_anly = anly_sol(:,3);

theta3_err = theta3 - theta3_anly;
theta4_err = theta4 - theta4_anly;

figure(30);
subplot(1,2,1)
plot(theta2,theta3_err,'r.');
axis([0,360,-360,360]);
xlabel('Theta 2');
ylabel('Error');
title('Theta 3 error v/s Theta 2');
subplot(1,2,2)
```

```
plot(theta2,theta4_err,'r.');
axis([0,360,-360,360]);
xlabel('Theta 2');
ylabel('Error');
title('Theta 4 error v/s Theta 2');
```



The theta 3 and theta 4 values obtained via both the solutions give a solution at the same values of theta 2.

3B(iii)

```
clear all
theta_o = [1,0];
L = [4 2 3 6];
theta1 = 30*(pi/180);
theta2 = (1:1:360)';
fun = @Fourbar_Pos_FSOLVE_GivenT2;
theta_f = fsolve(fun,theta_o);
```

```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
```

The code for 'fsolver' seems to be working for single values of theta 2, but fails to give it for a range of values of theta 2. Need to fix it.